Silent steps in transition systems and Markov chains

DOI:
10.6100/IR627345

Document status and date:
Published: 01/01/2007

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Silent Steps in Transition Systems and Markov Chains

IPA Dissertation Series 2007-08
Typeset using LaTeX2e
Printed by University Press Facilities, Eindhoven
Cover design by Jelena Radošević, adaptation by Paul Verspaget

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN
Trčka, Nikola
Silent steps in transition systems and Markov chains / door Nikola Trčka.
Eindhoven : Technische Universiteit Eindhoven, 2007
Proefschrift. ISBN 978-90-386-1045-0
NUR 993
Section headings: Markov chains / transition systems
CR: F.1.2 / F.3.1 / F.3.2 / G.3

The work in this thesis has been carried out under the auspices of the research school IPA (Institute for Programming research and Algorithmics).

The author was employed at the Eindhoven University of Technology, supported by the Netherlands Organization for Scientific Research (NWO), project 612.064.205

Silent Steps in Transition Systems and Markov Chains

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
Rector Magnificus, prof.dr.ir. C.J. van Duijn, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen
op donderdag 28 juni 2007 om 16.00 uur

door

Nikola Trčka

geboren te Belgrado, Servië

Dit proefschrift is goedgekeurd door de promotoren:

prof.dr. J.C.M. Baeten
en
prof.dr.ir. J.E. Rooda

Copromotor:
dr. S.P. Luttik

Preface

This thesis is an outcome of my Ph.D research at Eindhoven University of Technology, started in July 2003. Many people have contributed to my life and to my research in different ways. Here I can only mention some of them.

First of all I would like to thank my supervisor Jos Baeten for giving me a position while I was still a student. I highly appreciated working with Jos and being under his supervision. He is the leader of our group and the main reason for its friendly, relaxed, positive, and productive atmosphere. I would also like to thank Koos Rooda who accepted to be my second supervisor.

I am very happy to have Bas Luttik as a co-supervisor. He is an excellent supervisor. He carefully read every word I wrote, and at any time was available to answer my questions. From Bas I have learned a lot. Not only a vast amount of computer science, but also how to write better, how to be more precise, how to get a paper accepted, how to give good talks, and many many other things. Bas, thanks a lot!

A large part of the research presented in this thesis is joint work. Most of all I would like to thank my friend and colleague Jasen Markovski. Jasen came as a Ph.D. student just at the time when I was searching for a victim to join me in the "Markov chain business". He immediately got hooked on the topics and the result was a three-year period of great collaboration. Without Jasen many pages in this thesis would have been left blank. I also thank Rob van Glabbeek for helping Bas and me solve some complicated issues regarding divergence in branching bisimulation. Ever since he joined, he has been providing us with many insightful comments.

I thank the members of the committee, Onno Boxma, Holger Hermanns, and Jaco van de Pol, for reviewing the manuscript of this thesis and giving me valuable comments. I also thank Kees van Hee and Wan Fokkink for accepting to be members of the defense committee.

My work was supported by the NWO project TIPSy. I appreciated the project meetings and the discussions with the project members Wan Fokkink (in the beginning), Koos Rooda, Asia van de Mortel-Fronczak, Jaco van de

Pol, Elena Bortnik, and Anton Wijs.

I thank Holger Hermanns, Joost-Pieter Katoen, and Jane Hillston for inviting me to visit them and present my work. I thank Jane Hillston also for giving me the chance to work as a postdoc in her group in Edinburgh.

I thank my colleagues of the Formal Methods Group for contributing to the pleasant working atmosphere.

We have many friends in The Netherlands. They have made our stay a very enjoyable experience and full of memories. They know who they are and I thank them all. Unique thanks go to Ana, whose useful advice (and furniture) helped us survive in the beginning, Georgi, who showed his hospitality when it was needed, Christina, whom we stuffed with letters when we were away, and Jasen, whose "secretarial work" is greatly appreciated.

I thank Jeca for designing the cover of this thesis.

I thank my parents and my sister for their encouragement and support.

Finally, I thank my wife Marija for all the love and joy that she brings into my life. I look forward to the wonderful time that will come soon when our little family expands.

Eindhoven, May 2007 Nikola Trčka

Summary

Silent Steps in Transition Systems and Markov Chains

Formal methods provide a set of notations and techniques for construction of mathematical models of systems and for (automatic) verification of these models against requirements. The requirements are usually represented in terms of a set of properties that a system should satisfy. A property can be qualitative or quantitative. A qualitative property is a property pertaining to the functional behavior of a system (e.g. "the system never deadlocks"); a quantitative property is a property pertaining to a system's performance (e.g. "the throughput of the system is as desired").

This thesis consists of three parts. The first part solves some problems related to functional verification of systems. The third part considers performance analysis and contributes to the field of Markov processes. The second part serves as a bridge between the first and the third part. It recollects some standard results from the verification world but explains them in the standard matrix-analytic language of Markov processes. In each part the focus is on the elimination of silent steps, i.e., of steps in a system that are considered unobservable. A short summary of each part follows.

In Part I, we define timed doubly-labeled transition systems as transition systems that incorporate data, timing and successful termination. We define a silent step to be a step that does not change the global state and that involves the execution of an internal action. We also define an equivalence relation that abstracts away from silent steps. The main contribution of Part I is the sequence of adaptations that have to be made in order for this equivalence to be a congruence for a standard modeling language.

Part II approaches the theory of transition systems and bisimulations from matrix theory. We define transition systems with successful termination as tuples of matrices over a boolean algebra of actions. We also define

some standard operations on transition systems in matrix theory, and give matrix definitions of forward and backward strong bisimulation, of bisimulation up-to a relation, and of weak and branching bisimulation. The main purpose of Part II is to show the analogies between transition system theory and Markov chain theory.

Part III introduces two types of silent transitions in the theory of Markov reward processes. The first type of silent step is an instantaneous step that is assigned a probability with which it is selected. The second is an instantaneous step for which this probability is left unspecified. This is to express internal non-determinism. For each type, two different ways of eliminating silent steps are provided and compared, one based on lumping of states and the other on a more traditional aggregation approach. The results of Part III can serve as the correctness criterion for various compositional Markov (reward) chain generation methods.

Origin of the parts Most of the material presented in this thesis is an extension of the joint work that was published before in several papers.

- Part I is based on the following papers:
  - N. Trčka - *Verifying Chi Models of Industrial Systems in Spin.* In Proceedings of ICFEM'06, Macau, China.
  - B. Luttik, N. Trčka - *Stuttering Congruence for Chi.* In Proceedings of SPIN'05, San Francisco, CA, USA. A longer version published as a Computer Science Report 05/13, Eindhoven University of Technology, 2005.
  - E. Bortnik, N. Trčka, A.J. Wijs, B. Luttik, J.M. van de Mortel-Fronczak, J.C.M. Baeten, W.J. Fokkink, J.E. Rooda, *Analyzing a Chi Model of a Turntable System Using SPIN, CADP and UPPAAL.* Journal of Logic and Algebraic Programming, vol. 65, 2005, pp 51-104.
  - R. van Glabbeek, B. Luttik, N. Trčka - *Branching Bisimulation with Explicit Divergence.* Submitted for publication.
- Part II is based on the following unpublished manuscript:
  - N. Trčka - *Transition Systems in Matrix Theory.*
- Part III is based on the following papers:

– J. Markovski, N. Trčka - *Lumping Markov Chains with Silent Steps*.
  In Proceedings of QEST'06, Riverside, CA, USA.
  A longer version published as Computer Science Report 06/13, Eindhoven University of Technology, 2006.
– J. Markovski, N. Trčka - *Eliminating Fast Transitions and Silent Steps in Markov Chains by Aggregation: Reduction vs. Lumping*.
  Submitted for publication.

Table of Contents

List of Figures

Part I

Silent Congruence and Timed Silent Congruence

Chapter 1

Introduction

Formal methods provide a set of notations and techniques for construction of mathematical models of systems and for (automatic) verification of these models against some requirements, i.e. against a set of properties that a system should satisfy. These models, usually some kind of state transition graphs, are rarely obtained by hand. A system is first specified in a formal specification language, a language similar to a programming language. The distinguishing characteristic of a formal specification language, apart from features to express many different aspects of systems, is its formal semantics. The final model, called state space, is next automatically obtained by the semantics of the language. A typical formal specification language involves the notion of a process and provides different constructs used to compose processes, such as non-deterministic choice, repetition, sequential or parallel composition.

Sometimes, prior to verification there is a need to transform a system's specification into another specification. There are several reasons why this can be useful:

1. The generated state space of a specification is expected to be too large. This brings the need for its symbolic representation, that is, a simplified specification that lies between the original specification and its state space. *Linearization* of specifications in the modeling language μCRL [17] is an example of this method [96].

2. Different symbolic optimization techniques can be performed on the model, e.g. to reduce the size of the final state space before it is generated. For example, see again [96] for the symbolic manipulations on the linearized μCRL process.

3. The model is to be translated to another specification language for the purpose of verification in some other environment. In case the source and the target language do not have a common semantics, the new specification should be in a syntactical form that is easier or trivial to translate. The correctness of the translation process can then take place entirely within the realm of the source language. The usefulness of this approach was shown in [95, 74] where the translation of the modeling language χ [90] to PROMELA, the input language of the model checker SPIN [65], is presented. In Chapter 6 we discuss some parts of this translation process as an application of our results.

The goal of Part I is to find a suitable correctness criterion for these transformations. To be precise, we want to define an equivalence relation on process specifications that:

1. preserves all relevant properties of a system,
2. is a congruence, i.e., is compatible with all the standard constructs of a typical modeling language, and
3. allows for sufficient flexibility of transformations.

The first goal makes sure that every transformation modulo this equivalence is correct. The equivalence should satisfy the congruence property because we want to be able to transform only a part of the specification and still obtain an equivalent specification. The last goal is clear. We want to have freedom to simplify as much as possible. The equivalence should be as coarse as possible still satisfying the first two requirements.

In the following sections we explain how we will achieve this goal, motivating every decision.

1.1 Underlying model

To obtain an equivalence on process specifications we follow the standard approach and first choose a suitable mathematical model for the representation of systems. This model serves as the underlying model, i.e. the state space, generated from the specifications. The equivalence is first defined in that model and then lifted to the level of process specifications. We discuss the two most common formal models.

Labeled transition systems are a well established formalism for modeling of the qualitative aspects of systems, focusing on the behavioral part. A labeled transition system is a directed graph in which each node represents a state of the system, and each arrow is labeled by an action denoting that the system can perform a transition from the source state to its target state while executing that action. Figure 1.1a depicts a transition system.

Another well known formalism for the representation of systems are Kripke structures. They are also directed graphs with nodes representing states of a system, but they have labels associated to states denoting which propositions are satisfied by the system when in that state. Contrary to labeled transition systems, the focus is not on the actions that a system can perform but rather on their effect on its data-state. A Kripke structure is depicted in Figure 1.1b.

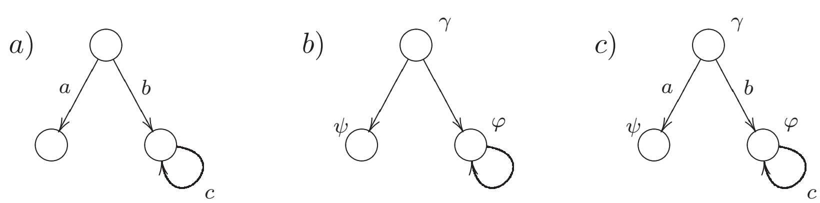


Figure 1.1: a) Labeled transition system, b) Kripke structure and c) doubly-labeled transition system

A *doubly-labeled transition system* is the combination of a labeled transition system and a Kripke structure. It generalizes both formalisms by allowing labels to be both on arrows and states. When the action labels are ignored, a doubly-labeled transition system becomes a Kripke structure; when the state labels are ignored, it becomes a labeled transition system. A doubly-labeled transition system obtained by combining the labeled transition system of Figure 1.1a and the Kripke structure of Figure 1.1b is depicted in Figure 1.1c.

The behavioral part and the data part of a system are often inseparable. Most modeling languages involve some notion of state variable (for data) together with action executions (e.g. for the synchronization of parallel components). This motivates us to work with a model that integrates these two features and to define our equivalence on it. One such model is a doubly-labeled transition system. Since modeling languages give rise to doubly-labeled transition systems only indirectly (they keep variables with values on states and not complete sets of propositions), we also need that our equivalence can be automatically interpreted on the semantics of those languages.

1.2 Properties of interest

In this section we define the main properties that must be preserved under transformation. The formal notion of a property of a system comes from the type of verification that is to be used.

The most widely used verification technique today is *model checking* [31]. This technique performs an exhaustive search of the state space checking if a certain property holds of the system. The property is specified by a formula of some *temporal logic*, a logic that allows us to say things like: if a machine is given certain input, then it will eventually produce a correct output. Once the property is formalized, model checking becomes a completely automated process.

There are many variants of temporal logics (consult e.g. [43, 94]) and there is no common agreement on which is superior. The logic can be *linear-time*, when reasoning is about a single sequence of states (like LTL [88]), or it can be *branching-time*, when reasoning involves several different branches starting from a state (like CTL [30] or Hennessy-Milner logic [59]). There are also logics that combine these two features, like CTL* [44] and μ-calculus [71] (see their comparison in [21]). Also, a logic can consider only infinite, or both finite and infinite, executions of the system. Depending on whether the underlying model is a labeled transition system or a Kripke structure, the logic is called *action-based* (reasoning is about what actions can be performed in a state) or *state-based* (reasoning is about the validity of propositions in a state). Traditionally, the logics LTL, CTL and CTL* are interpreted over Kripke structures, and therefore are state based, while Hennessy-Milner logic and μ-calculus are action based. However, due to its large expressivity, the logic CTL* has also been interpreted in the action-based setting of labeled-transition systems [36].

We choose the preservation of temporal logic formulas to be the major part of our correctness criterion. That is, if the system is to be checked by CTL model checking, then we require that the original and the transformed system satisfy the same set of CTL formulas. To obtain more applicability we want to preserve both state- and action-based logics, and both linear- and branching-time logics.

There are also two other important properties of systems that we want to preserve. The first is *deadlock*, i.e. a system's inability to proceed, and it should be preserved for obvious reasons. The second property is *divergence*.

It represents a systems's ability to repeat the same behavior indefinitely. It is a subject of discussions whether divergence is really important or not. However, divergence sensitivity is, to some extent, already built-in in many temporal logics. Because of this, and because, as we will see later, divergence becomes crucial when timing is introduced, we incorporate it into the theory.

1.3 Bisimulation relations

Establishing the correctness criterion directly is usually cumbersome. It is often more convenient to equate specifications by establishing that they are related according to some behavioral equivalence pertaining to the operational semantics of the modeling language. We adopt this approach.

For the setting of labeled transition systems and Kripke structures there is a variety of trace- and bisimulation-like equivalences each with a different temporal logic that it characterizes [52, 51]. Many of these equivalences have been defined for many modeling languages and shown to be congruences. If the original and the transformed model are to agree on every step they take, then they are equivalent modulo *strong bisimulation* equivalence [84, 79]. This equivalence is known to characterize the logic CTL* in the setting of Kripke structures, and to characterize Hennessy-Milner logic for labeled transition systems. Strong bisimulation is often not appropriate for establishing the correctness of the transformation because it equates too few states due to the requirement that *every* step needs to be simulated. A model is often transformed with the introduction of some auxiliary actions that do not change the global state of data (e.g. think of variables exchanging values and the temporary variable introduced to achieve that). So, in our case, it is better to work with weaker equivalences.

Sometimes a system can perform internal steps of which the impact is considered unobservable. Weaker equivalences abstract away from these steps but require that the other, i.e. visible, steps are simulated. For labeled transition systems internal steps are the steps labeled by the *internal* action τ and systems are usually related by a *weak* [79] or by a *branching* [53, 11] bisimulation equivalence. Several variants of Hennessy-Milner logics are given for both weak and branching bisimulations [60, 37]. In the setting of Kripke structures an internal transition is a transition from a state to a state that has the same set of propositions satisfied. Systems are then usually related by *stuttering equivalence* [22, 37]. This equivalence characterizes a variant of the logic CTL^*_{-X} (the logic CTL* but without the operator *next*) [22]. It is shown in [36, 37] that branching bisimulation and stuttering equivalence correspond, i.e. that they follow the same idea but in a different setting. Figure 1.2a depicts a labeled transition system with a branching bisimulation on it. Figure 1.2b depicts a (corresponding) Kripke structure with a (corresponding) stuttering equivalence on it. Note a small difference. Branching bisimulation does not require that the first state of the left system is connected to the second state of the right system.

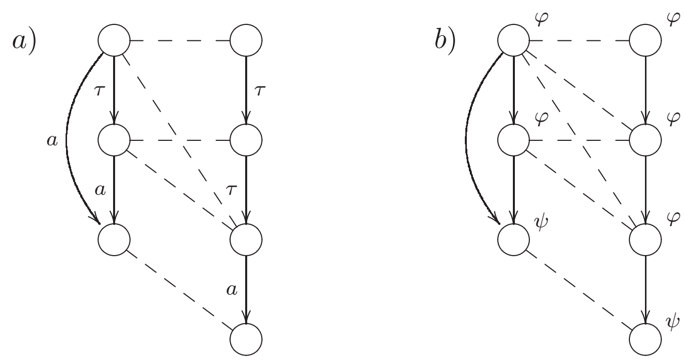


Figure 1.2: a) Branching bisimulation and b) stuttering equivalence

When working with doubly-labeled transition systems, the bisimulation we use should be a combination of the bisimulations developed for labeled transition systems and Kripke structures. As explained, to achieve the first part of the goal, that is, a sufficient flexibility of transformation, we need an equivalence that is weaker than strong bisimulation. We choose to combine branching bisimulation and stuttering equivalence. Since the two equivalences correspond, the decision is natural. An unobservable step in a model is then a step that is labeled by the internal action τ and such that the resulting state satisfies the same set of propositions as the starting one.

With the decision to (conservatively) combine branching bisimulation and stuttering equivalence we are sure that our transformed model satisfies the same set of $\text{CTL}^*_{-\text{X}}$ formulas in both the action- and state-based setting [36, 37]. Since this logic is very expressive, covers almost all logics used in practice, and combines branching-time and linear-time, it follows that if two specifications are related by the new equivalence, then the main part of the correctness requirement is satisfied. However, the equivalence does not guarantee that deadlock and divergence are also preserved. In the next section we explain how this can be solved.

1.4 Divergence

Divergence is infinite repetition of the same behavior and it should be preserved in the transformed system. Since branching bisimulation and stuttering equivalence make sure that all observable steps are properly simulated, it is clear that we can only need to consider a more strict version of divergence. A system is considered *divergent* if it can perform infinitely many internal steps. The idea is that this behavior cannot be ignored, i.e. that an unobservable step cannot be considered unobservable anymore if it is performed indefinitely, and that it must be equally simulated.

The original version of branching bisimulation abstracts totally from divergence. For example, the systems in Figure 1.3 are all branching bisimilar. In [53] another condition is added to the definition saying that a state related to every state on a divergent path must also be divergent and with all the states in its divergent path related to all the states in the divergent path of the other state. This notion is known as *branching bisimulation with explicit divergence*. It distinguishes all the three systems from Figure 1.3.

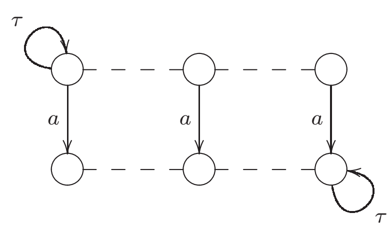


Figure 1.3: Branching bisimulation is blind to divergence

The most generally accepted version of CTL^*_{-X}, namely the one interpreted on maximal (infinite) paths only, does not ignore divergence except when relating a divergent state with a state that cannot perform any step. In other words, it identifies deadlock and livelock. The two systems in Figure 1.4a are indistinguishable while the systems in Figure 1.4b are distinguished by the CTL^*_{-X} formula $\forall \mathsf{F} \psi$. The formula encodes the property, that for all (maximal) paths there is a state in which ψ holds. This is clearly not satisfied by the first system since it has an execution where it only stays in the first state. To obtain a bisimulation-like notion that characterizes CTL^*_{-X}, instead of adding a divergence condition, in [37] a divergence sensitive version of stuttering bisimilarity is obtained by extending Kripke structures with a fresh state that serves as a sink-state for deadlocked or divergent states. This approach is not suitable in our case, because it does not allow us to interpret the equivalence directly on the operational semantics of some modeling language. In addition, divergence sensitive stuttering bisimilarity identifies deadlocked with non-deadlocked states, which not only violates our requirement of deadlock preservation but also introduces a congruence problem for parallel composition. So, adding a divergence condition and treating divergence in all cases is a more suitable approach for us. For stuttering equivalence several divergence conditions appeared in the literature [85, 50, 81]. Adapting these conditions to our setting causes some complications when proving transitivity of the relation, so we develop our own condition.

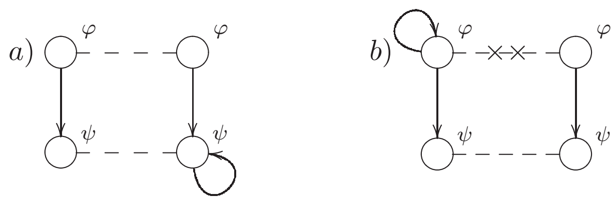


Figure 1.4: Stuttering equivalence does not ignore divergence except in deadlocked states

1.5 Some extensions

There are two more aspects of systems that we choose to cover; these are *successful termination* and *timing*. Successful termination can ease the modeling and it gives better axiomatizations; timing allows for modeling of time critical systems. In the setting of branching bisimulation, termination and timing have already been successfully added to the theory [9, 7, 6]. These additions can be naturally interpreted in the setting of stuttering equivalence. Since our aim is to combine the two equivalences, we choose to suitably adapt and (conservatively) include the conditions for timed branching bisimulation from [97, 7, 6] and for branching bisimulation with termination from [9]. Note that, although it is possible to include successful termination and timing in the correctness criterion, by suitably extending the logic $\text{CTL}^*_{-\text{X}}$ (see e.g. timed CTL [2]), we will not do so. There is no real standard for timed logics and our purpose is more to illustrate the problems when successful termination and timing are introduced to the theory. This is justified more by the fact that these features are usually only present in the specification and often discarded at the verification phase.

Successful Termination The main reason to incorporate explicit successful termination in a process theory is that it is needed for a proper treatment of sequential composition. The theory becomes more modular and algebraic if action execution and termination are not combined. This is important if in the future we decide to axiomatize our equivalence.

In labeled transition system setting (and, to the best of our knowledge, the issue has not been considered in the setting of Kripke structures) explicit termination is obtained by allowing a state to have a termination predicate attached to it, denoting that the system can successfully terminate in that state. In addition, as a counterpart to the deadlock constant, the so-called *empty process* is introduced into the specification language. This process can only successfully terminate and it serves as a neutral element for sequential and parallel composition. It is shown in [70, 98] that many things can be modeled more easily if the empty process is present in the language. This is one more reason to incorporate successful termination into our theory.

Branching bisimulation deals with successful termination similarly to action execution. A successful termination is simulated by a successful termination preceded by a zero or more silent steps (see Figure 1.5). This idea is lifted to our setting naturally and clearly as a conservative extension.

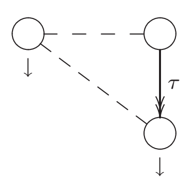


Figure 1.5: Branching bisimulation and successful termination

Timing In many systems timing plays a major role. A typical example, found in industrial systems, is a distributor that delivers products to a machine and discards them if the machine is not available within a certain amount of time. Other examples are found in the modeling of controllers. This forces us to incorporate timing into our setting.

When timing is to be incorporated into a theory several choices need to be made (see [7]). First, whether time will be *discrete*, i.e. divided into slices, or *dense*, measured on the continuous scale. Second, whether it will be *absolute*, measured by a global clock, or *relative* to the previous action. Third, a more technical one, is whether to associate the passage of time with actions, i.e. to stamp actions with duration or with the explicit time point indicating when they become available, or to leave actions unstamped (and timeless) and treat the passage of time independently.

We take time to be discrete; since computers are in general discrete, this is not a serious restriction. We also take it to be relative and independent of actions. This is considered to be the simplest version of timing (see e.g. [7, 48] for the possible complications in the other settings).

The passage of time is indicated in the model by a special transition $\overset{\Delta}{\mapsto}$ called *tick*. This transition represents that the system is moving to the next time slice. To correctly simulate ticks, branching bisimulation has been extended into a timed branching bisimulation [97, 7, 6]. The idea is the same as with actions; a tick must be simulated by a tick but preceded by zero or more internal steps (see Figure 1.6).

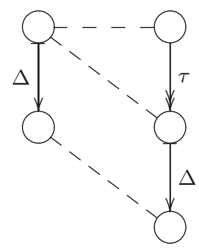


Figure 1.6: Branching bisimulation and discrete timing

It is straightforward to extend timed branching bisimulation to the setting of (timed) doubly-labeled transition systems. However, it is not straightforward to extend it to the setting with explicit termination if the congruence property is to remain. We devote Chapter 5 to adapting the timed branching bisimulation so that it stays a congruence for sequential composition. Even though we work with doubly-labeled transition systems and with divergence, an observant reader will notice that these additions are independent of the congruence problem.

Note that detecting and properly simulating divergence is crucial when timing is involved. Suppose that we relate the two systems from Figure 1.7. The first system can only move to the next time slice. The second system, however, can stay in the first state performing the internal (and timeless!)

step, thus stopping time. This certainly should not be considered as equivalent behavior.

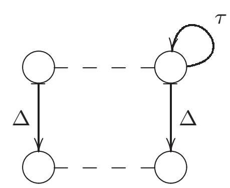


Figure 1.7: Divergence and timing

1.6 Refined goal

Based on the previous discussions we refine our goal set in the beginning. The new objective is to develop an equivalence that:

- is defined on process terms of some standard modeling language,
- is a conservative extension of branching bisimulation and stuttering equivalence,
- preserves deadlock,
- is divergence sensitive,
- incorporates timing and successful termination, and
- is a congruence.

Even though branching bisimulation and stuttering equivalence are known to be equivalences and congruences for standard basic process algebras, their lifting to our setting introduces several complications due to the addition of data, divergence, successful termination, and timing.

1.7 Outline

The outline of Part I is as follows.

In Chapter 2 we formalize the notion of a doubly-labeled transition system and introduce a bisimulation relation that is a combination of branching bisimulation and stuttering equivalence, and that incorporates divergence and successful termination. We show that it generalizes some existing divergence sensitive equivalences from the literature and that it preserves deadlock.

In Chapter 3 we introduce a specification language κ that generates doubly-labeled transition systems with successful termination. The language is very expressive and is designed to serve as a core of any language for modeling of systems.

In Chapter 4 we show that silent bisimulation is not a congruence for most of the operators of the specification language. Similar problems have been recognized and solved before and so, following the same footsteps, we adapt silent bisimulation and turn it into a congruence.

Chapter 5 adds timing into consideration. We define a timed doubly-labeled transition system and extend the language κ to enable the modeling of delays. We first show that the straightforward lifting of timed branching bisimulation to this setting does not work (fails to be a congruence for sequential composition) due to the presence of successful termination. We define a timed silent bisimulation as a bisimulation that treats timing and termination in a combined fashion, based on the fact that in the semantical rules they always go together.

Finally, in Chapter 6 we show how our ideas can be used to verify the translation from the engineering language χ to PROMELA, the input language of SPIN.

Chapter 2

Silent bisimulation

As explained in the introduction, to cover both data and behavioral aspects of systems we take doubly-labeled transition systems for our working model. In this chapter we give a formal definition of a doubly-labeled transition system and we introduce a notion of silent bisimulation as a combination of branching bisimulation and stuttering equivalence. The bisimulation incorporates divergence and successful termination but not yet timing. We first prove that silent bisimulation is an equivalence. Then we show that it generalizes other divergence-sensitive branching bisimulations and stuttering equivalences from the literature. Recall that this ensures that the most important correctness requirement, that is the preservation of the $\text{CTL}^*_{-\text{X}}$ formulas, is satisfied. It also ensures that divergence is properly simulated. We compare our divergence condition with some other divergence conditions we found in the literature, and argue that ours is the most compositional one. Finally, we show that silent bisimulation preserves deadlock, which is another correctness requirement.

2.1 Doubly-labeled transition system

Doubly-labeled transition system were first introduced in [37] as a tool to relate branching bisimulation and stuttering equivalence. A doubly-labeled transition system is a directed graph with labels on both arrows and states. The labels on arrows denote the actions that the system can perform, and the labels on states indicate which data propositions are satisfied in a state. Since we integrated successful termination into the setting, we also add a special predicate denoting which states are considered successfully terminated. The formal definition follows.

Definition 2.1.1 (Doubly-labeled transition system) Let A be a set of *actions* and let Π be a set of *atomic propositions*. A *doubly-labeled transition system* is a quadruple $(S, \rightarrow, \downarrow, \ell)$ where:

- S is a set of *states*,
- $\rightarrow \subseteq S \times \mathsf{A} \times S$ is the *transition relation*,
- $\downarrow \subseteq S$ is a set of *(successfully) terminated* states, and
- $\ell : S \rightarrow \mathcal{P}(\Pi)$ is the *state-labeling* function.

The set of all doubly-labeled transition systems with set of actions A and set of atomic propositions Π is denoted $\mathcal{T}_{\mathsf{A},\Pi}$. □

We write $s \xrightarrow{a} s'$ instead of $(s, a, s') \in \rightarrow$, and $s\downarrow$ instead of $s \in \downarrow$. We abbreviate the statement '$s \xrightarrow{a} s'$ or ($a = \tau$ and $s = s'$)' by $s \xrightarrow{(a)} s'$. We also write $s \rightarrow s'$ when $s \xrightarrow{\tau} s'$ and $\ell(s) = \ell(s')$ and call it an *internal* step. We denote by $\rightarrow^{+}$ and $\twoheadrightarrow$ respectively the transitive and the reflexive-transitive closure of $\rightarrow$.

2.2 Silent bisimulation

We now introduce a relation on doubly-labeled transition systems called silent bisimulation. The relation is essentially an extension of branching bisimulation with termination and divergence, with the extra requirement that it only relates states with the same set of propositions satisfied.

Definition 2.2.1 (Silent bisimulation) Let $(S, \rightarrow, \downarrow, \ell) \in \mathcal{T}_{\mathsf{A},\Pi}$. A symmetric binary relation $R \subseteq S \times S$ is called a *silent bisimulation* on $(S, \rightarrow, \downarrow, \ell)$ if, for all $(s, t) \in R$, the following holds:

$\langle\mathsf{lab}\rangle$ $\ell(s) = \ell(t)$,

$\langle\mathsf{term}\rangle$ if $s\downarrow$, then there exists a $t' \in S$ such that $t \twoheadrightarrow t'$, $t'\downarrow$ and $(s, t') \in R$,

$\langle\mathsf{tran}\rangle$ if $s \xrightarrow{a} s'$ for some $a \in \mathsf{A}$ and $s' \in S$, then there exist $t', t'' \in S$ such that $t \twoheadrightarrow t'' \xrightarrow{(a)} t'$, $(s, t'') \in R$ and $(s', t') \in R$, and

$\langle\mathsf{div}\rangle$ if there is an infinite sequence of states $s_0, s_1, s_2, \ldots \in S$ such that $s_0 = s$, $s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \cdots$ and $(s_i, t) \in R$ for all $i \geqslant 0$, then there exists a $t' \in S$ such that $t \rightarrow^{+} t'$ and $(s_k, t') \in R$ for some $k \geqslant 0$.

Two states s and t are *silently bisimilar*, denoted $s \sim_s t$, if there exists a silent bisimulation R such that $(s, t) \in R$. □

The condition ⟨lab⟩ comes from bisimulations defined on Kripke structures. It makes sure that related states satisfy the same atomic propositions. This is a condition coming from stuttering equivalence. The conditions ⟨term⟩ and ⟨tran⟩ are called the *termination* and the *transfer* condition respectively. Like in the case of branching bisimulation, they require that successful termination and action execution are simulated after a sequence of internal steps. The condition ⟨div⟩ is the *divergence* condition. It says that if a state is related to every state on some infinite execution, then this divergence must be simulated by a non-empty execution sequence of which the final state is related to some state on the diverging path. Note that we treat divergence only in this specific case. However, as we will show later, it turns out that this is not a restriction and that divergence is properly simulated in every case. We localized the requirement only to make bisimulation relations smaller since that is convenient in applications.

2.3 Equivalence proof

In this section we prove that $\sim_s$ is an equivalence relation. The usual way of proving that a bisimulation relation is transitive is to show that the composition of two bisimulation relations is again a bisimulation relation. However, this method fails here because the divergence condition is, in general, non-compositional due to the requirement that *every* state on the divergent execution from s must be in relation with t. We solve the problem by replacing the condition ⟨div⟩ of Definition 2.2.1 by a technically more convenient one (a "transitive" one). The new condition appears stronger at first, but we prove that, in combination with ⟨lab⟩, ⟨term⟩ and ⟨tran⟩, it induces the same notion of silent bisimulation. The reason why we did not use it in Definition 2.2.1 is that it is very complex; we prefer to use ⟨div⟩ in other applications.

We presuppose a doubly-labeled transition system $(S, \rightarrow, \downarrow, \ell) \in \mathcal{T}_{\mathsf{A},\Pi}$ and define the new divergence condition by:

⟨div'⟩ if there is an infinite sequence $s_0, s_1, s_2, \ldots \in S$ such that $s_0 = s$ and $s_0 \rightarrow^+ s_1 \rightarrow^+ s_2 \rightarrow^+ \cdots$, then there exists an infinite sequence $t_0, t_1, t_2, \ldots \in S$ and a mapping $\phi : \mathbb{N} \rightarrow \mathbb{N}$ such that $t_0 = t$, $t_0 \rightarrow^+ t_1 \rightarrow^+ t_2 \rightarrow^+ \cdots$ and $(s_{\phi(i)}, t_i) \in R$ for all $i \geqslant 0$.

To prove that the silent bisimulation coincides with the bisimulation defined using $\langle \mathsf{div'} \rangle$ we need to prove some lemmas first.

The following lemma shows that bisimilar states always reach bisimilar states.

Lemma 2.3.1 Let R be a binary relation on S that satisfies $\langle \mathsf{lab} \rangle$ and $\langle \mathsf{tran} \rangle$. If $(s,t) \in R$ and $s \twoheadrightarrow s'$, then there is a state t' such that $t \twoheadrightarrow t'$ and $(s',t') \in R$. □

Proof From $s \twoheadrightarrow s'$ we have that there exist $s_0, \ldots, s_n \in S$ such that $s_0 = s$, $s_0 \rightarrow \cdots \rightarrow s_n$ and $s_n = s'$. We construct, inductively on n, a sequence $t_0, \ldots, t_n \in S$ such that $t_0 = t$, $t_0 \twoheadrightarrow \cdots \twoheadrightarrow t_n$ and $(s_n, t_n) \in R$.

For the base case ($n = 0$) we take $t_0 = t$. Suppose $s_0 \rightarrow \cdots \rightarrow s_n \rightarrow s_{n+1}$. By the inductive hypothesis there exist $t_0, \ldots, t_n \in S$ such that $t_0 = t$, $t_0 \twoheadrightarrow \cdots \twoheadrightarrow t_n$ and $(s_n, t_n) \in R$. By $\langle \mathsf{tran} \rangle$ it now follows that there exist $t'_n, t''_n \in S$ such that $t_n \twoheadrightarrow t''_n \xrightarrow{(\tau)} t'_n$, $(s_n, t''_n) \in R$ and $(s_{n+1}, t'_n) \in R$. If $t''_n = t'_n$, then trivially $\ell(t'_n) = \ell(t''_n)$; if $t''_n \xrightarrow{\tau} t'_n$, using $\langle \mathsf{lab} \rangle$, we have $\ell(t'_n) = \ell(s_{n+1}) = \ell(s_n) = \ell(t''_n)$. Therefore, $t_n \twoheadrightarrow t'_n$, and so we take $t_{n+1} = t'_n$. ■

It is clear that $\langle \mathsf{div'} \rangle$ implies $\langle \mathsf{div} \rangle$. The following lemma shows that in combination with $\langle \mathsf{lab} \rangle$ and $\langle \mathsf{tran} \rangle$ the converse also holds.

Lemma 2.3.2 If $R \subseteq S \times S$ satisfies $\langle \mathsf{lab} \rangle$, $\langle \mathsf{tran} \rangle$ and $\langle \mathsf{div} \rangle$, then it also satisfies $\langle \mathsf{div'} \rangle$. □

Proof Suppose that $(s,t) \in R$ and that there exists an infinite sequence $s_0, s_1, s_2, \ldots \in S$ such that $s_0 = s$, $s_0 \rightarrow^+ s_1 \rightarrow^+ s_2 \rightarrow^+ \cdots$ and $(s_i, t) \in R$ for all $i \geqslant 0$. We construct, inductively, an infinite sequence $t_0, t_1, t_2, \ldots \in S$ and a mapping $\phi : \mathbb{N} \rightarrow \mathbb{N}$ such that $t_0 = t$, $t_0 \rightarrow^+ t_1 \rightarrow^+ t_2 \rightarrow^+ \cdots$ and $(s_{\phi(j)}, t_j) \in R$ for all $j \geqslant 0$. For the base case we set $t_0 = t$ and $\phi(0) = 0$. Then it clearly holds that $(s_{\phi(0)}, t_0) \in R$. Suppose we have constructed the sequence and the mapping up to n elements. Then $(s_{\phi(n)}, t_n) \in R$. Let $u = \phi(n)$. Since $s_u \rightarrow^+ s_{u+1} \rightarrow^+ s_{u+2} \rightarrow^+ \cdots$, by the definition of $\rightarrow^+$ it follows that there exist $m_0, m_1, \ldots \geqslant 0$ and $s^j_{u+i} \in P$ for $i \geqslant 0$ and $j = 0, \ldots, m_i$ such that, for all $i \geqslant 0$, $s^0_{u+i} = s_{u+i}$, $s^{m_i}_{u+i} \rightarrow s^0_{u+i+1}$ and, if $m_i > 0$, then $s^j_{u+i} \rightarrow s^{j+1}_{u+i}$ for all $j = 0, \ldots, m_i - 1$.

We distinguish two cases.

(i) Suppose $(s^j_{u+i}, t_n) \in R$ for all $i \geqslant 0$ and $j = 0, \ldots, m_i$.

Then, since R satisfies $\langle \mathsf{div} \rangle$, there exist $t''_n \in S$, $k \geqslant 0$ and $l \in \{0, \ldots, m_k\}$ such that $t_n \rightarrow^+ t''_n$ and $(s^l_{u+k}, t''_n) \in R$. Note that $s^l_{u+k} \twoheadrightarrow$

s_{u+k+1}. By Lemma 2.3.1 there exists a $t'_n \in P$ such that $t''_n \twoheadrightarrow t'_n$ and $(s_{u+k+1}, t'_n) \in R$. Clearly, $t_n \to^+ t'_n$.

(ii) Suppose $(s^j_{u+i}, t_n) \in R$ for all $i = 0, \ldots, k$ and $j = 0, \ldots, m_i$, but $(s', t_n) \notin R$ where with s' we denote s^{l+1}_{u+k} or s^0_{u+k+1} depending if $l < m_k$ or $l = m_k$. Since R satisfies ⟨tran⟩, there exist $t''_n, t'''_n \in S$ such that $t_n \twoheadrightarrow t'''_n \xrightarrow{(\tau)} t''_n$, $(s^l_{u+k}, t'''_n) \in R$ and $(s', t''_n) \in R$. From $(s', t''_n) \in R$, we obtain $\ell(t''_n) = \ell(s') = \ell(s_{u+k}) = \ell(t'''_n)$. Because $(s', t_n) \notin R$ we have that either $t_n \neq t'''_n$ or $t'''_n \neq t''_n$. Therefore $t_n \to^+ t''_n$. Note that $s' \twoheadrightarrow s_{u+k+1}$. By Lemma 2.3.1 there exists a $t'_n \in P$ such that $t''_n \twoheadrightarrow t'_n$ and $(s_{u+k+1}, t'_n) \in R$. Clearly, $t_n \to^+ t'_n$.

In both cases we now take $t_{n+1} = t'_n$ and $\phi(n+1) = u + k + 1$. ■

The following now easily follows from Lemma 2.3.2.

Corollary 2.3.3 Let R be symmetric binary relation on S. Then R is a silent bisimulation iff it satisfies the conditions ⟨lab⟩, ⟨term⟩, ⟨tran⟩ and ⟨div'⟩. □

To prove that $\sim_s$ is an equivalence relation we also need to show that the conditions ⟨lab⟩, ⟨term⟩, ⟨tran⟩, ⟨div'⟩ are preserved under union.

Lemma 2.3.4 Let R_i for $i \in I$ be binary relations on S and let con $\in$ {lab, term, tran, div'}. If all relations R_i satisfy ⟨con⟩, then so does their union $R = \bigcup_{i \in I} R_i$. □

Proof Suppose that R_i satisfies ⟨lab⟩ for all $i \in I$. To prove that R also satisfies ⟨lab⟩, suppose that $(s, t) \in R$. Then $(s, t) \in R_i$ for some $i \in I$. Since R_i satisfies ⟨lab⟩, it follows that $\ell(s) = \ell(t)$.

Suppose that R_i satisfies ⟨term⟩ for all $i \in I$. Suppose that $(s, t) \in R$ and that $s\downarrow$. From $(s, t) \in R$ it follows that $(s, t) \in R_i$ for some $i \in I$. Since R_i satisfies ⟨term⟩, there exists a $t' \in S$ such that $t \twoheadrightarrow t'$, $t'\downarrow$ and $(s', t') \in R_i$. Hence $(s', t') \in R$.

Suppose now that R_i satisfies ⟨tran⟩ for all $i \in I$. Suppose $(s, t) \in R$ and $s \xrightarrow{a} s'$ for some $a \in \mathsf{A}$ and $s' \in S$. As before, $(s, t) \in R$ implies that $(s, t) \in R_i$ for some $i \in I$. Using that R_i satisfies ⟨tran⟩, we obtain that there exist $t', t'' \in S$ such that $t \twoheadrightarrow t'' \xrightarrow{(a)} t'$, $(s, t'') \in R_i$ and $(s', t') \in R_i$, and hence $(s, t'') \in R$ and $(s', t') \in R$.

Finally, suppose that R_i satisfies ⟨div'⟩ for all $i \in I$. To prove that R also satisfies ⟨div'⟩, suppose that $(s, t) \in R$ and that there is an infinite sequence of states $s_0, s_1, s_2, \ldots \in S$ such that $s_0 = s$ and $s_0 \to^+ s_1 \to^+ s_2 \to^+ \cdots$. From $(s,t) \in R$ it follows that $(s,t) \in R_i$ for some $i \in I$. By ⟨div'⟩ there exists an infinite sequence of states $t_0, t_1, t_2, \ldots \in S$ and a mapping $\phi : \mathbb{N} \to \mathbb{N}$ such that $t_0 = t$, $t_0 \to^+ t_1 \to^+ t_2 \to^+ \cdots$ and $(s_{\phi(k)}, t_k) \in R_i$ for all $k \geqslant 0$. From the latter it follows that $(s_{\phi(k)}, t_k) \in R$ for all $k \geqslant 0$. ■

Since $\sim_s$ coincides with the union of all silent bisimulations, the following is a direct consequence of Lemmas 2.3.4 and 2.3.3.

Corollary 2.3.5 The relation $\sim_s$ is a silent bisimulation. □

Note that we could have proved Corollary 2.3.5 also by showing that the union of silent bisimulations is again a silent bisimulation. However, the proof of this is more complicated than the one of Lemma 2.3.4 due to the nature of ⟨div⟩.

The following lemma shows that the composition of two silent bisimulations is again a silent bisimulation. This property is crucial for the transitivity proof.

Lemma 2.3.6 Let R_1 and R_2 be binary relations on S and let $R = R_1 \circ R_2$ be their composition. Then

(a) if R_1 and R_2 satisfy ⟨lab⟩, ⟨term⟩ and ⟨tran⟩, then so does R,

(b) if R_1 and R_2 satisfy ⟨div'⟩, then so does R. □

Proof (a) Suppose R_1 and R_2 satisfy ⟨lab⟩. To prove that R also satisfies ⟨lab⟩ suppose $(s,u) \in R$. Then there exists a state t such that $(s,t) \in R_1$ and $(t,u) \in R_2$. Since R_1 satisfies ⟨lab⟩, $\ell(s) = \ell(t)$. Since R_2 satisfies ⟨lab⟩, $\ell(t) = \ell(u)$. Thus, $\ell(s) = \ell(u)$.

Suppose R_1 and R_2 satisfy ⟨term⟩ and ⟨tran⟩. Suppose $(s,u) \in R$. Then there is a $t \in S$ such that $(s,t) \in R_1$ and $(t,u) \in R_2$.

Suppose $s{\downarrow}$. Because R_1 satisfies ⟨term⟩, there exists a $t' \in S$ such that $t \twoheadrightarrow t'$, $t'{\downarrow}$ and $(s,t') \in R_1$. Because R_2 satisfies ⟨lab⟩ and ⟨tran⟩, by Lemma 2.3.1 there is a $u'' \in S$ such that $u \twoheadrightarrow u''$ and $(t', u'') \in R_2$. Since R_2 also satisfies ⟨term⟩, there exists a $u' \in S$ such that $u'' \twoheadrightarrow u'$, $u'{\downarrow}$ and $(t', u') \in R_2$. From $(s,t') \in R_1$ and $(t',u') \in R_2$ we obtain $(s,u') \in R$.

Suppose $s \xrightarrow{a} s'$. Since R_1 satisfies ⟨tran⟩, there exist states t' and t'' such that $t \twoheadrightarrow t'' \xrightarrow{(a)} t'$, $(s,t'') \in R_1$ and $(s',t') \in R_1$. Since R_2 satisfies ⟨tran⟩, by Lemma 2.3.1 there is a state u'' such that $u \twoheadrightarrow u''$ and $(t'', u'') \in R_2$. We now distinguish two cases:

(i) Suppose that $a = \tau$ and $t'' = t'$. Then $u \twoheadrightarrow u'' \xrightarrow{(a)} u''$. From $(s, t'') \in R_1$ and $(t'', u'') \in R_2$ it follows that $(s, u'') \in R$, and from $(s', t') \in R_1$ and $(t', u'') \in R_2$ it follows that $(s', u'') \in R$.

(ii) Suppose that $t'' \xrightarrow{a} t'$. Then there exist states u''' and u' such that $u'' \twoheadrightarrow u''' \xrightarrow{(a)} u'$, $(t'', u''') \in R_2$ and $(t', u') \in R_2$. So, $u \twoheadrightarrow u''' \xrightarrow{(a)} u'$. From $(s, t'') \in R_1$ and $(t'', u''') \in R_2$ it follows that $(s, u''') \in R$. From $(s', t') \in R_1$ and $(t', u') \in R_2$ it follows that $(s', u') \in R$.

(b) Suppose R_1 and R_2 satisfy ⟨div'⟩. To prove that R also satisfies ⟨div'⟩ suppose $(s, u) \in R$ and that there is an infinite sequence of states $s_0, s_1, s_2, \ldots \in S$ such that $s_0 = s$ and $s_0 \to^+ s_1 \to^+ s_2 \to^+ \cdots$. As before, there is a $t \in S$ such that $(s, t) \in R_1$ and $(t, u) \in R_2$. Since R_1 satisfies ⟨div'⟩, there exists $t_0, t_1, t_2, \ldots \in S$ and a mapping $\phi_1 : \mathbb{N} \to \mathbb{N}$ such that $t_0 = t$, $t_0 \to^+ t_1 \to^+ t_2 \to^+ \cdots$ and $(s_{\phi_1(j)}, t_j) \in R_1$ for all $j \geqslant 0$. Since R_2 satisfies ⟨div'⟩, there exists $u_0, u_1, u_2, \ldots \in S$ and a mapping $\phi_2 : \mathbb{N} \to \mathbb{N}$ such that $u_0 = u$, $u_0 \to^+ u_1 \to^+ u_2 \to^+ \cdots$ and $(t_{\phi_2(k)}, u_k) \in R_2$ for all $k \geqslant 0$. Clearly, $(s_{\phi_1(\phi_2(k))}, u_k) \in R$. ■

Now we can prove the following theorem.

Theorem 2.3.7 The relation $\sim_s$ is an equivalence relation. □

Proof The binary relation $\{(s, s) \mid s \in S\}$, i.e. the diagonal on S, is a symmetric relation that clearly satisfies the conditions ⟨lab⟩, ⟨term⟩, ⟨tran⟩ and ⟨div'⟩. So, by Corollary 2.3.3, $\sim_s$ is reflexive.

That $\sim_s$ is symmetric follows immediately from the required symmetry of the witnessing relation.

To prove that $\sim_s$ is transitive, suppose $s \sim_s t$ and $t \sim_s u$. Then there exist symmetric binary relations R_1 and R_2 satisfying ⟨lab⟩, ⟨term⟩, ⟨tran⟩ and ⟨div'⟩, and such that $(s, t) \in R_1$ and $(t, u) \in R_2$. The relation $R = (R_1 \circ R_2) \cup (R_2 \circ R_1)$ is clearly symmetric and, by Lemmas 2.3.4 and 2.3.6, also satisfies ⟨lab⟩, ⟨term⟩, ⟨tran⟩ and ⟨div'⟩. Since $(s, u) \in R$, it follows that $s \sim_s u$. ■

2.4 Stuttering closure

In this section we prove that $\sim_s$ satisfies the so called 'stuttering property'. We use this property in the next section, to compare the transfer and divergence conditions that we use with those that appear in the literature.

Definition 2.4.1 (Stuttering property) A binary relation R on S has the *stuttering property* if, whenever $t_0 \to \cdots \to t_n$, $(s, t_0) \in R$ and $(s, t_n) \in R$, then $(s, t_i) \in R$ for all $i = 0, \dots, n$. □

To prove that $\sim_s$ satisfies the stuttering property we first show that every relation $R \subseteq S \times S$ can be extended to a relation $\widehat{R} \subseteq S \times S$, called its stuttering closure, that has the stuttering property. Then, we show that if R is a silent bisimulation, then so is $\widehat{R}$.

Definition 2.4.2 (Stuttering closure) Let R be a binary relation on S. The *stuttering closure* of R, denoted $\widehat{R}$, is defined by

$$\widehat{R} = \{(s,t) \mid \exists \underline{s}, \bar{s}, \underline{t}, \bar{t} \in S : \underline{s} \twoheadrightarrow s \twoheadrightarrow \bar{s},\ \underline{t} \twoheadrightarrow t \twoheadrightarrow \bar{t},\ (\underline{s}, \bar{t}) \in R \text{ and } (\bar{s}, \underline{t}) \in R\} \ .$$

□

Figure 2.1 illustrates the idea of stuttering closure.

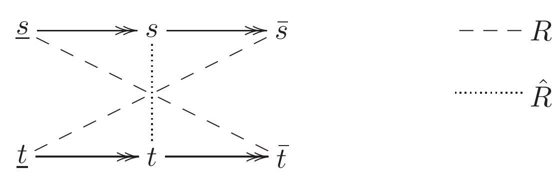


Figure 2.1: Stuttering closure

Clearly $R \subseteq \widehat{R}$. We establish a few basic properties of the stuttering closure.

Lemma 2.4.3 The stuttering closure $\widehat{R}$ of a binary relation R has the stuttering property. □

Proof Suppose that $t_0 \to \cdots \to t_n$, $(s, t_0) \in \widehat{R}$ and $(s, t_n) \in \widehat{R}$. On the one hand, there exist states $\bar{s}$ and $\underline{t_0}$ such that $s \twoheadrightarrow \bar{s}$, $\underline{t_0} \twoheadrightarrow t_0$ and $(\bar{s}, \underline{t_0}) \in R$. On the other hand there exist states $\underline{s}$ and $\bar{t}_n$ such that $\underline{s} \twoheadrightarrow s$, $t_n \twoheadrightarrow \bar{t}_n$ and $(\underline{s}, \bar{t}_n) \in R$. Now, since $\underline{s} \twoheadrightarrow s \twoheadrightarrow \bar{s}$ and $\underline{t_0} \twoheadrightarrow t_0 \twoheadrightarrow t_i \twoheadrightarrow t_n \twoheadrightarrow \bar{t}_n$, it follows that $(s, t_i) \in \widehat{R}$ for all $i = 0, \dots, n$. ■

Lemma 2.4.4 The stuttering closure $\widehat{R}$ of a symmetric relation R is symmetric. □

Proof Suppose $(s,t) \in \widehat{R}$. Then there exist $\underline{s}, \bar{s}, \underline{t}, \bar{t} \in S$ such that $\underline{s} \twoheadrightarrow s \twoheadrightarrow \bar{s}$, $\underline{t} \twoheadrightarrow t \twoheadrightarrow \bar{t}$, $(\underline{s}, \bar{t}) \in R$ and $(\bar{s}, \underline{t}) \in R$. Since R is symmetric, it follows that $(\bar{t}, \underline{s}) \in R$ and $(\underline{t}, \bar{s}) \in R$. Hence $(t,s) \in \widehat{R}$. ■

Lemma 2.4.5 Let $\widehat{R}$ be the stuttering closure of $R \subseteq S \times S$. If $(s,t) \in \widehat{R}$ and R satisfies ⟨lab⟩ and ⟨tran⟩, then there exists $t' \in S$ such that $t \twoheadrightarrow t'$ and $(s,t') \in R$. □

Proof Suppose $(s,t) \in \widehat{R}$. Then there exist $\underline{s}, \bar{s}, \underline{t}, \bar{t} \in S$ such that $\underline{s} \twoheadrightarrow s \twoheadrightarrow \bar{s}$, $\underline{t} \twoheadrightarrow t \twoheadrightarrow \bar{t}$, $(\underline{s}, \bar{t}) \in R$ and $(\bar{s}, \underline{t}) \in R$. From $(\underline{s}, \bar{t}) \in R$ and $\underline{s} \twoheadrightarrow s$ it follows by Lemma 2.3.1 that there exists a $t' \in S$ such that $(t \twoheadrightarrow)\, \bar{t} \twoheadrightarrow t'$ and $(s,t') \in R$. ■

Lemma 2.4.6 Let R be a binary relation on S. If R satisfies ⟨lab⟩, ⟨term⟩ and ⟨tran⟩, then so does its stuttering closure $\widehat{R}$. □

Proof Suppose that $(s,t) \in \widehat{R}$. Then there exist $\underline{s}, \bar{t} \in S$ such that $\underline{s} \twoheadrightarrow s$, $t \twoheadrightarrow \bar{t}$ and $(\underline{s}, \bar{t}) \in R$. From $\underline{s} \twoheadrightarrow s$ and $t \twoheadrightarrow \bar{t}$ we have $\ell(\underline{s}) = \ell(s)$ and $\ell(t) = \ell(\bar{t})$. Since R satisfies ⟨lab⟩, from $(\underline{s}, \bar{t}) \in R$ it follows that $\ell(\underline{s}) = \ell(\bar{t})$. Thus $\ell(s) = \ell(t)$.

Suppose that $(s,t) \in \widehat{R}$ and $s\downarrow$. Since R satisfies ⟨lab⟩ and ⟨tran⟩, by Lemma 2.4.5 there exists $\tilde{t}$ such that $t \twoheadrightarrow \tilde{t}$ and $(s, \tilde{t}) \in R$. From $s\downarrow$ it follows that there exist states t' such that $(t \twoheadrightarrow)\, \tilde{t} \twoheadrightarrow t'$, $t'\downarrow$ and $(s,t') \in R$. The latter implies $(s,t') \in \widehat{R}$.

Suppose that $(s,t) \in \widehat{R}$ and that $s \xrightarrow{a} s'$ for some $s' \in S$. Then by Lemma 2.4.5 there exists $\tilde{t}$ such that $t \twoheadrightarrow \tilde{t}$ and $(s, \tilde{t}) \in R$. Hence, since $s \xrightarrow{a} s'$, by ⟨tran⟩ we have that there exist states t'' and t' such that $(t \twoheadrightarrow\,)\, \tilde{t} \twoheadrightarrow t'' \xrightarrow{(a)} t'$, $(s,t'') \in R$ and $(s',t') \in R$. Now, $(s,t'') \in R$ and $(s',t') \in R$ respectively imply $(s,t'') \in \widehat{R}$ and $(s',t') \in \widehat{R}$. ■

Lemma 2.4.7 If R satisfies ⟨lab⟩, ⟨tran⟩ and ⟨div'⟩, then $\widehat{R}$ satisfies ⟨div'⟩.□

Proof Suppose that $(s,t) \in \widehat{R}$ and that there exists an infinite sequence $s_0, s_1, s_2, \ldots \in S$ such that $s_0 = s$ and $s_0 \xrightarrow{+} s_1 \xrightarrow{+} s_2 \xrightarrow{+} \cdots$. By Lemma 2.4.5 there exists a $t'' \in S$ such that $t \twoheadrightarrow t''$ and $(s,t'') \in R$. Since R satisfies the condition ⟨div'⟩, there exists an infinite sequence $t_0, t_1, t_2, \ldots \in S$ and a mapping $\phi : \mathbb{N} \to \mathbb{N}$ such that $t_0 = t''$, $t_0 \xrightarrow{+} t_1 \xrightarrow{+} t_2 \xrightarrow{+} \cdots$ and $(s_{\phi(i)}, t_i) \in R$ for all $i \geqslant 0$. From $t \twoheadrightarrow t''$ and $(t'' =) t_0 \xrightarrow{+} t_1$ it follows that $t \xrightarrow{+} t_1$. Since $(s_{\phi(i)}, t_i) \in R$, we have $(s_{\phi(i)}, t_i) \in \widehat{R}$. ■

We can now prove the main theorem.

Theorem 2.4.8 The relation $\sim_s$ has the stuttering property. □

Proof By Corollaries 2.3.3 and 2.3.5, $\sim_s$ satisfies the conditions ⟨lab⟩, ⟨term⟩, ⟨tran⟩ and ⟨div'⟩. By Lemma 2.4.6, its stuttering closure $\widehat{\sim_s}$ also satisfies ⟨lab⟩, ⟨term⟩ and ⟨tran⟩. By Lemma 2.4.7, $\widehat{\sim_s}$ also satisfies ⟨div'⟩. By Lemma 2.4.4, it is symmetric. So, $\widehat{\sim_s} \subseteq \sim_s$. By definition, $\sim_s \subseteq \widehat{\sim_s}$, so we obtain $\sim_s = \widehat{\sim_s}$. It follows by Lemma 2.4.3 that $\sim_s$ has the stuttering property. ■

2.5 Alternative definitions

In this section we present some other transfer and divergence conditions and show that they lead to the same notion of silent bisimilarity. Some of the conditions presented are obtained from conditions used in the literature to define equivalence relations similar to silent bisimilarity.

Previously we have seen that a silent bisimulation could have also been defined with the divergence condition ⟨div'⟩ instead of ⟨div⟩. Since ⟨div'⟩ implies ⟨div⟩, we can define a silent bisimulation using also an *interpolant* of ⟨div⟩ and ⟨div'⟩, i.e. using any condition that is implied by ⟨div'⟩ and implies ⟨div⟩. For example, one such condition is:

⟨div''⟩ if there exists an infinite sequence $s_0, s_1, s_2, \ldots \in S$ such that $s_0 = s$ and $s_0 \to s_1 \to s_2 \to \cdots$, then there exists a $t' \in S$ such that $t \xrightarrow{+} t'$ and $(s_k, t') \in R$ for some $k \geqslant 0$.

Note that all, ⟨div⟩, ⟨div'⟩ and ⟨div''⟩, lead to the same relation on the *bisimulation level*, that is they give rise to the same bisimulation relation. We now present some conditions that are equal only on the *level of bisimilarity*, i.e. only for maximal silent bisimulations.

First we give an alternative to the termination and the transfer condition. Theorem 2.4.8 tells us that instead of using ⟨term⟩ and ⟨tran⟩ we could define $\sim_s$ with the following conditions:

⟨$\mathsf{term_{stt}}$⟩ if $s{\downarrow}$, then there exists $t_0, \ldots, t_n \in S$ such that $t_0 = t$, $t_0 \to \cdots \to t_n$, $t_n{\downarrow}$ and $(s, t_i) \in R$ for all $i = 0, \ldots, n$, and

⟨$\mathsf{tran_{stt}}$⟩ if $s \xrightarrow{a} s'$ for some $a \in \mathsf{A}$ and $s' \in S$, then there exist $t_0, \ldots, t_n, t' \in S$ such that $t_0 = t$, $t_0 \to \cdots \to t_n \xrightarrow{(a)} t'$, $(s, t_i) \in R$ for all $i = 0, \ldots, n$ and $(s', t') \in R$.

We now give some other divergence conditions.

Let $s \sim_s''' t$ denote that $(s,t) \in R$ for some binary relation $R \subseteq S \times S$ that satisfies the conditions $\langle \mathsf{lab} \rangle$, $\langle \mathsf{term} \rangle$, $\langle \mathsf{tran} \rangle$ of Definition 2.2.1 and the condition:

$\langle \mathsf{div}''' \rangle$ if there exists an infinite sequence $s_0, s_1, s_2, \ldots \in S$ such that $s_0 = s$, $s_0 \to s_1 \to s_2 \to \cdots$ and $(s_i, t) \in R$ for all $i \geqslant 0$, then there exists an infinite sequence $t_0, t_1, t_2, \ldots \in S$ such that $t_0 \to t_1 \to t_2 \to \cdots$ and $(s_i, t_j) \in R$ for all $i, j \geqslant 0$.

When silent bisimulation is defined using the condition $\langle \mathsf{div}''' \rangle$ and interpreted on a singly-labeled transition system without termination, then it coincides with the notion of *branching bisimulation with explicit divergence* proposed in [51].

We now show that $\sim_s = \sim_s'''$. Since $\langle \mathsf{div}''' \rangle$ implies $\langle \mathsf{div} \rangle$, we have $\sim_s''' \subseteq \sim_s$. To establish $\sim_s \subseteq \sim_s'''$, we use the following lemma.

Lemma 2.5.1 The relation $\sim_s$ satisfies $\langle \mathsf{div}''' \rangle$. □

Proof Suppose $s \sim_s t$ and that there is an infinite sequence $s_0, s_1, s_2, \ldots \in S$ such that $s_0 = s$, $s_0 \to s_1 \to s_2 \to \cdots$ and $s_i \sim_s t$ for all $i \geqslant 0$. By Corollaries 2.3.3 and 2.3.5, $\sim_s$ satisfies $\langle \mathsf{div}' \rangle$, so there exists an infinite sequence of states $t_0, t_1, t_2, \ldots \in S$ and a mapping $\phi : \mathbb{N} \to \mathbb{N}$ such that $t_0 = t$, $t_0 \to^+ t_1 \to^+ t_2 \to^+ \cdots$ and $s_{\phi(j)} \sim_s t_j$ for all $j \geqslant 0$. By Theorem 2.3.7 $\sim_s$ is an equivalence, so, for all $i, j \geqslant 0$, we have $t_j \sim_s s_{\phi(j)} \sim_s t \sim_s s_i$. Let $t_j \to^+ t_{j+1}$, for some $j \geqslant 0$, be witnessed by $t_j^0, \ldots, t_j^{n_j} \in S$ such that $t_j^0 = t_j$ and $t_j^0 \to \cdots \to t_j^{n_j} \to t_{j+1}$. Because $s \sim_s t_j^0$ and $s \sim_s t_{j+1}$, it follows from Theorem 2.4.8 that $s \sim_s t_j^k$ for all $k = 0, \ldots, n_j$. ■

Since $\sim_s = \sim_s'''$, we can replace the condition $\langle \mathsf{div} \rangle$ by any interpolant of $\langle \mathsf{div}''' \rangle$ and $\langle \mathsf{div} \rangle$ and end up with the same equivalence. For instance, we could replace it by one of the following conditions:

$\langle \mathsf{div}_1 \rangle$ if there exists an infinite sequence $s_0, s_1, s_2, \ldots \in S$ such that $s_0 = s$, $s_0 \to s_1 \to s_2 \to \cdots$ and $(s_i, t) \in R$ for all $i \geqslant 0$, then there exist $t_0, \ldots, t_n \in S$ such that $t_0 = t$, $t_0 \to \cdots \to t_n \to t_{n+1}$ and $(s, t_j) \in R$ for all $j = 0, \ldots, n$ and $(s_1, t_{n+1}) \in R$.

$\langle \mathsf{div}_2 \rangle$ if there exist an infinite sequence $s_0, s_1, s_2, \ldots \in S$ such that $s_0 = s$, $s_0 \to s_1 \to s_2 \to \cdots$ and $(s_i, t) \in R$ for all $i \geqslant 0$, then there exists a $t' \in S$ such that $t \to t'$ and $(s_k, t') \in R$ for some $k \geqslant 0$.

$\langle \mathsf{div}_3 \rangle$ if there exist an infinite sequence $s_0, s_1, s_2, \ldots \in S$ such that $s_0 = s$, $s_0 \to s_1 \to s_2 \to \cdots$ and $(s_i, t) \in R$ for all $i \geqslant 0$, then there exists a $t' \in S$ such that $t \to t'$ and $(s_k, t') \in R$ for some $k > 0$.

When silent bisimulation is defined using the conditions $\langle \mathsf{tran_{stt}} \rangle$ and $\langle \mathsf{div}_1 \rangle$ (resp. $\langle \mathsf{tran_{stt}} \rangle$ and $\langle \mathsf{div}_2 \rangle$) and in a setting without termination, it coincides with the notion of *visible bisimulation* (when τ is the only invisible action) of [85] (resp. of [50]). When silent bisimulation is defined using the conditions $\langle \mathsf{tran_{stt}} \rangle$ and $\langle \mathsf{div}_3 \rangle$ and interpreted on a Kripke structure, it coincides with the notion of *stuttering equivalence* from [81].

Let us now explain why we consider $\langle \mathsf{div} \rangle$ to be more compositional then the other divergence conditions mentioned. In Corollary 2.3.3 we have established that a relation defined with $\langle \mathsf{div} \rangle$ and the one defined with $\langle \mathsf{div'} \rangle$ are equivalent on the bisimulation level. The conditions $\langle \mathsf{div'''} \rangle$, $\langle \mathsf{div}_1 \rangle$, $\langle \mathsf{div}_2 \rangle$ and $\langle \mathsf{div}_3 \rangle$ are all "non-transitive", i.e. cannot be composed, and so, when proving transitivity, one must use a condition similar to $\langle \mathsf{div'} \rangle$. Note, however, that all these conditions somehow incorporate stuttering steps. This makes it impossible to prove that the new notion is equivalent to the old one on the bisimulation level, but only on the level of bisimilarity where the stuttering property can be proved first. Clearly, this introduces an unnecessary complication.

Remark 2.5.2 One may argue that transitivity of a bisimilarity can be proved by providing its temporal logic characterization. This is, of course, only true if, in the proof of characterization, transitivity is not used. □

By showing that, when interpreted on a simpler model, silent bisimilarity coincides with the known equivalences, we prove that it preserves the validity of all the corresponding modal logics, most importantly of the logic $\text{CTL}^*_{-\text{X}}$. Recall from the introduction that this is the most important part of our correctness criterion. We finish the chapter by showing that silent bisimilarity also preserves deadlock.

2.6 Deadlock preservation

Intuitively, a state is considered deadlocked if it is not successfully terminated and cannot do an action. A state is said to have deadlock if from it a deadlocked state can be reached.

Definition 2.6.1 (Deadlock) A state s is *deadlocked* if $s\not\downarrow$ and $s \stackrel{a}{\not\to}$ for all $a \in \mathsf{A}$. A state s *has deadlock* if it is deadlocked or if there exist $s_1, \ldots, s_n \in$

S and $a_1, \ldots, a_n \in \mathsf{A}$, for $n \geqslant 1$, such that $s \xrightarrow{a_1} \cdots \xrightarrow{a_n} s_n$ and s_n is deadlocked. $\square$

The following lemma plays the major role in the proof of deadlock preservation.

Lemma 2.6.2 Let R satisfy $\langle \mathsf{lab} \rangle$, $\langle \mathsf{term} \rangle$, $\langle \mathsf{tran} \rangle$ and $\langle \mathsf{div'} \rangle$. If $(s,t) \in R$ and s is deadlocked, then there exists a $t' \in S$ such that $t \twoheadrightarrow t'$ and t' is deadlocked. $\square$

Proof Suppose first that $t_0 = t$ and $t_0 \xrightarrow{\tau} t_1 \xrightarrow{\tau} t_2 \xrightarrow{\tau} \cdots$. for some $t_0, t_1, t_2, \ldots \in S$. Then, because s is deadlocked, it follows easily (by induction and by Lemma 2.3.1) that $(s, t_i) \in R$ for all $i \geqslant 0$. From this, we have $\ell(s) = \ell(t_i)$ for all $i \geqslant 0$, and so $t_0 \to t_1 \to t_2 \to \cdots$. Since R is symmetric and satisfies $\langle \mathsf{div} \rangle$, there exists an $s' \in S$ such that $s \to s'$. This, however, contradicts the fact that s is deadlocked. We conclude that there exists a $t' \in S$ such that $t \twoheadrightarrow t'$ and $t' \overset{\tau}{\not\rightarrow}$. As before, by Lemma 2.3.1, $(t', s) \in R$.

Suppose $t' \xrightarrow{a} t''$ for some $a \in \mathsf{A}$ and $a \neq \tau$. Since R satisfies $\langle \mathsf{tran} \rangle$ and since $s \overset{\tau}{\not\rightarrow}$, there exists an $s' \in S$ such that $s \xrightarrow{a} s'$. This is a contradiction because s is deadlocked. We conclude that $t' \overset{a}{\not\rightarrow}$ for all $a \in \mathsf{A}$.

Suppose $t'{\downarrow}$. Since R satisfies $\langle \mathsf{term} \rangle$ and since $s \overset{\tau}{\not\rightarrow}$, we have $s{\downarrow}$. Contradiction. We conclude that $t' \not\downarrow$.

Since $t' \not\downarrow$ and $t' \overset{a}{\not\rightarrow}$ for all $a \in \mathsf{A}$, it follows that t' is deadlocked. $\blacksquare$

Now we can prove that silent bisimilar states have equal deadlock behavior.

Corollary 2.6.3 If $s \sim_s t$ and s has deadlock, then t has deadlock. $\square$

Proof Let $s \sim_s t$ be witnessed by a silent bisimulation R. Since s has deadlock, either it is deadlocked or there exist $s_1, \ldots, s_n \in S$ and $a_1, \ldots, a_n \in \mathsf{A}$, $n \geqslant 1$, such that $s \xrightarrow{a_1} \cdots \xrightarrow{a_n} s_n$ and s_n is deadlocked.

If s is deadlocked, then by Lemma 2.6.2 it follows directly that t has deadlock. Suppose $s \xrightarrow{a_1} \cdots \xrightarrow{a_n} s_n$ for $n \geqslant 1$ and some $s_1, \ldots, s_n \in S$ and $a_1, \ldots, a_n \in \mathsf{A}$. We prove, by induction on n, that there exist $t_0, t_1, t'_1, \ldots, t_n, t'_n \in S$ such that $t_0 = t$, $t_{i-1} \twoheadrightarrow t'_i \xrightarrow{(a_i)} t_i$ and $(s_i, t_i) \in R$, for $i = 1, \ldots, n$. For $n = 1$, since $(s,t) \in R$, there exist $t', t'' \in S$ such that $t \twoheadrightarrow t'' \xrightarrow{(a_1)} t'$ and $(s_1, t') \in R$. Set $t'_1 = t''$ and $t_1 = t'$. Suppose the statement holds for $1 \leqslant k \leqslant n$ and let $s \xrightarrow{a_1} \cdots \xrightarrow{a_n} s_n \xrightarrow{a_{n+1}} s_{n+1}$. By the inductive hypothesis, there are $t_0, t_1, t'_1, \ldots, t_n, t'_n \in S$ such that $t_0 = t$, $t_{i-1} \twoheadrightarrow t'_i \xrightarrow{(a_i)} t_i$ and $(s_i, t_i) \in R$, for $i = 1, \ldots, n$. From $(s_n, t_n) \in R$ it follows that there exist $t'_n, t''_n \in S$ such that $t_n \twoheadrightarrow t''_n \xrightarrow{(a_{n+1})} t'_n$ and $(s_{n+1}, t'_n) \in R$. Set $t'_{n+1} = t''_n$ and $t_{n+1} = t'_n$.

Now suppose that s_n is deadlocked. Then, because $(s_n, t_n) \in R$, by Lemma 2.6.2, there is a $t' \in S$ such that $t_n \twoheadrightarrow t'$ and t' is deadlocked. Clearly, this implies that t has deadlock. ■

Chapter 3

The language κ

We explained in the introduction that, in general, we want our results to apply to as many modeling languages as possible. It is, of course, not possible to cover all languages. In this chapter we introduce a process specification language called κ (from "core") that, we think, represents the core of most languages. We use this language to establish the congruence property of silent bisimulation in the next chapter.

A typical modeling language (or a process algebra) usually incorporates features such as non-deterministic choice, sequential composition, parallel composition with synchronization, and repetition. To model some aspects of systems more easily it is also common for a language to have constructs to handle data. There are many ways in which these features can be implemented. For example, the communication mechanism can be in CCS [78], CSP [64] or ACP [9] style, data flow can be achieved with variables and assignments (like in Spin [65] for example, or in most imperative programming languages) or some constructs can be parameterized with data (like in μCRL [17]), repetition can be in terms of a repetition operator or obtained with a general recursion, etc. Our idea is to design a language that is general enough to present the applicability of our theory, but not too general, so that the focus is always on the important things.

The language κ is inspired by the engineering specification language χ [90] (more precisely, on its first formalization called χ_σ [20]). The reason we introduce a new language, and not work with χ, is because χ is more application oriented and would hide the full generality of our results. Our language has the standard modeling features, i.e. alternative, sequential and parallel composition. Synchronization of parallel components is in the (most general) ACP style (unlike in χ, where it is CSP style). Data is also incorporated in a very general way, using variables and scoping, and with the possibility to specify (as an outside parameter) how every basic language construct is interpreted in a given data state. Process behavior is affected by data through the concept of guard that, like in χ, originates from the guarded command language [40]. To avoid having to deal with too many technicalities, infinite behavior can be modeled in κ only by means of a repetition operator. We, however, believe that our results hold in the setting with (general) recursion as well. Most features of the modeling languages (χ, SPIN and μCRL) and of process algebras (ACP, CSP, CCS) can be easily mapped to κ.

We now present the syntax and semantics of the untimed version of κ. In Chapter 5 we extend κ with discrete time.

3.1 Syntax and semantics

We presuppose a set of actions A, that includes the special action τ. We also presuppose a set V of *variables*, a set D of *data values*, a set E, that includes D and V, of *data expressions*, and a set of *atomic propositions* Π. We define B to be the set of *boolean expressions* over the set Π and assume that it includes the set of truth values $\{true, false\}$.

Before we give the syntax of κ we introduce the notion of *valuation*. A valuation is usually a semantical notion that assigns values to variables in the global scope. In κ variables can also be declared locally, by the scope operator, and we let a valuation also be attached as a parameter to this operator. Although this does not correspond to common practice (where a valuation is not part of the syntax and is not mixed with the syntactical declaration of local variables [87, 83]), it is to avoid unnecessary additions to syntax and to simplify the presentation of the theory. Note that, again to keep the focus on important things, we take a very abstract view of a valuation, and do not use the more implementation oriented approach with stacks [87, 20, 13].

Definition 3.1.1 (Valuation) A partial mapping $\sigma : V \rightharpoonup D$ with a finite domain (denoted $\mathrm{dom}(\sigma)$) is called a *valuation*. The set of all valuations is denoted Σ. □

That is, a valuation assigns values only to some variables; other variables have no values assigned to them. We assume that any valuation naturally extends to a partial function from the set of data expressions E to D.

We now give the syntax of κ. We presuppose a set Act of *action execution processes*. The set of κ process terms, denoted P, is build over the set of atomic processes (that includes Act) by using the eight operators of the language.

The set P is generated by the following grammar:

$$\begin{aligned} P \ ::= \ & \varepsilon \ \mid \ \delta \ \mid \ \boldsymbol{\alpha} \ \mid \ \mathsf{b} :\rightarrow P \ \mid \ P \cdot P \ \mid \ P + P \ \mid \ P^* \ \mid \ P \parallel P \\ & \mid \ [\![\varsigma \mid P]\!] \ \mid \ \partial_\Xi(P) \ \mid \ \tau_I(P) \ , \end{aligned}$$

where $\boldsymbol{\alpha} \in \mathsf{Act}$, $\mathsf{b} \in \mathsf{B}$, $\varsigma \in \Sigma$, $\Xi \subseteq \mathsf{A} \setminus \{\tau\}$ and $I \subseteq \mathsf{A}$.

The processes ε, δ and $\boldsymbol{\alpha}$ are called *atomic*; the others are *compound*. Let us informally explain their meaning.

Atomic processes

1. The constant δ stands for the *deadlock* process. It cannot execute an action nor terminate successfully.

2. The *empty* process ε cannot do an action either, but it is considered successfully terminated.

3. The *action execution* $\boldsymbol{\alpha} \in \mathsf{Act}$ executes some action, given by the function act defined later, and successfully terminates. In most process algebras the set Act is taken to be the same as the set of actions A. However, in some languages (e.g. in χ and μCRL) the syntactical constructs in Act can be parameterized and it should be distinguished from their instances that appear on the labels in the state space.

Compound processes

1. The *guarded process* $\mathsf{b} :\rightarrow p$ behaves as p when the value of the guard $\mathsf{b} \in \mathsf{B}$ is *true*, and is deadlocked otherwise.

2. The *sequential composition* $p \cdot q$ behaves as p followed by the process q, or as q if p is successfully terminated.

3. The *alternative composition* $p+q$ stands for a non-deterministic choice between p and q.

4. The *repetition operator* * is for the modeling of infinite behavior. The process p^* behaves as p executed zero (successful termination) or more times.

5. The *parallel composition* $p \| q$ executes p and q concurrently in an interleaved fashion. In addition, the two processes can also communicate, i.e. execute two matching actions synchronously.

6. The *scope operator* is used for declarations of local variables. The process $[\![\varsigma \mid p]\!]$ behaves as p in the (local) valuation ς.

7. The *encapsulation operator* ∂_Ξ disables all actions from Ξ. Since $\Xi \subseteq \mathsf{A} \setminus \{\tau\}$, the internal action cannot be disabled.

8. The *hiding operator* τ_I renames all actions from I into the special action τ.

The language κ is very expressive. It allows for modeling of many standard constructs present in other modeling and programming languages. For example, the construct *if b then p else q* is easily represented by $\mathsf{b} :\rightarrow p + \neg\mathsf{b} :\rightarrow q$. Also, the κ process $(\mathsf{b} :\rightarrow p)^* \cdot (\neg\mathsf{b} :\rightarrow \varepsilon)$, corresponds to the *while* b *do p* construct (instead of the ε, some process $\boldsymbol{\alpha}$ can be used to directly express the statement(s) with which the loop is exited).

We now give the formal (operational) semantics of κ. The semantics is given in terms of *configurations* which represent processes together with their context, i.e. processes in a global valuation (see [5] for an alternative approach with the *state* operator). Formally, a configuration is an element of the set $P \times \Sigma$. Due to the presence of ε, a distinction between successful and unsuccessful termination is made.

The formal semantics of κ is parameterized by the following four functions. The first three functions are needed for the correct handling of data, and are modifications similar functions used in [5, 57]; the fourth function is standardly used for modeling communication in ACP style process algebras.

- A partial function $\mathsf{check} : \Pi \times \Sigma \rightharpoonup \{\mathit{true}, \mathit{false}\}$ describes the propositions that are considered true in a given valuation. We assume that check naturally extends to a partial function from B to $\{\mathit{true}, \mathit{false}\}$. The function corresponds to the function *test* from [57] and its main purpose is to give semantics to guards.

  To give an example, if $x = d$, for $x \in V$ and $d \in D$, is a proposition from Π, then we would typically have $\mathsf{check}(x = d, \sigma) = \mathit{true}$ iff $\sigma(x) = d$. In addition, if $(x = d_1 \wedge y = d_2) \in \mathsf{B}$, then $\mathsf{check}(x = d_1 \wedge y = d_2, \sigma) = \mathit{true}$ iff $\mathsf{check}(x = d_1, \sigma) = \mathit{true}$ and $\mathsf{check}(y = d_2, \sigma) = \mathit{true}$.

  It is required that, if two valuations satisfy exactly the same set of propositions, then they must be equal. This is needed to make a link with the semantics of κ and doubly-labeled transition systems. Formally, if $\mathsf{check}(\varphi, \sigma_1) = \mathsf{check}(\varphi, \sigma_2)$ for all $\varphi \in \Pi$, then $\sigma_1 = \sigma_2$. The function check from the above example clearly satisfies this requirement.

- A function $\mathsf{act} : \mathsf{Act} \times \Sigma \to \mathcal{P}(\mathsf{A})$ describes the actions that can be observed when an action execution process is executed in some valuation. A similar function appears in [5] but with the set A used instead of Act and $\mathbb{P}(\mathsf{A})$.

  Most of the time we can take that $\mathsf{act}(\boldsymbol{\alpha}, \sigma) = \{a\}$ for some $a \in \mathsf{A}$. This is because the action a usually denotes the instance of $\boldsymbol{\alpha}$ with the parametric variables replaced by their values in the current valuation σ. A typical example is an assignment process $x := e$ where x is a variable and e is an expression. Then we would define $\mathsf{act}(x := e, \sigma) = \{x := d\}$ where d is the value of e in σ. However, in some languages that incorporate the send/receive style of communication we need a more general version of act. For example, the semantics of the receive process $a?x$ could be to receive any possible value along the channel a and assign it to x. Then we would have to define $\mathsf{act}(a?x, \sigma) = \{a?d \mid d \in D\}$. Note that in this case the possible set of observed actions does not depend on σ.

  For every $a \in \mathsf{A}$, we define the special action execution process $\boldsymbol{a}$ and assume that $\mathsf{act}(\boldsymbol{a}, \sigma) = \{a\}$ for all $\sigma \in \Sigma$.

- A function $\mathsf{eff} : \mathsf{A} \times \Sigma \to \mathcal{P}(\Sigma)$ denotes the resulting valuations when an action is executed. Our definition of eff corresponds to the one of [57]; in [5] this function is defined with the codomain Σ.

  To give a typical example, if a denotes the action that should assign some value $d \in D$ to a variable $x \in V$, then we would have $\mathsf{eff}(a, \sigma) = \{\sigma'\}$ where σ' is the same as σ except that $\sigma'(x) = d$.

  Note that, in general, we allow actions to change the valuation in multiple ways. As called in [57] these actions become *non-deterministic state transformers*. For example this possibility is needed if we want to embed the choice quantification operator [73] from μCRL.

- A partial function $\mathsf{comm} : (\mathsf{A} \setminus \{\tau\}) \times (\mathsf{A} \setminus \{\tau\}) \rightharpoonup \mathsf{A}$ is a *communication* function. If $\mathsf{comm}(a, b) = c$, then this means that the actions a and b can communicate and that the resulting action is c. The internal action cannot communicate with any other action but it can be the result of a communication.

We need to introduce two more notions to deal with local scopes. For a valuation $\sigma \in \Sigma$ and a set $X \subseteq \mathrm{dom}(\sigma)$ we write $\sigma_{/X}$ to denote the restriction of σ to the set X. To correctly override global variables by local variables of the same name, we introduce an operator $\_ \ll \_ : \Sigma \times \Sigma \to \Sigma$ defined by:

$$\mathrm{dom}(\sigma \ll \varsigma) = \mathrm{dom}(\sigma) \cup \mathrm{dom}(\varsigma)$$
$$(\sigma \ll \varsigma)(x) = \begin{cases} \varsigma(x), & \text{if } x \in \mathrm{dom}(\varsigma) \\ \sigma(x), & \text{if } x \in \mathrm{dom}(\sigma) \backslash \mathrm{dom}(\varsigma). \end{cases}$$

The operator $\ll$ binds weaker than $/$.

We can now give the formal (operational) semantics of κ. The operational rules for atomic processes are given in Table 3.1; for the operators they are in Table 3.2. Note that the operational rules give rise to a doubly-labeled transition system $(S, \to, \downarrow, \ell) \in \mathcal{T}_{\mathsf{A},\Pi}$ with

$$S = P \times \Sigma \text{ and } \ell(\langle p, \sigma \rangle) = \{\varphi \in \Pi \mid \mathsf{check}(\varphi, \sigma) = \mathit{true}\}.$$

The requirement we imposed on the function check ensures that $\ell(\langle p, \sigma_1 \rangle) = \ell(\langle q, \sigma_2 \rangle)$ iff $\sigma_1 = \sigma_2$.

$$\frac{}{\langle \varepsilon, \sigma \rangle \downarrow} \langle \mathsf{eps} \rangle \qquad \frac{a \in \mathsf{act}(\boldsymbol{\alpha}, \sigma), \;\; \sigma' \in \mathsf{eff}(a, \sigma)}{\langle \boldsymbol{\alpha}, \sigma \rangle \xrightarrow{a} \langle \varepsilon, \sigma' \rangle} \langle \mathsf{act\text{-}exec} \rangle$$

Table 3.1: Operational semantics for atomic processes

Most of the operational rules in Table 3.2 are either standard or directly correspond to the informal semantics described before. The only two rules that maybe need more explanation are Rule $\langle \mathsf{par\text{-}tran}_2 \rangle$ and Rule $\langle \mathsf{scp\text{-}tran} \rangle$.

Rule $\langle \mathsf{par\text{-}tran}_2 \rangle$ describes how synchronization is performed. The requirement $\sigma' \ll \sigma''_{/\mathrm{dom}(\sigma'') \backslash \mathrm{dom}(\sigma')} = \sigma'' \ll \sigma'_{/\mathrm{dom}(\sigma') \backslash \mathrm{dom}(\sigma'')}$ in the premise is the conflict absence requirement. Its purpose is to establish that σ' and σ'' are not in conflict, i.e. that they assign equal values to same variables. This is needed because the two valuations are to be combined in one valuation. Without conflicts, $\sigma' \ll \sigma''_{/\mathrm{dom}(\sigma'') \backslash \mathrm{dom}(\sigma')}$ (or $\sigma'' \ll \sigma'_{/\mathrm{dom}(\sigma') \backslash \mathrm{dom}(\sigma'')}$) can be seen as the combination of σ' and σ''.

Rule $\langle \mathsf{scp\text{-}tran} \rangle$ has a complicated conclusion. This is because the valuation σ' from the premise must be divided into its 'local' and its 'global' part in the conclusion. The valuation $\sigma'_{/\mathrm{dom}(\varsigma)}$ restricts σ' to the local variables, that is to those in the domain of ς. The valuation $\sigma \ll \sigma'_{/\mathrm{dom}(\sigma') \backslash \mathrm{dom}(\varsigma)}$ leaves the variables that are also in $\mathrm{dom}(\varsigma)$ intact. The other variables are given values by σ'.

$$\frac{\mathsf{check}(\mathsf{b},\sigma) = true,\ \langle p,\sigma\rangle{\downarrow}}{\langle \mathsf{b} :\rightarrow p,\sigma\rangle{\downarrow}}\ \langle\mathsf{grd\text{-}term}\rangle$$

$$\frac{\mathsf{check}(\mathsf{b},\sigma) = true,\ \langle p,\sigma\rangle \xrightarrow{a} \langle p',\sigma'\rangle}{\langle \mathsf{b} :\rightarrow p,\sigma\rangle \xrightarrow{a} \langle p',\sigma'\rangle}\ \langle\mathsf{grd\text{-}tran}\rangle$$

$$\frac{\langle p,\sigma\rangle{\downarrow},\ \langle q,\sigma\rangle{\downarrow}}{\langle p\cdot q,\sigma\rangle{\downarrow}}\ \langle\mathsf{seq\text{-}term}\rangle \qquad \frac{\langle p,\sigma\rangle{\downarrow},\ \langle q,\sigma\rangle \xrightarrow{a} \langle q',\sigma'\rangle}{\langle p\cdot q,\sigma\rangle \xrightarrow{a} \langle q',\sigma'\rangle}\ \langle\mathsf{seq\text{-}tran}_1\rangle$$

$$\frac{\langle p,\sigma\rangle \xrightarrow{a} \langle p',\sigma'\rangle}{\langle p\cdot q,\sigma\rangle \xrightarrow{a} \langle p'\cdot q,\sigma'\rangle}\ \langle\mathsf{seq\text{-}tran}_2\rangle \qquad \frac{\langle p,\sigma\rangle{\downarrow}}{\langle p+q,\sigma\rangle{\downarrow},\ \langle q+p,\sigma\rangle{\downarrow}}\ \langle\mathsf{alt\text{-}term}\rangle$$

$$\frac{\langle p,\sigma\rangle \xrightarrow{a} \langle p',\sigma'\rangle}{\langle p+q,\sigma\rangle \xrightarrow{a} \langle p',\sigma'\rangle,\ \langle q+p,\sigma\rangle \xrightarrow{a} \langle p',\sigma'\rangle}\ \langle\mathsf{alt\text{-}tran}\rangle \qquad \frac{}{\langle p^*,\sigma\rangle{\downarrow}}\ \langle\mathsf{rep\text{-}term}\rangle$$

$$\frac{\langle p,\sigma\rangle \xrightarrow{a} \langle p',\sigma'\rangle}{\langle p^*,\sigma\rangle \xrightarrow{a} \langle p'\cdot p^*,\sigma'\rangle}\ \langle\mathsf{rep\text{-}tran}\rangle \qquad \frac{\langle p,\sigma\rangle{\downarrow},\ \langle q,\sigma\rangle{\downarrow}}{\langle p\parallel q,\sigma\rangle{\downarrow},\ \langle q\parallel p,\sigma\rangle{\downarrow}}\ \langle\mathsf{par\text{-}term}\rangle$$

$$\frac{\langle p,\sigma\rangle \xrightarrow{a} \langle p',\sigma'\rangle}{\langle p\parallel q,\sigma\rangle \xrightarrow{a} \langle p'\parallel q,\sigma'\rangle,\ \langle q\parallel p,\sigma\rangle \xrightarrow{a} \langle q\parallel p',\sigma'\rangle}\ \langle\mathsf{par\text{-}tran}_1\rangle$$

$$\frac{\begin{array}{c}\langle p,\sigma\rangle \xrightarrow{a} \langle p',\sigma'\rangle,\ \langle q,\sigma\rangle \xrightarrow{b} \langle q',\sigma''\rangle,\ \mathsf{comm}(a,b) = c,\\ \sigma' \ll {\sigma''}_{/\mathrm{dom}(\sigma'')\setminus\mathrm{dom}(\sigma')} = \sigma'' \ll {\sigma'}_{/\mathrm{dom}(\sigma')\setminus\mathrm{dom}(\sigma'')} = \sigma'''\end{array}}{\langle p\parallel q,\sigma\rangle \xrightarrow{c} \langle p'\parallel q',\sigma'''\rangle,\ \langle q\parallel p,\sigma\rangle \xrightarrow{c} \langle q'\parallel p',\sigma'''\rangle}\ \langle\mathsf{par\text{-}tran}_2\rangle$$

$$\frac{\langle p,\sigma\ll\varsigma\rangle{\downarrow}}{\langle [\![\varsigma \mid p]\!],\sigma\rangle{\downarrow}}\ \langle\mathsf{scp\text{-}term}\rangle \qquad \frac{\langle p,\sigma\ll\varsigma\rangle \xrightarrow{a} \langle p',\sigma'\rangle,\ X = \mathrm{dom}(\sigma')\setminus\mathrm{dom}(\varsigma)}{\langle [\![\varsigma \mid p]\!],\sigma\rangle \xrightarrow{a} \langle [\![{\sigma'}_{/\mathrm{dom}(\varsigma)} \mid p']\!],\sigma\ll{\sigma'}_{/X}\rangle}\ \langle\mathsf{scp\text{-}tran}\rangle$$

$$\frac{\langle p,\sigma\rangle{\downarrow}}{\langle \partial_\Xi(p),\sigma\rangle{\downarrow}}\ \langle\mathsf{enc\text{-}term}\rangle \qquad \frac{\langle p,\sigma\rangle \xrightarrow{a} \langle p',\sigma'\rangle,\ a\notin\Xi}{\langle \partial_\Xi(p),\sigma\rangle \xrightarrow{a} \langle \partial_\Xi(p'),\sigma'\rangle}\ \langle\mathsf{enc\text{-}tran}\rangle$$

$$\frac{\langle p,\sigma\rangle{\downarrow}}{\langle \tau_I(p),\sigma\rangle{\downarrow}}\ \langle\mathsf{hide\text{-}term}\rangle \qquad \frac{\langle p,\sigma\rangle \xrightarrow{a} \langle p',\sigma'\rangle,\ a\notin I}{\langle \tau_I(p),\sigma\rangle \xrightarrow{a} \langle \tau_I(p'),\sigma'\rangle}\ \langle\mathsf{hide\text{-}tran}_1\rangle$$

$$\frac{\langle p,\sigma\rangle \xrightarrow{a} \langle p',\sigma'\rangle,\ a\in I}{\langle \tau_I(p),\sigma\rangle \xrightarrow{\tau} \langle \tau_I(p'),\sigma'\rangle}\ \langle\mathsf{hide\text{-}tran}_2\rangle$$

Table 3.2: Operational semantics for compound processes

Chapter 4

Silent congruence

In the previous chapter we have shown that a κ process with a valuation generates a doubly-labeled transition system. In Chapter 2 we introduced a notion of silent bisimulation on doubly-labeled transition systems. As we said in the introduction, we are interested in symbolic techniques and so we need a corresponding notion of bisimulation defined directly on process terms. We also want the new notion to be a congruence to allow for compositional manipulation. In this chapter we first lift the definition of silent bisimilarity to the level of κ processes. We show that the new notion is not a congruence and we adapt it, in a step by step manner, to obtain a congruence.

4.1 Silent bisimulation on processes

A natural way to lift the relation $\sim_s$ to κ processes is as follows.

Definition 4.1.1 (Silent bisimulation on processes) Two processes p and q are *silently bisimilar*, denoted $p \sim_s q$, if there exists a silent bisimulation R such that $(\langle p, \sigma\rangle, \langle q, \sigma\rangle) \in R$ for all $\sigma \in \Sigma$. □

Silent bisimulation on processes is an equivalence relation. This easily follows from Lemmas 2.3.4 and 2.3.6 of Chapter 2. We show that it is a congruence relation for guards, and for the scope, the encapsulation, and the hiding operator.

Theorem 4.1.2 For all $p, q \in P$ and all $\mathsf{b} \in \mathsf{B}$, if $p \sim_s q$, then $\mathsf{b} :\rightarrow p \sim_s \mathsf{b} :\rightarrow q$. □

Proof Let R_{pq} be a silent bisimulation such that $(\langle p, \sigma\rangle, \langle q, \sigma\rangle) \in R_{pq}$ for all $\sigma \in \Sigma$. Let

$$\begin{aligned} R \;=\; & \{(\langle \mathsf{b} :\rightarrow p, \sigma\rangle, \langle \mathsf{b} :\rightarrow q, \sigma\rangle) \mid \sigma \in \Sigma\} \\ & \cup \{(\langle \mathsf{b} :\rightarrow p, \sigma\rangle, \langle s, \sigma\rangle) \mid \mathsf{check}(\mathsf{b}, \sigma) = \mathit{true},\ (\langle p, \sigma\rangle, \langle s, \sigma\rangle) \in R_{pq}\} \\ & \cup \{(\langle r, \sigma\rangle, \langle \mathsf{b} :\rightarrow q, \sigma\rangle,) \mid \mathsf{check}(\mathsf{b}, \sigma) = \mathit{true},\ (\langle r, \sigma\rangle, \langle q, \sigma\rangle) \in R_{pq}\}. \end{aligned}$$

We show that R is a silent bisimulation. Note that it is clear from the definition that R satisfies $\langle \mathsf{lab}\rangle$. It is symmetric because R_{pq} is symmetric. We show that it also satisfies $\langle \mathsf{term}\rangle$, $\langle \mathsf{tran}\rangle$ and $\langle \mathsf{div'}\rangle$. By Corollary 2.3.3 this is enough to prove that R is a silent bisimulation.

- We first check the conditions for the pairs in the first set.

  Cond. $\langle \mathsf{term}\rangle$**:** Suppose $\langle \mathsf{b} :\rightarrow p, \sigma\rangle\downarrow$. Rule $\langle \mathsf{grd\text{-}term}\rangle$ is the final rule of any derivation with $\langle \mathsf{b} :\rightarrow p, \sigma\rangle\downarrow$ as conclusion, so it holds that $\mathsf{check}(\mathsf{b}, \sigma) = \mathit{true}$ and $\langle p, \sigma\rangle\downarrow$. Since $(\langle p, \sigma\rangle, \langle q, \sigma\rangle) \in R_{pq}$, there exists a $q' \in P$ such that $\langle q, \sigma\rangle \twoheadrightarrow \langle q', \sigma\rangle$, $\langle q', \sigma\rangle\downarrow$ and $(\langle p, \sigma\rangle, \langle q', \sigma\rangle) \in R_{pq}$. If $q = q'$, then by Rule $\langle \mathsf{grd\text{-}term}\rangle$, we have $\langle \mathsf{b} :\rightarrow q, \sigma\rangle\downarrow$. Otherwise, by Rule $\langle \mathsf{grd\text{-}tran}\rangle$, $\langle \mathsf{b} :\rightarrow q, \sigma\rangle \twoheadrightarrow \langle q', \sigma\rangle$. Since we have $\mathsf{check}(\mathsf{b}, \sigma) = \mathit{true}$ and $(\langle p, \sigma\rangle, \langle q', \sigma\rangle) \in R_{pq}$, according to the definition of R, that $(\langle \mathsf{b} :\rightarrow p, \sigma\rangle, \langle q', \sigma\rangle) \in R$.

  Cond. $\langle \mathsf{tran}\rangle$**:** Suppose $\langle \mathsf{b} :\rightarrow p, \sigma\rangle \xrightarrow{a} \langle t, \sigma'\rangle$ for some $\sigma' \in \Sigma$ and $t \in P$. Since Rule $\langle \mathsf{grd\text{-}tran}\rangle$ must be the final rule of any derivation of this transition as conclusion, it holds that $\mathsf{check}(\mathsf{b}, \sigma) = \mathit{true}$, $\langle p, \sigma\rangle \xrightarrow{a} \langle p', \sigma'\rangle$ and $t = p'$. Since $(\langle p, \sigma\rangle, \langle q, \sigma\rangle) \in R_{pq}$, there exist $q', q'' \in P$ such that $\langle q, \sigma\rangle \twoheadrightarrow \langle q'', \sigma\rangle \xrightarrow{(a)} \langle q', \sigma'\rangle$, $(\langle p, \sigma\rangle, \langle q'', \sigma\rangle) \in R_{pq}$ and $(\langle p', \sigma'\rangle, \langle q', \sigma'\rangle) \in R_{pq}$. From this, by Rule $\langle \mathsf{grd\text{-}tran}\rangle$, we have $\langle \mathsf{b} :\rightarrow q, \sigma\rangle \twoheadrightarrow \langle q'', \sigma\rangle \xrightarrow{(a)} \langle q', \sigma'\rangle$. Since $\mathsf{check}(\mathsf{b}, \sigma) = \mathit{true}$ and $(\langle p, \sigma\rangle, \langle q'', \sigma\rangle) \in R_{pq}$, according to the definition of R, we have $(\langle \mathsf{b} :\rightarrow p, \sigma\rangle, \langle q'', \sigma\rangle) \in R$. Finally, since $(\langle p', \sigma'\rangle, \langle q', \sigma'\rangle) \in R_{pq} \subseteq R$, also $(\langle p', \sigma'\rangle, \langle q', \sigma'\rangle) \in R$.

  Cond. $\langle \mathsf{div'}\rangle$**:** Suppose that there exist $t_0, t_1, t_2, \ldots \in P$, such that $t_0 = \mathsf{b} :\rightarrow p$ and $\langle t_0, \sigma\rangle \rightarrow \langle t_1, \sigma\rangle \rightarrow \langle t_2, \sigma\rangle \rightarrow \cdots$. Rule $\langle \mathsf{grd\text{-}tran}\rangle$ is the only rule in any derivation with $\langle t_0, \sigma\rangle \rightarrow \langle t_1, \sigma\rangle$ as conclusion, and so $\mathsf{check}(\mathsf{b}, \sigma) = \mathit{true}$ and $\langle p, \sigma\rangle \rightarrow \langle t_1, \sigma\rangle \rightarrow \langle t_2, \sigma\rangle \rightarrow \cdots$. Since R_{pq} satisfies $\langle \mathsf{div'}\rangle$, there exist $q_0, q_1, q_2, \ldots \in P$ and a mapping $\phi : \mathbb{N} \rightarrow \mathbb{N}$, such that $q_0 = q$, $\langle q_0, \sigma\rangle \rightarrow \langle q_1, \sigma\rangle \rightarrow \langle q_2, \sigma\rangle \rightarrow \cdots$ $(t_{\phi(i)}, q_i) \in R_{pq}$ for all $i \geqslant 0$. By Rule $\langle \mathsf{grd\text{-}tran}\rangle$,

$\langle \mathsf{b} :\to q_0, \sigma\rangle \to \langle q_1, \sigma\rangle \to \langle q_2, \sigma\rangle \to \cdots$. Since $(t_{\phi(i)}, q_i) \in R_{pq}$, according to the definition of R, we have $(t_{\phi(0)}, \mathsf{b} :\to q) \in R$ and $(t_{\phi(i)}, q_i) \in R$ for $i > 0$.

- We now check the conditions for the pairs in the second set.

Cond. $\langle\mathsf{term}\rangle$**:** Suppose $\langle \mathsf{b} :\to p, \sigma\rangle{\downarrow}$. Since Rule $\langle\mathsf{grd\text{-}term}\rangle$ is the final rule of any derivation with $\langle \mathsf{b} :\to p, \sigma\rangle{\downarrow}$ as conclusion, and since $\mathsf{check}(\mathsf{b}, \sigma) = \mathit{true}$, we have $\langle p, \sigma\rangle{\downarrow}$. Since $(\langle p, \sigma\rangle, \langle s, \sigma\rangle) \in R_{pq}$, there exists an $s' \in P$ such that $\langle s, \sigma\rangle \twoheadrightarrow \langle s', \sigma\rangle$, $\langle s', \sigma\rangle{\downarrow}$ and $(\langle p, \sigma\rangle, \langle s', \sigma\rangle) \in R_{pq}$. According to the definition of R, we have $(\langle \mathsf{b} :\to p, \sigma\rangle, \langle s', \sigma\rangle) \in R$.

Cond. $\langle\mathsf{tran}\rangle$**:** Suppose $\langle \mathsf{b} :\to p, \sigma\rangle \xrightarrow{a} \langle t, \sigma'\rangle$ for some $\sigma' \in \Sigma$ and $t \in P$. Since Rule $\langle\mathsf{grd\text{-}tran}\rangle$ must be the final rule of any derivation of this transition as conclusion and since $\mathsf{check}(\mathsf{b}, \sigma) = \mathit{true}$, $\langle p, \sigma\rangle \xrightarrow{a} \langle p', \sigma'\rangle$ and $t = p'$. Since $(\langle p, \sigma\rangle, \langle s, \sigma\rangle) \in R_{pq}$, there exist $s', s'' \in P$ such that $\langle s, \sigma\rangle \twoheadrightarrow \langle s'', \sigma\rangle \xrightarrow{(a)} \langle s', \sigma'\rangle$, $(\langle p, \sigma\rangle, \langle s'', \sigma\rangle) \in R_{pq}$ and $(\langle p', \sigma'\rangle, \langle s', \sigma'\rangle) \in R_{pq}$. Now, because $(\langle p, \sigma\rangle, \langle s'', \sigma\rangle) \in R_{pq}$, according to the definition of R, we have $(\langle \mathsf{b} :\to p, \sigma\rangle, \langle s'', \sigma\rangle) \in R$. Also, since $(\langle p', \sigma'\rangle, \langle s', \sigma'\rangle) \in R_{pq} \subseteq R$, $(\langle p', \sigma'\rangle, \langle s', \sigma'\rangle) \in R$.

Cond. $\langle\mathsf{div'}\rangle$**:** Suppose that there exist $t_0, t_1, t_2, \ldots \in P$, such that $t_0 = \mathsf{b} :\to p$ and $\langle t_0, \sigma\rangle \to \langle t_1, \sigma\rangle \to \langle t_2, \sigma\rangle \to \cdots$. Rule $\langle\mathsf{grd\text{-}tran}\rangle$ is the only rule in any derivation with $\langle t_0, \sigma\rangle \to \langle t_1, \sigma\rangle$ as conclusion and, since $\mathsf{check}(\mathsf{b}, \sigma) = \mathit{true}$, we have $\langle p, \sigma\rangle \to \langle t_1, \sigma\rangle \to \langle t_2, \sigma\rangle \to \cdots$. Since R_{pq} satisfies $\langle\mathsf{div'}\rangle$, there exist $s_0, s_1, s_2, \ldots \in P$ and a mapping $\phi : \mathbb{N} \to \mathbb{N}$, such that $s_0 = s$, $\langle s_0, \sigma\rangle \to \langle s_1, \sigma\rangle \to \langle s_2, \sigma\rangle \to \cdots$ $(\langle t_{\phi(i)}, \sigma\rangle, \langle s_i, \sigma\rangle) \in R_{pq}$ for all $i \geqslant 0$. By Rule $\langle\mathsf{grd\text{-}tran}\rangle$, $\langle \mathsf{b} :\to s_0, \sigma\rangle \to \langle s_1, \sigma\rangle \to \langle s_2, \sigma\rangle \to \cdots$. Since $(\langle t_{\phi(i)}, \sigma\rangle, \langle s_i, \sigma\rangle) \in R_{pq}$, according to the definition of R, we have $(\langle t_{\phi(0)}, \sigma\rangle, \langle \mathsf{b} :\to s, \sigma\rangle) \in R$ and $(\langle t_{\phi(i)}, \sigma\rangle, \langle s_i, \sigma\rangle) \in R$ for $i > 0$.

- Finally, we now check the conditions for the pairs in the third set.

Cond. $\langle\mathsf{term}\rangle$**:** Suppose $\langle r, \sigma\rangle{\downarrow}$. Since $(\langle r, \sigma\rangle, \langle q, \sigma\rangle) \in R_{pq}$, there is a $q' \in P$ such that $\langle q, \sigma\rangle \twoheadrightarrow \langle q', \sigma\rangle$, $\langle q', \sigma\rangle{\downarrow}$ and $(\langle r, \sigma\rangle, \langle q', \sigma\rangle) \in R_{pq}$. Since $\mathsf{check}(\mathsf{b}, \sigma) = \mathit{true}$, by Rule $\langle\mathsf{grd\text{-}term}\rangle$, $\langle \mathsf{b} :\to q, \sigma\rangle \twoheadrightarrow \langle q', \sigma\rangle$. Since $\mathsf{check}(\mathsf{b}, \sigma) = \mathit{true}$ and $(\langle r, \sigma\rangle, \langle q', \sigma\rangle) \in R_{pq}$, according to the definition of R, $(\langle r, \sigma\rangle, \langle q', \sigma\rangle) \in R$.

Cond. $\langle\mathsf{tran}\rangle$**:** Suppose $\langle r,\sigma\rangle \xrightarrow{a} \langle r',\sigma'\rangle$ for some $\sigma' \in \Sigma$ and $r' \in P$. Since $(\langle r,\sigma\rangle,\langle q,\sigma\rangle) \in R_{pq}$, it follows that there exist $q',q'' \in P$ such that $\langle q,\sigma\rangle \twoheadrightarrow \langle q'',\sigma\rangle \xrightarrow{(a)} \langle q',\sigma'\rangle$, $(\langle r,\sigma\rangle,\langle q'',\sigma\rangle) \in R_{pq}$ and $(\langle r',\sigma'\rangle,\langle q',\sigma'\rangle) \in R_{pq}$. Since $\mathsf{check}(\mathsf{b},\sigma) = \mathit{true}$, it follows by Rule $\langle\mathsf{grd\text{-}tran}\rangle$ that $\langle \mathsf{b} :\rightarrow q,\sigma\rangle \twoheadrightarrow \langle q'',\sigma\rangle \xrightarrow{(a)} \langle q',\sigma'\rangle$. Since $(\langle r,\sigma\rangle,\langle q'',\sigma\rangle) \in R_{pq}$ and $(\langle r',\sigma'\rangle,\langle q',\sigma'\rangle) \in R_{pq}$, according to the definition of R, $(\langle r,\sigma\rangle,\langle q'',\sigma\rangle) \in R$ and $(\langle r',\sigma'\rangle,\langle q',\sigma'\rangle) \in R$.

Cond. $\langle\mathsf{div'}\rangle$**:** Suppose that there exist $r_0,r_1,r_2,\ldots \in P$, such that $r_0 = r$ and $\langle r_0,\sigma\rangle \rightarrow \langle r_1,\sigma\rangle \rightarrow \langle r_2,\sigma\rangle \rightarrow \cdots$. Since R_{pq} satisfies $\langle\mathsf{div'}\rangle$, there exist $q_0,q_1,q_2,\ldots \in P$ and a mapping $\phi : \mathbb{N} \rightarrow \mathbb{N}$, such that $q_0 = q$, $\langle q_0,\sigma\rangle \rightarrow \langle q_1,\sigma\rangle \rightarrow \langle q_2,\sigma\rangle \rightarrow \cdots$ and $(r_{\phi(i)},q_i) \in R_{pq}$ for all $i \geqslant 0$. Because $\mathsf{check}(\mathsf{b},\sigma) = \mathit{true}$, by Rule $\langle\mathsf{grd\text{-}tran}\rangle$, we have $\langle \mathsf{b} :\rightarrow q_0,\sigma\rangle \rightarrow \langle q_1,\sigma\rangle \rightarrow \langle q_2,\sigma\rangle \rightarrow \cdots$. Since $(\langle r_{\phi(i)},\sigma\rangle,\langle q_i,\sigma\rangle) \in R_{pq}$, according to the definition of R, $(\langle r_{\phi(0)},\sigma\rangle,\langle \mathsf{b} :\rightarrow q,\sigma\rangle) \in R$ and $(\langle t_{\phi(i)},\sigma\rangle,\langle q_i,\sigma\rangle) \in R$ for $i > 0$.■

The following lemmas from the core of the proof that silent bisimulation on processes is a congruence for the scope, the encapsulation, and the hiding operator. We state these results as lemmas because we will use them later in the text.

Lemma 4.1.3 If R_S is a silent bisimulation, then

$$R = \{(\langle[\![\varsigma \mid p]\!],\sigma\rangle,\langle[\![\varsigma \mid q]\!],\sigma\rangle) \mid \varsigma,\sigma \in \Sigma, (\langle p,\sigma \ll \varsigma\rangle,\langle q,\sigma \ll \varsigma\rangle) \in R_S\}$$

is also a silent bisimulation. □

Proof We show that R satisfies the conditions of Definition 2.2.1. It is symmetric because R_S is symmetric. That it satisfies $\langle\mathsf{lab}\rangle$ follows directly from its definition. We show that it also satisfies $\langle\mathsf{term}\rangle$, $\langle\mathsf{tran}\rangle$ and $\langle\mathsf{div}\rangle$.

Cond. $\langle\mathsf{term}\rangle$**:** Suppose $\langle[\![\varsigma \mid p]\!],\sigma\rangle{\downarrow}$. Since Rule $\langle\mathsf{scp\text{-}term}\rangle$ is the final rule of any derivation with $\langle[\![\varsigma \mid p]\!],\sigma\rangle{\downarrow}$ as conclusion, it holds that $\langle p,\sigma \ll \varsigma\rangle{\downarrow}$. Since $(\langle p,\sigma \ll \varsigma\rangle,\langle q,\sigma \ll \varsigma\rangle) \in R_S$, there exists an $s' \in P$ such that $\langle q,\sigma \ll \varsigma\rangle \twoheadrightarrow \langle q',\sigma \ll \varsigma\rangle$, $\langle q',\sigma \ll \varsigma\rangle{\downarrow}$, and $(\langle p,\sigma \ll \varsigma\rangle,\langle q',\sigma \ll \varsigma\rangle) \in R_S$. Hence, by Rule $\langle\mathsf{scp\text{-}term}\rangle$, $\langle[\![\varsigma \mid q']\!],\sigma\rangle{\downarrow}$. Using that $(\sigma \ll \varsigma)_{/\mathrm{dom}(\varsigma)} = \varsigma$ and $\sigma \ll (\sigma \ll \varsigma)_{/(\mathrm{dom}(\sigma)\cup\mathrm{dom}(\varsigma))\setminus\mathrm{dom}(\varsigma)} = \sigma \ll (\sigma \ll \varsigma)_{/\mathrm{dom}(\sigma)\setminus\mathrm{dom}(\varsigma)} = \sigma \ll \sigma_{/\mathrm{dom}(\sigma)\setminus\mathrm{dom}(\varsigma)} = \sigma$, by Rule $\langle\mathsf{scp\text{-}tran}\rangle$, we obtain $\langle[\![\varsigma \mid q]\!],\sigma\rangle \twoheadrightarrow \langle[\![\varsigma \mid q']\!],\sigma\rangle$. Finally, since $(\langle p,\sigma \ll \varsigma\rangle,\langle q',\sigma \ll \varsigma\rangle) \in R_S$, according to the definition of R, $([\![\varsigma \mid p]\!],[\![\varsigma \mid q']\!]) \in R$.

Cond. $\langle\mathsf{tran}\rangle$**:** Suppose $\langle [\![\varsigma \mid p]\!], \sigma\rangle \xrightarrow{a} \langle r, \sigma'\rangle$ for some $\sigma' \in \Sigma$ and $r \in P$. Since Rule $\langle\mathsf{scp\text{-}tran}\rangle$ is the final rule of any derivation with this transition as conclusion, we have that there exist $p' \in P$ and $\sigma'' \in \Sigma$ such that $\langle p, \sigma \ll \varsigma\rangle \xrightarrow{a} \langle p', \sigma''\rangle$, $\sigma' = \sigma \ll \sigma''_{/\mathrm{dom}(\sigma)\setminus\mathrm{dom}(\varsigma)}$ and $r = [\![\varsigma' \mid p']\!]$ for $\varsigma' = \sigma''_{/\mathrm{dom}(\varsigma)}$. Since $(\langle p, \sigma \ll \varsigma\rangle, \langle q, \sigma \ll \varsigma\rangle) \in R_S$, there exist $q', q'' \in P$ such that $\langle q, \sigma \ll \varsigma\rangle \twoheadrightarrow \langle q'', \sigma \ll \varsigma\rangle \xrightarrow{(a)} \langle q', \sigma''\rangle$, and also $(\langle p, \sigma \ll \varsigma\rangle, \langle q'', \sigma \ll \varsigma\rangle) \in R_S$ and $(\langle p', \sigma''\rangle, \langle q', \sigma''\rangle) \in R_S$. Using again that $(\sigma \ll \varsigma)_{/\mathrm{dom}(\varsigma)} = \varsigma$ and $\sigma \ll (\sigma \ll \varsigma)_{/(\mathrm{dom}(\sigma)\cup\mathrm{dom}(\varsigma))\setminus\mathrm{dom}(\varsigma)} = \sigma$, by Rule $\langle\mathsf{scp\text{-}tran}\rangle$, we have $\langle [\![\varsigma \mid q]\!], \sigma\rangle \twoheadrightarrow \langle [\![\varsigma \mid q'']\!], \sigma\rangle \xrightarrow{(a)} \langle [\![\varsigma' \mid q']\!], \sigma''\rangle$. We calculate the following: $\sigma' \ll \varsigma' = (\sigma \ll \sigma''_{/\mathrm{dom}(\sigma'')\setminus\mathrm{dom}(\varsigma)}) \ll \sigma''_{/\mathrm{dom}(\varsigma)} = \sigma \ll (\sigma''_{/\mathrm{dom}(\sigma'')\setminus\mathrm{dom}(\varsigma)} \ll \sigma''_{/\mathrm{dom}(\varsigma)}) = \sigma \ll \sigma''_{/\mathrm{dom}(\sigma'')} = \sigma''$. Hence, since $(\langle p, \sigma \ll \varsigma\rangle, \langle q'', \sigma \ll \varsigma\rangle) \in R_S$ and $(\langle p', \sigma'\rangle, \langle q', \sigma'\rangle) \in R_S$, according to the definition of R it follows that $(\langle [\![\varsigma \mid p]\!], \sigma\rangle, \langle [\![\varsigma \mid q'']\!], \sigma\rangle) \in R$ and $(\langle [\![\varsigma' \mid p']\!], \sigma''\rangle, \langle [\![\varsigma' \mid q']\!], \sigma''\rangle) \in R$.

Cond. $\langle\mathsf{div}\rangle$**:** Suppose that there exist $\sigma \in \Sigma$ and $r_0, r_1, r_2, \ldots \in P$, such that $r_0 = [\![\varsigma \mid p]\!]$ and $\langle r_0, \sigma\rangle \to \langle r_1, \sigma\rangle \to \langle r_2, \sigma\rangle \to \cdots$ where also $(\langle r_i, \sigma\rangle, \langle [\![\varsigma \mid q]\!], \sigma\rangle) \in R$ for all $i \geqslant 0$. According to the definition of R, there exist $p_0, p_1, p_2, \ldots \in P$ such that, for all $i \geqslant 0$, $r_i = [\![\varsigma \mid p_i]\!]$ and $(\langle p_i, \sigma \ll \varsigma\rangle, \langle q, \sigma \ll \varsigma\rangle) \in R_S$. Since Rule $\langle\mathsf{scp\text{-}tran}\rangle$ is the final rule in any derivation with $\langle r_i, \sigma\rangle \to \langle r_{i+1}, \sigma\rangle$ as conclusion, we have that $p_0 = p$ and $\langle p_0, \sigma \ll \varsigma\rangle \to \langle p_1, \sigma \ll \varsigma\rangle \to \langle p_2, \sigma \ll \varsigma\rangle \to \cdots$. Since R_S satisfies $\langle\mathsf{div}\rangle$, it follows that there exist $q' \in P$ and $k \geqslant 0$ such that $\langle q, \sigma \ll \varsigma\rangle \xrightarrow{+} \langle q', \sigma \ll \varsigma\rangle$ and $(\langle p_k, \sigma \ll \varsigma\rangle, \langle q', \sigma \ll \varsigma\rangle) \in R_S$. As before, by Rule $\langle\mathsf{scp\text{-}tran}\rangle$, $\langle [\![\varsigma \mid q]\!], \sigma\rangle \xrightarrow{+} \langle [\![\varsigma \mid q']\!], \sigma\rangle$. According to the definition of R, $(\langle [\![\varsigma \mid p_k]\!], \sigma\rangle, \langle [\![\varsigma \mid q']\!], \sigma\rangle) \in R$. ■

Lemma 4.1.4 If R_S is a silent bisimulation, then

$$R = \{(\langle \partial_\Xi(p), \sigma\rangle, \langle \partial_\Xi(q), \sigma\rangle) \mid (\langle p, \sigma\rangle, \langle q, \sigma\rangle) \in R_{pq}\},$$

with $\Xi \subseteq \mathsf{A} \setminus \{\tau\}$, is also a silent bisimulation. □

Proof We show that R satisfies the conditions of Definition 2.2.1. It is symmetric because R_S is symmetric. That it satisfies $\langle\mathsf{lab}\rangle$ follows directly from its definition. We show that it also satisfies $\langle\mathsf{term}\rangle$, $\langle\mathsf{tran}\rangle$ and $\langle\mathsf{div}\rangle$.

Cond. $\langle\mathsf{term}\rangle$**:** Suppose $\langle \partial_\Xi(p), \sigma\rangle\downarrow$. Rule $\langle\mathsf{enc\text{-}term}\rangle$ is the final rule with this as conclusion so we have $\langle p, \sigma\rangle\downarrow$. Since $(\langle p, \sigma\rangle, \langle q, \sigma\rangle) \in R_S$, there is a $q' \in P$ such that $\langle q,\sigma\rangle \twoheadrightarrow \langle q',\sigma\rangle$, $\langle q',\sigma\rangle\downarrow$ and $(\langle p,\sigma\rangle,\langle q',\sigma\rangle) \in R_S$. Now, by Rules $\langle\mathsf{enc\text{-}term}\rangle$ and $\langle\mathsf{enc\text{-}tran}\rangle$, $\langle\partial_\Xi(s),\sigma\rangle \twoheadrightarrow \langle\partial_\Xi(q'),\sigma\rangle$ and $\langle\partial_\Xi(q'),\sigma\rangle\downarrow$. Since $(\langle p,\sigma\rangle,\langle q',\sigma\rangle) \in R_S$, according to the definition of R, $(\langle\partial_\Xi(r),\sigma\rangle,\langle\partial_\Xi(q'),\sigma\rangle) \in R$.

Cond. $\langle\mathsf{tran}\rangle$**:** Suppose $\langle\partial_\Xi(p),\sigma\rangle \xrightarrow{a} \langle r,\sigma'\rangle$ for some $\sigma' \in \Sigma$ and $r \in P$. Since Rule $\langle\mathsf{enc\text{-}tran}\rangle$ is the final rule with this transition as conclusion, we have $\langle p,\sigma\rangle \xrightarrow{a} \langle p',\sigma'\rangle$, $r = \partial_\Xi(p')$ and $a \notin \Xi$. Since $(\langle p,\sigma\rangle,\langle q,\sigma\rangle) \in R_S$, there exist $q', q'' \in P$ such that $\langle q,\sigma\rangle \twoheadrightarrow \langle q'',\sigma\rangle \xrightarrow{(a)} \langle q',\sigma'\rangle$, $(\langle p,\sigma\rangle,\langle q'',\sigma\rangle) \in R_S$ and $(\langle p',\sigma'\rangle,\langle q',\sigma'\rangle) \in R_S$. Now, because $\tau \notin \Xi$, by Rule $\langle\mathsf{enc\text{-}tran}\rangle$, $\langle\partial_\Xi(s_0),\sigma\rangle \twoheadrightarrow \langle\partial_\Xi(q''),\sigma\rangle \xrightarrow{(a)} \langle\partial_\Xi(q'),\sigma'\rangle$. Since $(\langle p,\sigma\rangle,\langle q'',\sigma\rangle) \in R_S$ and $(\langle p',\sigma'\rangle,\langle q',\sigma'\rangle) \in R_S$, according to the definition of R, we have that $(\langle\partial_\Xi(p),\sigma\rangle,\langle\partial_\Xi(q''),\sigma\rangle) \in R$ and $(\langle\partial_\Xi(p'),\sigma'\rangle,\langle\partial_\Xi(q'),\sigma'\rangle) \in R$.

Cond. $\langle\mathsf{div}\rangle$**:** Suppose that there exist $r_0, r_1, r_2, \ldots \in P$, such that $r_0 = \partial_\Xi(p)$, $\langle r_0,\sigma\rangle \to \langle r_1,\sigma\rangle \to \langle r_2,\sigma\rangle \to \cdots$ and that $(\langle r_i,\sigma\rangle,\langle\partial_\Xi(q),\sigma\rangle) \in R$ for all $i \geqslant 0$. According to the definition of R, there exist $p_0, p_1, p_2, \ldots \in P$ such that $r_i = \partial_\Xi(p_i)$ for all $i \geqslant 0$. Since Rule $\langle\mathsf{enc\text{-}tran}\rangle$ is the only rule that has $\langle\partial_\Xi(p_i),\sigma\rangle \to \langle\partial_\Xi(p_{i+1}),\sigma\rangle$ as conclusion, we have $\langle p_0,\sigma\rangle \to \langle p_1,\sigma\rangle \to \langle p_2,\sigma\rangle \to \cdots$. By the definition of R, also $(\langle p_i,\sigma\rangle,\langle\partial_\Xi(q),\sigma\rangle) \in R_S$. Since R_S satisfies $\langle\mathsf{div}\rangle$, there exist $q' \in P$ and $k \geqslant 0$ such that $\langle q,\sigma\rangle \to^+ \langle q',\sigma\rangle$ and $(\langle p_k,\sigma\rangle,\langle q',\sigma\rangle) \in R_S$. By Rule $\langle\mathsf{enc\text{-}tran}\rangle$, $\langle\partial_\Xi(q),\sigma\rangle \to^+ \langle\partial_\Xi(q'),\sigma\rangle$. Since $(\langle p_k,\sigma\rangle,\langle q',\sigma\rangle) \in R_S$, according to the definition of R, $(\langle\partial_\Xi(p_k),\sigma\rangle,\langle\partial_\Xi(q'),\sigma\rangle) \in R$. ■

Lemma 4.1.5 If R_S is a silent bisimulation, then

$$R = \{(\langle\tau_I(p),\sigma\rangle,\langle\tau_I(q),\sigma\rangle) \mid (\langle p,\sigma\rangle,\langle q,\sigma\rangle) \in R_S\},$$

with $I \subseteq \mathsf{A}$, is also a silent bisimulation. □

Proof We show that R satisfies the conditions of Definition 2.2.1. It is symmetric because R_S is symmetric. That it satisfies $\langle\mathsf{lab}\rangle$ follows directly from its definition. We show that it also satisfies $\langle\mathsf{term}\rangle$, $\langle\mathsf{tran}\rangle$ and $\langle\mathsf{div}\rangle$.

Cond. $\langle\mathsf{term}\rangle$**:** Suppose $\langle\tau_I(p),\sigma\rangle\downarrow$. Rule $\langle\mathsf{hide\text{-}term}\rangle$ is the final rule with this as conclusion so we have $\langle p,\sigma\rangle\downarrow$. Since $(\langle p,\sigma\rangle,\langle q,\sigma\rangle) \in R_S$, there is a $q' \in P$ such that $\langle q,\sigma\rangle \twoheadrightarrow \langle q',\sigma\rangle$, $\langle q',\sigma\rangle\downarrow$ and $(\langle p,\sigma\rangle,\langle q',\sigma\rangle) \in R_S$.

By Rule $\langle\mathsf{hide\text{-}term}\rangle$, and by Rules $\langle\mathsf{hide\text{-}tran}_1\rangle$ and $\langle\mathsf{hide\text{-}tran}_2\rangle$, we obtain $\langle\tau_I(q),\sigma\rangle \twoheadrightarrow \langle\tau_I(q'),\sigma\rangle$ and $\langle\tau_I(q'),\sigma\rangle\downarrow$. Since $(\langle p,\sigma\rangle,\langle q',\sigma\rangle)\in R_S$, according to the definition of R, $(\langle\tau_I(p),\sigma\rangle,\langle\tau_I(q'),\sigma\rangle)\in R$.

Cond. $\langle\mathsf{tran}\rangle$**:** Suppose $\langle\tau_I(p),\sigma\rangle \xrightarrow{a} \langle r,\sigma'\rangle$ for some $\sigma'\in\Sigma$ and $r\in P$. Since Rules $\langle\mathsf{hide\text{-}tran}_1\rangle$ and $\langle\mathsf{hide\text{-}tran}_2\rangle$ are the final rules with this transition as conclusion, we have $\langle p,\sigma\rangle \xrightarrow{b} \langle p',\sigma'\rangle$ and $r=\tau_I(p')$, for some $b\in\mathsf{A}$ such that either $b\in I$ and $a=\tau$ or $b\notin I$ and $a=b$. Since $(\langle p,\sigma\rangle,\langle q,\sigma\rangle)\in R_S$ there exist $q',q''\in P$ such that $\langle q,\sigma\rangle \twoheadrightarrow \langle q'',\sigma\rangle \xrightarrow{(b)} \langle q',\sigma\rangle$, $(\langle p,\sigma\rangle,\langle q'',\sigma\rangle)\in R_S$, and $(\langle p',\sigma'\rangle,\langle q',\sigma'\rangle)\in R_S$. Note that, if $b=\tau$, then also $a=\tau$. Now, by Rules $\langle\mathsf{hide\text{-}tran}_1\rangle$ and $\langle\mathsf{hide\text{-}tran}_2\rangle$, $\langle\tau_I(q),\sigma\rangle \twoheadrightarrow \langle\tau_I(q''),\sigma\rangle \xrightarrow{(a)} \langle\tau_I(q'),\sigma'\rangle$. Since $(\langle p,\sigma\rangle,\langle q'',\sigma\rangle)\in R_S$, and $(\langle p',\sigma'\rangle,\langle q',\sigma'\rangle)\in R_S$, according to the definition of R, we have $(\langle\tau_I(p),\sigma\rangle,\langle\tau_I(q''),\sigma\rangle)\in R$ and $(\langle\tau_I(p'),\sigma'\rangle,\langle\tau_I(q'),\sigma'\rangle)\in R$.

Cond. $\langle\mathsf{div}\rangle$**:** Suppose that there exist $r_0,r_1,r_2,\ldots\in P$, such that $r_0=\tau_I(p)$, $\langle r_0,\sigma\rangle \to \langle r_1,\sigma\rangle \to \langle r_2,\sigma\rangle \to \cdots$ and $(\langle r_i,\sigma\rangle,\langle\tau_I(q),\sigma\rangle)\in R$ for all $i\geqslant 0$. By Rules $\langle\mathsf{hide\text{-}tran}_1\rangle$ and $\langle\mathsf{hide\text{-}tran}_2\rangle$, there exist $p_0,p_1,p_2,\ldots\in P$ and $a_0,a_1,a_2,\ldots\in\mathsf{A}$ such that $\langle p_0,\sigma\rangle \xrightarrow{a_0} \langle p_1,\sigma\rangle \xrightarrow{a_1} \langle p_2,\sigma\rangle \xrightarrow{a_2} \cdots$ and, for all $i\geqslant 0$, $r_i=\tau_I(p_i)$ and either $a_i=\tau$ or $a_i\in I$. According to the definition of R, $(\langle p_i,\sigma\rangle,\langle q,\sigma\rangle)\in R_S$ for all $i\geqslant 0$.

Suppose first that $a_i=\tau$ for all $i\geqslant 0$. Since R_S satisfies $\langle\mathsf{div}\rangle$, there exist an $q'\in P$ and $k\geqslant 0$ such that $\langle q,\sigma\rangle \to^{+} \langle q',\sigma\rangle$ and $(\langle p_k,\sigma\rangle,\langle q',\sigma\rangle)\in R_S$. By Rules $\langle\mathsf{hide\text{-}tran}_1\rangle$ and $\langle\mathsf{hide\text{-}tran}_2\rangle$, we obtain $\langle\tau_I(q),\sigma\rangle \to^{+} \langle\tau_I(q'),\sigma\rangle$. Since $(\langle p_k,\sigma\rangle,\langle q',\sigma\rangle)\in R_S$, according to the definition of R, $(\langle\tau_I(p),\sigma\rangle,\langle\tau_I(q''),\sigma\rangle)\in R$.

Let now $n\geqslant 0$ be the smallest index such that $a_n\neq\tau$ and $a_n\in I$. Since $(\langle p_n,\sigma\rangle,\langle q,\sigma\rangle)\in R_S$, there exist $q',q''\in P$ such that $\langle q,\sigma\rangle \twoheadrightarrow \langle q',\sigma\rangle \xrightarrow{a_n} \langle q',\sigma\rangle$, $(\langle p_n,\sigma\rangle,\langle q'',\sigma\rangle)\in R_S$ and $(\langle p_{n+1},\sigma\rangle,\langle q'',\sigma\rangle)\in R_S$. By Rules $\langle\mathsf{hide\text{-}tran}_1\rangle$ and $\langle\mathsf{hide\text{-}tran}_2\rangle$, $\langle\tau_I(q),\sigma\rangle \twoheadrightarrow \langle\tau_I(q''),\sigma\rangle \xrightarrow{\tau} \langle\tau_I(q'),\sigma\rangle$. From this it clearly follows that $\langle\tau_I(q),\sigma\rangle \to^{+} \langle\tau_I(q'),\sigma\rangle$. Since $(\langle p_{n+1},\sigma\rangle,\langle q'',\sigma\rangle)\in R_S$, according to the definition of R, we have $(\langle\tau_I(p_{n+1}),\sigma\rangle,\langle\tau_I(q''),\sigma\rangle)\in R$. ■

Now we can easily prove the following.

Theorem 4.1.6 For all $p,q\in P$, if $p\sim_s q$, then

1. $[\![\varsigma \mid p]\!] \sim_s [\![\varsigma \mid q]\!]$, for all $\varsigma\in\Sigma$;

2. $\tau_I(p) \sim_s \tau_I(q)$, for all $I \subseteq \mathsf{A}$; and

3. $\partial_\Xi(p) \sim_s \partial_\Xi(q)$, for all $\Xi \subseteq \mathsf{A} \setminus \{\tau\}$. □

In the next section we deal with parallel composition.

4.2 Stateless silent bisimulation

Silent bisimulation is not a congruence for parallel composition. To show this consider the following example.

Example 4.2.1 a. Let $a \in \mathsf{A}$, let $\mathsf{b} \in \mathsf{B}$ and let $p \in P$. Suppose that, for all $\sigma \in \Sigma$ and all $\sigma' \in \mathsf{eff}(a, \sigma) = \{\sigma'\}$, $\mathsf{check}(\mathsf{b}, \sigma') = \mathit{true}$. Then, the processes $\boldsymbol{a} \cdot \mathsf{b} :\rightarrow p$ and $\boldsymbol{a} \cdot p$ are silently bisimilar. They both do the action a and proceed as the process p. Let now $b \in \mathsf{A}$ be such that, for all $\sigma \in \Sigma$ and all $\sigma' \in \mathsf{eff}(b, \sigma)$, $\mathsf{check}(\mathsf{b}, \sigma') = \mathit{false}$. When put in parallel with the process $\boldsymbol{b}$ the two processes from above behave differently in any valuation. The process $(\boldsymbol{a} \cdot \mathsf{b} :\rightarrow p) \parallel \boldsymbol{b}$ can execute a, then b, and then deadlock. The process $(\boldsymbol{a} \cdot p) \parallel \boldsymbol{b}$ cannot deadlock (assuming that p does not deadlock).

b. It is easily shown that $\boldsymbol{a} \cdot \tau_{\{a\}}(\boldsymbol{a}) \sim_s \boldsymbol{a}$ if, for example, for all $\sigma \in \Sigma$ and all $\sigma' \in \mathsf{eff}(a, \sigma)$, we have $\mathsf{eff}(a, \sigma') = \{\sigma'\}$. However, in general, $\boldsymbol{a} \cdot \tau_{\{a\}}(\boldsymbol{a}) \parallel \boldsymbol{b} \not\sim_s \boldsymbol{a} \parallel \boldsymbol{b}$. To show this, let $\sigma \in \Sigma$ be some valuation and suppose $\mathsf{eff}(a, \sigma) = \{\sigma'\}$, $\mathsf{eff}(b, \sigma') = \{\sigma''\}$ and $\mathsf{eff}(a, \sigma'') = \{\sigma'''\}$. Now, the left-hand side process can change the valuation from σ to σ', then to σ'' and then, finally, to σ'''. This behavior cannot be simulated by the right-hand side process in case $\sigma''' \neq \sigma''$. □

The reason why silent bisimilarity fails to be a congruence for parallel composition is because it is blind to a change in the intermediate data state caused by a parallel component. The same problem also occurs when strong bisimulation is lifted to a relation on process terms and the solution is known [57, 20, 80]. The idea is to require that after performing a step two bisimilar processes are again bisimilar, but in *every* valuation and not only in the resulting one. A condition that ensures this requirement is given in the following definition.

Definition 4.2.2 A binary relation R on $S = P \times \Sigma$ is called *stateless* iff,

$$(\langle p, \sigma\rangle, \langle q, \sigma\rangle) \in R \text{ implies that for all } \sigma' \in \Sigma, (\langle p, \sigma'\rangle, \langle q, \sigma'\rangle) \in R. \quad \square$$

Before we use this definition to define a relation on processes we establish some properties of stateless relations. The following two lemmas show that the union and composition of two stateless relations is again stateless.

Lemma 4.2.3 Let R_i for $i \in I$ be some stateless binary relations on $S = P \times \Sigma$. Then their union $\bigcup_{i \in I} R_i$ is also stateless □

Proof Let $R = \bigcup_{i \in I} R_i$. Suppose $(\langle p, \sigma \rangle, \langle q, \sigma \rangle) \in R$ for some $p, q \in P$ and $\sigma \in \Sigma$. Then there exists an $i \geqslant 0$ such that $(\langle p, \sigma \rangle, \langle q, \sigma \rangle) \in R_i$. Let $\sigma' \in \Sigma$. Then, because R_i is stateless, also $(\langle p, \sigma' \rangle, \langle q, \sigma' \rangle) \in R_i$. From this $(\langle p, \sigma' \rangle, \langle q, \sigma' \rangle) \in R$. ■

Lemma 4.2.4 Let R_1 and R_2 be stateless binary relations on $S = P \times \Sigma$. Then their composition $R_1 \circ R_2$ is also stateless. □

Proof Let now $R = R_1 \circ R_2$ and suppose $(\langle p, \sigma \rangle, \langle r, \sigma \rangle) \in R$ for some $p, r \in P$ and $\sigma \in \Sigma$. Then there exists a $q \in P$ such that $(\langle p, \sigma \rangle, \langle q, \sigma \rangle) \in R_1$ and $(\langle q, \sigma \rangle, \langle r, \sigma \rangle) \in R_2$. Let $\sigma' \in \Sigma$. Then, because R_1 and R_2 are stateless, $(\langle p, \sigma' \rangle, \langle q, \sigma' \rangle) \in R_1$ and $(\langle q, \sigma' \rangle, \langle r, \sigma' \rangle) \in R_2$. From this $(\langle p, \sigma' \rangle, \langle r, \sigma' \rangle) \in R$. ■

Now we could lift the definition of $\sim_s$ to the level of κ processes by saying that two processes, p and q, are silently bisimilar iff there is a stateless silent bisimulation relation R such that $(\langle p, \sigma \rangle, \langle q, \sigma \rangle) \in R$ for some, and thus for all, $\sigma \in \Sigma$. Having proved Lemmas 4.2.3 and 4.2.4 we can straightforwardly prove that this relation is an equivalence relation. However, since the definition gives an extra proof obligation in application (the stateless property), it is more usual to define the equivalence as follows.

Definition 4.2.5 (Stateless Silent Bisimulation on Processes) A symmetric relation $R \subseteq P \times P$ is a *stateless silent bisimulation* (on processes) iff, for all $(p, q) \in R$ and for all $\sigma \in \Sigma$,

⟨sl-term⟩ if $\langle p, \sigma \rangle \downarrow$, then there exist $q' \in P$ such that

$$\langle q, \sigma \rangle \twoheadrightarrow \langle q', \sigma \rangle,\ \langle q', \sigma \rangle \downarrow \text{ and } (p, q') \in R,$$

⟨sl-tran⟩ if $\langle p, \sigma \rangle \xrightarrow{a} \langle p', \sigma' \rangle$, then there exist $q', q'' \in P$ such that

$$\langle q, \sigma \rangle \twoheadrightarrow \langle q'', \sigma \rangle \xrightarrow{(a)} \langle q', \sigma' \rangle,\ (p, q'') \in R \text{ and } (p', q') \in R,$$

⟨sl-div⟩ if there exists an infinite sequence $p_0, p_1, p_2, \ldots \in P$ such that $p_0 = p$,

$$\langle p_0, \sigma\rangle \rightarrow \langle p_1, \sigma\rangle \rightarrow \langle p_2, \sigma\rangle \rightarrow \cdots \quad \text{and} \quad (p_i, q) \in R$$

for all $i \geqslant 0$, then there exist $q' \in P$ and $k \geqslant 0$ such that

$$\langle q, \sigma\rangle \rightarrow^{+} \langle q', \sigma\rangle \quad \text{and} \quad (p_k, q') \in R.$$

Two processes p and q are *stateless silent bisimilar*, denoted $p \sim_s^{sl} q$, if there exists a stateless silent bisimulation $R \subseteq P \times P$ such that $(p, q) \in R$. □

We now show that the above definition indeed leads to the desired notion of silent bisimilarity. The core of the proof is divided into two lemmas.

Lemma 4.2.6 Let R_P be a binary relation on processes and let

$$R = \{(\langle p, \sigma\rangle, \langle q, \sigma\rangle) \mid (p, q) \in R_P, \sigma \in \Sigma\}.$$

If R_P satisfies ⟨sl-con⟩, for $\mathsf{con} \in \{\mathsf{term}, \mathsf{tran}, \mathsf{div}\}$, then R satisfies ⟨con⟩. □

Proof Suppose that R_P satisfies ⟨sl-term⟩ and that $\langle p, \sigma\rangle\downarrow$. From $(p, q) \in R_P$ it follows that there exists an $s' \in P$ such that $\langle q, \sigma\rangle \twoheadrightarrow \langle q', \sigma\rangle$, $\langle q', \sigma\rangle\downarrow$ and $(p, q') \in R_P$. According to the definition of R, $(\langle p, \sigma\rangle, \langle q', \sigma\rangle) \in R$.

Suppose that R_P satisfies ⟨sl-tran⟩ and suppose $\langle p, \sigma\rangle \xrightarrow{a} \langle p', \sigma'\rangle$ for some $p' \in P, \sigma' \in \Sigma$. From $(p, q) \in R_P$ it follows that there exist $q', q'' \in P$ such that $\langle q, \sigma\rangle \twoheadrightarrow \langle q'', \sigma\rangle \xrightarrow{(a)} \langle q', \sigma'\rangle$, $(p, q'') \in R_P$, and $(p', q') \in R_P$. According to the definition of R, $(\langle p, \sigma\rangle, \langle q'', \sigma\rangle) \in R$ and $(\langle p', \sigma'\rangle, \langle q', \sigma'\rangle) \in R$.

Suppose that R_P satisfies ⟨sl-div⟩ and suppose $\langle p_0, \sigma\rangle \rightarrow \langle p_1, \sigma\rangle \rightarrow \langle p_2, \sigma\rangle \rightarrow \cdots$ for some $p_0 (= p), p_1, p_2 \ldots \in P$ such that $(\langle p_i, \sigma\rangle, \langle q, \sigma\rangle) \in R$ for all $i \geqslant 0$. This implies that $(p, q_i) \in R_P$ for all $i \leqslant n$. From that there exist $q' \in P$ and $k \geqslant 0$ such that $\langle q, \sigma\rangle \rightarrow^{+} \langle q', \sigma\rangle$ and $(p_k, q') \in R_P$. According to the definition of R, $(\langle p_k, \sigma\rangle, \langle q, \sigma\rangle) \in R$.

Lemma 4.2.7 Let R_S be a stateless binary relation on $S = P \times \Sigma$ and let

$$R = \{(p, q) \mid (\langle p, \sigma\rangle, \langle q, \sigma\rangle) \in R_S \text{ for some/all } \sigma \in \Sigma\}.$$

If R_S satisfies ⟨con⟩, for $\mathsf{con} \in \{\mathsf{term}, \mathsf{tran}, \mathsf{div}\}$, then R satisfies ⟨sl-con⟩. □

Proof Suppose that R_S satisfies $\langle\mathsf{term}\rangle$ and suppose $\langle p,\sigma\rangle\downarrow$ for some $\sigma \in \Sigma$. It follows that $(\langle p,\sigma\rangle,\langle q,\sigma\rangle) \in R_S$, and so there exists a $q' \in P$ such that $\langle q,\sigma\rangle \twoheadrightarrow \langle q',\sigma\rangle$, $\langle q',\sigma\rangle\downarrow$ and $(\langle p,\sigma\rangle,\langle q',\sigma\rangle) \in R_S$. According to the definition of R, $(p,q') \in R$.

Suppose that R_S satisfies $\langle\mathsf{tran}\rangle$ and suppose $\langle p,\sigma\rangle \xrightarrow{a} \langle p',\sigma'\rangle$ for some $r' \in P, \sigma, \sigma' \in \Sigma$. It follows that there exist $s', s'' \in P$ such that $\langle q,\sigma\rangle \twoheadrightarrow \langle q'',\sigma\rangle \xrightarrow{(a)} \langle q',\sigma'\rangle$, $(\langle p,\sigma\rangle,\langle q',\sigma\rangle) \in R_S$, and $(\langle p',\sigma'\rangle,\langle q',\sigma'\rangle) \in R_S$. According the definition of R, $(p,q'') \in R$ and $(p',q') \in R$.

Suppose that R_S satisfies $\langle\mathsf{div}\rangle$ and suppose $\langle p_0,\sigma\rangle \to \langle p_1,\sigma\rangle \to \langle p_2,\sigma\rangle \to \cdots$ for some $p_0(=r), p_1, p_2 \ldots \in P$ and $\sigma \in \Sigma$ such that $(p_i,q) \in R$ for all $i \geqslant 0$. This implies that $(\langle p,\sigma\rangle,\langle q_i,\sigma\rangle) \in R_S$ for all $i \geqslant 0$. From that there exist q' and $j \geqslant 0$ such that $\langle q,\sigma\rangle \to^+ \langle q',\sigma\rangle$ and $(\langle p_j,\sigma\rangle,\langle q',\sigma\rangle) \in R_S$. From the stateless property of R_S, and the definition of R, $(p_j,q') \in R$. ■

The following now easily follows.

Theorem 4.2.8 For all $p,q \in P$, $p \sim_s^{sl} q$ iff there is a stateless silent bisimulation relation $R \in S \times S$ such that $(\langle p,\sigma\rangle,\langle q,\sigma\rangle) \in R$ for some, and thus for all, $\sigma \in \Sigma$. □

Proof Suppose first that $p \sim_s^{sl} q$. Then there exists a binary relation R_{pq} that satisfies the conditions of Definition 4.2.5 and such that $(p,q) \in R_{pq}$. Let $R = \{(\langle r,\sigma\rangle,\langle s,\sigma\rangle) \mid (r,s) \in R_{pq}, \sigma \in \Sigma\}$. The relation R is by definition stateless. It also satisfies $\langle\mathsf{lab}\rangle$. From this and Lemma 4.2.6 we conclude that R is an silent bisimulation. Clearly, $(\langle p,\sigma\rangle,\langle q,\sigma\rangle) \in R$ for some, and thus for all, $\sigma \in \Sigma$.

Suppose now that there is a stateless silent bisimulation R_{pq} such that $(\langle p,\sigma\rangle,\langle q,\sigma\rangle) \in R_{pq}$ for all $\sigma \in \Sigma$. Let $R = \{(r,s) \mid (\langle r,\sigma\rangle,\langle s,\sigma\rangle) \in R_{pq}\}$. Note that $(p,q) \in R$. From Lemma 4.2.7 it follows that R satisfies $\langle\mathsf{sl\text{-}term}\rangle$, $\langle\mathsf{sl\text{-}tran}\rangle$ and $\langle\mathsf{sl\text{-}div}\rangle$, and thus it is a stateless silent bisimulation on processes. ■

Theorem 4.2.8 establishes a direct link between the stateless silent bisimulation and the standard silent bisimulation. This allows allows us to use results from before, most importantly Lemmas 4.2.3 and 4.2.4, and to easily prove the following.

Corollary 4.2.9 The relation $\sim_s^{sl}$ is an equivalence relation. □

Corollary 4.2.10 The relation $\sim_s^{sl}$ is a stateless silent bisimulation on processes. □

We now show that stateless silent bisimulation on processes is a compatible with parallel composition. The following lemma is the core of the proof.

Lemma 4.2.11 Let $R_P \subseteq P \times P$ and $\bar{R}_P \subseteq P \times P$ be stateless silent bisimulations. Then

$$R = \{(p \parallel \bar{p}, q \parallel \bar{q}) \mid (p, q) \in R_P, (\bar{p}, \bar{q}) \in \bar{R}_P\}$$

is also a stateless silent bisimulation. □

Proof Note that R is symmetric because R_P and $\bar{R}_P$ are. We show that it satisfies $\langle\mathsf{sl\text{-}term}\rangle$, $\langle\mathsf{sl\text{-}tran}\rangle$ and $\langle\mathsf{sl\text{-}div}\rangle$.

Cond. $\langle\mathsf{sl\text{-}term}\rangle$**:** Suppose $\langle p \parallel \bar{p}, \sigma\rangle\downarrow$ for some $\sigma \in \Sigma$. Since Rule $\langle\mathsf{par\text{-}term}\rangle$ is the only rule with $\langle p \parallel \bar{p}, \sigma\rangle\downarrow$ as conclusion, we have $\langle p, \sigma\rangle\downarrow$ and $\langle \bar{p}, \sigma\rangle\downarrow$. Since $(p, q) \in R_P$ there exists an $q' \in P$ such that $\langle q, \sigma\rangle \twoheadrightarrow \langle q', \sigma\rangle$, $\langle q', \sigma\rangle\downarrow$, and $(p, q') \in R_P$. Since $(\bar{p}, \bar{q}) \in \bar{R}_P$, there exists a $\bar{q}' \in P$ $\langle \bar{q}, \sigma\rangle \twoheadrightarrow \langle \bar{q}', \sigma\rangle$, $\langle \bar{q}', \sigma\rangle\downarrow$, and $(\bar{p}, \bar{q}') \in \bar{R}_P$. By Rule $\langle\mathsf{par\text{-}tran_1}\rangle$, $\langle q \parallel \bar{q}, \sigma\rangle \twoheadrightarrow \langle q' \parallel \bar{q}, \sigma\rangle \twoheadrightarrow \langle q' \parallel \bar{q}', \sigma\rangle$, and by Rule $\langle\mathsf{par\text{-}term}\rangle$ $\langle q' \parallel \bar{q}', \sigma\rangle\downarrow$. Since $(p, q') \in R_P$ and $(\bar{p}, \bar{q}') \in \bar{R}_P$, according to the definition of R it holds that $(p \parallel \bar{p}, q' \parallel \bar{q}') \in R$.

Cond. $\langle\mathsf{sl\text{-}tran}\rangle$**:** Suppose $\langle p \parallel \bar{p}, \sigma\rangle \xrightarrow{a} \langle r, \sigma'\rangle$ for some $\sigma, \sigma' \in \Sigma$ and $r \in P$. The final rule of a derivation with this transition as conclusion is either Rule $\langle\mathsf{par\text{-}tran_1}\rangle$ or Rule $\langle\mathsf{par\text{-}tran_2}\rangle$; we treat these cases separately.

If the final rule applied is Rule $\langle\mathsf{par\text{-}tran_1}\rangle$, then $\langle p, \sigma\rangle \xrightarrow{a} \langle p', \sigma\rangle$ and $r = p' \parallel \bar{p}$ (or, symmetrically $\langle \bar{p}, \sigma\rangle \xrightarrow{a} \langle \bar{p}', \sigma\rangle$ and $r = p \parallel \bar{p}'$). Since $(p, q) \in R_P$, there exist $q', q'' \in P$ such that $\langle q, \sigma\rangle \twoheadrightarrow \langle q'', \sigma\rangle \xrightarrow{(a)} \langle q', \sigma'\rangle$, $(p, q'') \in R_P$ and $(p', q') \in R_P$. By Rule $\langle\mathsf{par\text{-}tran_1}\rangle$, $\langle q \parallel \bar{q}, \sigma\rangle \twoheadrightarrow \langle q'' \parallel \bar{q}, \sigma\rangle \xrightarrow{(a)} \langle q' \parallel \bar{q}, \sigma'\rangle$. Since $(p, q'') \in R_P$ and $(p', q') \in R_P$, according to the definition of R, $(p \parallel \bar{p}, q'' \parallel \bar{q}) \in R$ and $(p' \parallel \bar{p}, q' \parallel \bar{q}) \in R$. Note that the last step is where the stateless property is crucial. We used the fact that $\langle p', \sigma\rangle$ and $\langle q', \sigma\rangle$ are silently bisimilar. This might not be true in general if R_P were not stateless.

If the final rule applied is Rule $\langle\mathsf{par\text{-}tran_2}\rangle$, then there exist $b, c \in \mathsf{A}$ such that $\langle p, \sigma\rangle \xrightarrow{b} \langle p', \sigma''\rangle$, $\langle \bar{p}, \sigma\rangle \xrightarrow{c} \langle \bar{p}', \sigma'''\rangle$, $r = p' \parallel \bar{p}'$, and $a = \mathsf{act}(\mathsf{comm}(b, c), \sigma)$. In addition, we have $\sigma' = \mathsf{eff}(\mathsf{comm}(b, c), \sigma) = \sigma'' \ll \sigma'''_{/\mathrm{dom}(\sigma''')\setminus\mathrm{dom}(\sigma'')} = \sigma''' \ll \sigma''_{/\mathrm{dom}(\sigma'')\setminus\mathrm{dom}(\sigma''')}$. Since $(p, q) \in R_P$, there exist q', q'' such that $\langle q, \sigma\rangle \twoheadrightarrow \langle q'', \sigma\rangle \xrightarrow{b} \langle q', \sigma''\rangle$, $(p, q'') \in R_P$, and $(p', q') \in R_P$. Since $(\bar{p}, \bar{q}) \in \bar{R}_P$, there exist $\bar{q}', \bar{q}'' \in P$ such that $\langle \bar{q}, \sigma \rangle \twoheadrightarrow \langle \bar{q}'', \sigma \rangle \xrightarrow{c} \langle \bar{q}', \sigma''' \rangle$, $(\bar{p}, \bar{q}'') \in \bar{R}_P$ and $(\bar{p}', \bar{q}') \in \bar{R}_P$. By Rule $\langle \mathsf{par\text{-}tran}_1 \rangle$, $\langle q \parallel \bar{q}, \sigma \rangle \twoheadrightarrow \langle q'' \parallel \bar{q}, \sigma \rangle \twoheadrightarrow \langle q'' \parallel \bar{q}'', \sigma \rangle$. By Rule $\langle \mathsf{par\text{-}tran}_2 \rangle$, $\langle q'' \parallel \bar{q}'', \sigma \rangle \xrightarrow{a} \langle q' \parallel \bar{q}', \sigma' \rangle$. Since $(p, q'') \in R_P$, $(p', q') \in R_P$, $(\bar{p}, \bar{q}'') \in \bar{R}_P$ and $(\bar{p}', \bar{q}') \in \bar{R}_P$, according to the definition of R, $(p \parallel \bar{p}, q'' \parallel \bar{q}'') \in R$ and $(p' \parallel \bar{p}', q' \parallel \bar{q}') \in R$.

Cond. $\langle \mathsf{sl\text{-}div} \rangle$**:** Suppose that there exist $\sigma \in \Sigma$ and $r_0, r_1, r_2, \ldots \in P$, such that $r_0 = p \parallel \bar{p}$, $\langle r_0, \sigma \rangle \to \langle r_1, \sigma \rangle \to \langle r_2, \sigma \rangle \to \cdots$ and $(r_i, q \parallel \bar{q}) \in R$ for all $i \geqslant 0$. By Rules $\langle \mathsf{par\text{-}tran}_1 \rangle$ and $\langle \mathsf{par\text{-}tran}_2 \rangle$ it easily follows that there exist $p_0, p_1, p_2, \ldots \in P$ and $\bar{p}_0, \bar{p}_1, \bar{p}_2, \ldots \in P$ such that $p_0 = p$, $\bar{p}_0 = \bar{p}$ and, for all $i \geqslant 0$, $r_i = p_i \parallel \bar{p}_i$ and either $\langle p_i, \sigma \rangle \to \langle p_{i+1}, \sigma \rangle$ and $\bar{p}_{i+1} = \bar{p}_i$, or $\langle \bar{p}_i, \sigma \rangle \to \langle \bar{p}_{i+1}, \sigma \rangle$ and $p_{i+1} = p_i$, or $\langle p_i, \sigma \rangle \xrightarrow{a} \langle p_{i+1}, \sigma \rangle$ and $\langle \bar{p}_i, \sigma \rangle \xrightarrow{b} \langle \bar{p}_{i+1}, \sigma \rangle$ for some $a, b \in \mathsf{A}$ such that $\mathsf{comm}(a, b) = \tau$. From $r_i = p_i \parallel \bar{p}_i$ and $(r_i, q \parallel \bar{q}) \in R$, according to the definition of R we have $(p_i, q) \in R_P$ and $(\bar{p}_i, \bar{q}) \in \bar{R}_P$ for all $i \geqslant 0$.

Suppose that there exists an $n \geqslant 0$ such that $\langle p_n, \sigma \rangle \xrightarrow{a} \langle p_{n+1}, \sigma \rangle$ and $\langle \bar{p}_n, \sigma \rangle \xrightarrow{b} \langle \bar{p}_{n+1}, \sigma \rangle$ and suppose that this n is the smallest such index. Since $(p_n, q) \in R_P$, there exist $q', q'' \in P$ such that $\langle q, \sigma \rangle \twoheadrightarrow \langle q', \sigma \rangle \xrightarrow{a} \langle q', \sigma \rangle$, $(p_n, q'') \in R_P$ and $(p_{n+1}, q') \in R_P$. Since $(\bar{p}_i, \bar{q}) \in \bar{R}_P$, there exist $\bar{q}', \bar{q}'' \in P$ such that $\langle \bar{q}, \sigma \rangle \twoheadrightarrow \langle \bar{q}', \sigma \rangle \xrightarrow{b} \langle \bar{q}', \sigma \rangle$, $(\bar{p}_n, \bar{q}'') \in \bar{R}_P$ and $(\bar{p}_{n+1}, \bar{q}') \in \bar{R}_P$. Now, by Rules $\langle \mathsf{par\text{-}tran}_1 \rangle$ and $\langle \mathsf{par\text{-}tran}_2 \rangle$, $\langle q \parallel \bar{q}, \sigma \rangle \twoheadrightarrow \langle q'' \parallel \bar{q}, \sigma \rangle \twoheadrightarrow \langle q'' \parallel \bar{q}'', \sigma \rangle \xrightarrow{\tau} \langle q' \parallel \bar{q}', \sigma \rangle$. Clearly, $\langle q \parallel \bar{q}, \sigma \rangle \xrightarrow{}^{+} \langle q' \parallel \bar{q}', \sigma \rangle$. Since $(p_{n+1}, q') \in R_P$ and $(\bar{p}_{n+1}, \bar{q}') \in \bar{R}_P$, according to the definition of R $(p_{n+1} \parallel \bar{p}_{n+1}, q' \parallel \bar{q}') \in R$.

Suppose there is no such n. Then there exists an infinite sequence $i_0, i_1, i_2, \ldots$ such that $i_0 = 0$, $\langle p_{i_0}, \sigma \rangle \to \langle p_{i_1}, \sigma \rangle \to \langle p_{i_2}, \sigma \rangle \to \cdots$ and $r_{i_k} = p_{i_k} \parallel \bar{p}_{i_k}$ for all $k \geqslant 0$ (or the symmetric case when there is a similar sequence from $\langle \bar{p}, \sigma \rangle$). Since $(p_{i_k}, q) \in R_P$ for all $k \geqslant 0$, we have that there exists $q' \in P$ and $l \geqslant 0$ such that $\langle q, \sigma \rangle \xrightarrow{}^{+} \langle q', \sigma \rangle$ and $(p_{i_l}, q) \in R_P$. By Rule $\langle \mathsf{par\text{-}tran}_1 \rangle$, $\langle q \parallel \bar{q}, \sigma \rangle \xrightarrow{}^{+} \langle q' \parallel \bar{q}, \sigma \rangle$. Since $(p_{i_l}, q) \in R_P$ and $(\bar{p}_{i_l}, \bar{q}) \in \bar{R}_P$, according to the definition of R $(p_{i_l} \parallel \bar{p}_{i_l}, q' \parallel \bar{q}) \in R$. ■

The following now easily follows from Lemma 4.2.11.

Theorem 4.2.12 For all $p, q, \bar{p}, \bar{q} \in P$, if $p \sim_s^{sl} q$ and $\bar{p} \sim_s^{sl} \bar{q}$, then $p \parallel \bar{p} \sim_s^{sl} q \parallel \bar{q}$. □

We have proved that stateless silent bisimilarity is a congruence for parallel composition. That it is also a congruence for the encapsulation, scope and the hiding operator follows directly from Lemmas 4.1.3, 4.1.4, 4.1.5 where the defined relations R are clearly stateless. We show in the next section that one more thing needs to be done before we obtain a congruence for all operators in κ.

4.3 Root condition and congruence proof

Stateless silent bisimilarity is not a congruence for alternative composition, sequential composition and repetition. In fact, by requiring the bisimulation relation to be stateless, the congruence property for guards is lost as well. Consider the following example.

Example 4.3.1 Note that $\delta \sim_s^{sl} \boldsymbol{\tau} \cdot \delta$ if $\mathsf{eff}(\boldsymbol{\tau}, \sigma) = \{\sigma\}$ for all $\sigma \in \Sigma$. However,

a. $\boldsymbol{\tau} + \delta \not\sim_s^{sl} \boldsymbol{\tau} + \boldsymbol{\tau} \cdot \delta$, for the right-hand side process can perform the τ and then deadlock, while the left-hand side process never deadlocks;

b. $(\boldsymbol{a} + \varepsilon) \cdot \delta \not\sim_s^{sl} (\boldsymbol{a} + \varepsilon) \cdot \boldsymbol{\tau} \cdot \delta$, for the right-hand side process can perform the τ action and deadlock, avoiding to do the action a, while the left-hand side must always do the action a;

   and

c. $\boldsymbol{a}^* \cdot \delta \not\sim_s^{sl} \boldsymbol{a}^* \cdot \boldsymbol{\tau} \cdot \delta$, for the right-hand side process can perform the τ action and deadlock, while the left-hand can only execute the action a indefinitely. □

Note that the problem in all the three cases from above appears because they all involve some kind of non-deterministic choice. In the first case, the addition of the $\boldsymbol{\tau}$ process in front of δ masks the deadlock, giving the right hand side process the possibility to choose the "wrong" path. In the other two cases, the problem is similar, only that the non-determinism is not explicit, but implicitly hidden in the sequential composition, resp. the repetition.

The next example shows that $\sim_s^{sl}$ is also not a congruence for guards.

Example 4.3.2 Note that $\boldsymbol{a} \sim_s^{sl} \boldsymbol{\tau} \cdot \boldsymbol{a}$ when $\mathsf{eff}(\boldsymbol{\tau}, \sigma) = \{\sigma\}$ for all $\sigma \in \Sigma$. However, for some $\mathsf{b} \in \mathsf{B}$ that is not *true* in all valuations, $\mathsf{b} :\rightarrow \boldsymbol{a} \not\sim_s^{sl} \mathsf{b} :\rightarrow (\boldsymbol{\tau} \cdot \boldsymbol{a})$. This is because, for any valuation for which b is *true*, the right-hand side process performs τ, passes the guard and then behaves as

$\boldsymbol{a}$. The process $\mathsf{b} :\rightarrow \boldsymbol{a}$ can simulate this τ action only by doing nothing. However, the stateless property requires then that $\mathsf{b} :\rightarrow \boldsymbol{a}$ and $\boldsymbol{a}$ behave the same way in any valuation which is impossible. □

The problem illustrated in the first example and its solution are well known; we need to add a root condition [15]. This condition requires that related processes must simulate each other's initial steps in the strong sense. Adding the root condition to the relation also solves the problem for guards.

Definition 4.3.3 (Root condition) A pair $(p, q) \in P \times P$ satisfies the *root condition* in $R \subseteq P \times P$ if, for all $\sigma \in \Sigma$,

$\langle$root-term$\rangle$ $\langle p, \sigma\rangle\downarrow$ iff $\langle q, \sigma\rangle\downarrow$,

$\langle$root-tran$_1\rangle$ if $\langle p, \sigma\rangle \xrightarrow{a} \langle p', \sigma'\rangle$ for some $\sigma' \in \Sigma$, then there exists $q' \in P$ such that $\langle q, \sigma\rangle \xrightarrow{a} \langle q', \sigma'\rangle$ and $(p', q') \in R$, and

$\langle$root-tran$_2\rangle$ if $\langle q, \sigma\rangle \xrightarrow{a} \langle q', \sigma'\rangle$ for some $\sigma' \in \Sigma$, then there exists $p' \in P$ such that $\langle p, \sigma\rangle \xrightarrow{a} \langle p', \sigma'\rangle$ and $(p', q') \in R$. □

We now define a notion of silent congruence.

Definition 4.3.4 (Silent congruence) Two processes p and q are *silently congruent*, denoted $p \approx_s q$, iff there is a stateless silent bisimulation relation on processes R such that $(p, q) \in R$ and (p, q) satisfies that root condition in R. □

Clearly, $p \approx_s q$ implies $p \sim_s q$. The root condition is compositional so it is straightforward to prove that $\approx_s$ is an equivalence relation. We show that $\approx_s$ is a congruence. For that we need to prove some lemmas first.

The first lemma is the stateless analogue of Lemma 2.3.1; the second shows that we can define stateless silent bisimulation using the stateless variant of $\langle$div"$\rangle$. When proving the relation $\approx_s$ compositional it is sometimes more convenient to work with the condition $\langle$div"$\rangle$ than with the condition $\langle$div$\rangle$.

Lemma 4.3.5 If $R \subseteq P \times P$ satisfies $\langle$sl-tran$\rangle$ and if $(p, q) \in R$ and $\langle p, \sigma\rangle \twoheadrightarrow \langle p', \sigma\rangle$ for some $\sigma \in \Sigma$, then there is a $q' \in P$ such that $\langle q, \sigma\rangle \twoheadrightarrow \langle q', \sigma\rangle$ and $(p', q') \in R$. □

Proof By Lemma 4.2.6 it follows that the relation $R_S = \{(\langle p, \sigma\rangle, \langle q, \sigma\rangle) \mid (p, q) \in R,\ \sigma \in \Sigma\}$ satisfies $\langle$tran$\rangle$. It is clear from the definition that it also satisfies $\langle$lab$\rangle$. By Lemma 2.3.1, there is exists a $q' \in P$ such that $\langle q, \sigma\rangle \twoheadrightarrow \langle q', \sigma\rangle$ and $(\langle p', \sigma\rangle, \langle q', \sigma\rangle) \in R_S$. From $(\langle p', \sigma\rangle, \langle q', \sigma\rangle) \in R_S$, it follows that $(p', q') \in R$. ■

Lemma 4.3.6 If $R \subseteq P \times P$ satisfies $\langle \textsf{sl-tran} \rangle$ and $\langle \textsf{sl-div} \rangle$, then it also satisfies

$\langle \textsf{sl-div''} \rangle$ if there is an infinite sequence $p_0, p_1, p_2, \ldots \in P$ such that $p_0 = p$ and $\langle p_0, \sigma \rangle \to \langle p_1, \sigma \rangle \to \langle p_2, \sigma \rangle \to \cdots$ for some $\sigma \in \Sigma$, then there exists a $q' \in P$ such that $\langle q, \sigma \rangle \xrightarrow{}^{+} \langle q', \sigma \rangle$ and $(p_k, q') \in R$ for some $k \geqslant 0$. □

Proof Define $R_S = \{(\langle p, \sigma \rangle, \langle q, \sigma \rangle) \mid (p, q) \in R,\ \sigma \in \Sigma\}$. By Lemma 4.2.6 it follows that R_S satisfies $\langle \textsf{tran} \rangle$ and $\langle \textsf{div} \rangle$. It is clear from its the definition that it also satisfies $\langle \textsf{lab} \rangle$. We have shown before that then it must also satisfy $\langle \textsf{div''} \rangle$. It follows that there exists a $q' \in P$ such that $\langle q, \sigma \rangle \xrightarrow{}^{+} \langle q', \sigma \rangle$ and $(\langle p_k, \sigma \rangle, \langle q', \sigma \rangle) \in R_S$ for some $k \geqslant 0$. From $(\langle p_k, \sigma \rangle, \langle q', \sigma \rangle) \in R_S$ we have $(p_k, q') \in R$. ■

We are now ready for the congruence proof.

Theorem 4.3.7 For all $p, q, \bar{p}, \bar{q} \in P$, if $p \approx_s q$ and $\bar{p} \approx_s \bar{q}$, then

1. $\mathsf{b} :\to p \approx_s \mathsf{b} :\to q$ for all $\mathsf{b} \in \mathsf{B}$,
2. $p \cdot \bar{p} \approx_s q \cdot \bar{q}$,
3. $p + \bar{p} \approx_s q + \bar{q}$,
4. $p^* \approx_s q^*$,
5. $p \parallel \bar{p} \approx_s q \parallel \bar{q}$,
6. $[\![\varsigma \mid p]\!] \approx_s [\![\varsigma \mid q]\!]$ for all valuations $\varsigma \in \Sigma$,
7. $\partial_\Xi(p) \approx_s \partial_\Xi(q)$ for all $\Xi \subseteq \mathsf{A} \setminus \{\tau\}$.
8. $\tau_I(p) \approx_s \tau_I(q)$ for all $I \subseteq \mathsf{A}$. □

Proof All cases are proven in the same fashion. We let R_{pq} and $R_{\bar{p}\bar{q}}$ be two stateless silent bisimulations such that $(p, q) \in R_{pq}$ and $(\bar{p}, \bar{q}) \in R_{\bar{p}\bar{q}}$ satisfy the root conditions in them respectively. Then, using these relations, we construct a symmetric relation R and prove that it is a stateless silent bisimulation and that a desired pair satisfies the root condition in it. When checking the conditions for a stateless silent bisimulation we ignore symmetric cases. For the root condition we only check the condition $\langle \textsf{root-tran}_1 \rangle$ and the implication from left to right of the condition $\langle \textsf{root-term} \rangle$; the verification of the condition $\langle \textsf{root-tran}_2 \rangle$ and of the other implication of $\langle \textsf{root-term} \rangle$

proceed similarly. Note that if a pair in R also satisfies the root condition in R, then it automatically satisfies the conditions $\langle\mathsf{sl\text{-}term}\rangle$, $\langle\mathsf{sl\text{-}tran}\rangle$ and $\langle\mathsf{sl\text{-}div''}\rangle$. By Lemma 4.3.6 it follows that this pair also satisfies $\langle\mathsf{sl\text{-}div}\rangle$, and thus all the conditions for stateless silent bisimulation.

1. Let $R = \{(\mathsf{b} :\rightarrow p, \mathsf{b} :\rightarrow q), (\mathsf{b} :\rightarrow q, \mathsf{b} :\rightarrow p)\} \cup R_{pq}$. It is enough to show that the pair $(\mathsf{b} :\rightarrow p, \mathsf{b} :\rightarrow q)$ satisfies the root condition in R.

   Cond. $\langle\mathsf{root\text{-}term}\rangle$**:** Suppose $\langle \mathsf{b} :\rightarrow p, \sigma\rangle\downarrow$ for some $\sigma \in \Sigma$. Note that Rule $\langle\mathsf{grd\text{-}term}\rangle$ is the final rule of any derivation with $\langle \mathsf{b} :\rightarrow p, \sigma\rangle\downarrow$ as conclusion, so it holds that $\mathsf{check}(\mathsf{b}, \sigma) = \mathit{true}$ and $\langle p, \sigma\rangle\downarrow$. Since (p, q) satisfies the root condition in R_{pq}, we have $\langle q, \sigma\rangle\downarrow$. By Rule $\langle\mathsf{grd\text{-}term}\rangle$, $\langle \mathsf{b} :\rightarrow q, \sigma\rangle\downarrow$.

   Cond. $\langle\mathsf{root\text{-}tran}_1\rangle$**:** Suppose $\langle \mathsf{b} :\rightarrow p, \sigma\rangle \xrightarrow{a} \langle r, \sigma'\rangle$. Rule $\langle\mathsf{grd\text{-}tran}\rangle$ must be the final rule in any derivation of this transition, so it holds that $\mathsf{check}(\mathsf{b}, \sigma) = \mathit{true}$, $\langle p, \sigma\rangle \xrightarrow{a} \langle p', \sigma'\rangle$ and $r = p'$. Since (p, q) satisfies the root condition in R_{pq}, there exists a $q' \in P$ such that $\langle q, \sigma\rangle \xrightarrow{a} \langle q', \sigma'\rangle$ and $(p', q') \in R_{pq}$, and hence, by Rule $\langle\mathsf{grd\text{-}tran}\rangle$, $\langle \mathsf{b} :\rightarrow q, \sigma\rangle \xrightarrow{a} \langle q', \sigma'\rangle$.

2. Let $R = \{(r \cdot \bar{p}, s \cdot \bar{q}) \mid (r, s) \in R_{pq}\} \cup \{(r \cdot \bar{q}, s \cdot \bar{p}) \mid (r, s) \in R_{pq}\} \cup R_{\bar{p}\bar{q}}$.

   Cond. $\langle\mathsf{sl\text{-}term}\rangle$**:** Suppose $\langle r \cdot \bar{p}, \sigma\rangle\downarrow$ for some $\sigma \in \Sigma$. Rule $\langle\mathsf{seq\text{-}term}\rangle$ is the final rule in any derivation of $\langle r \cdot \bar{p}, \sigma\rangle\downarrow$, so it holds that $\langle r, \sigma\rangle\downarrow$ and $\langle \bar{p}, \sigma\rangle\downarrow$. Since $(r, s) \in R_{pq}$, there exists an $s' \in P$ such that $\langle s, \sigma\rangle \twoheadrightarrow \langle s', \sigma\rangle$, $\langle s', \sigma\rangle\downarrow$, and $(r, s') \in R_{pq}$. So by Rule 10, $\langle s \cdot \bar{q}, \sigma\rangle \twoheadrightarrow \langle s' \cdot \bar{q}, \sigma\rangle$. Furthermore, since $(\bar{p}, \bar{q})$ satisfies the root condition in $R_{\bar{p}\bar{q}}$, it follows that $\langle \bar{q}, \sigma\rangle\downarrow$. By Rule $\langle\mathsf{seq\text{-}term}\rangle$, $\langle s' \cdot \bar{q}, \sigma\rangle\downarrow$. Finally, since $(r, s') \in R_{pq}$, according to the definition of R, $(r \cdot \bar{p}, s' \cdot \bar{q}) \in R$.

   Cond. $\langle\mathsf{sl\text{-}tran}\rangle$**:** Suppose $\langle r \cdot \bar{p}, \sigma\rangle \xrightarrow{a} \langle t, \sigma'\rangle$ for some $\sigma, \sigma' \in \Sigma$ and $t \in P$. As before, the final rule of a derivation with this transition as conclusion is either Rule $\langle\mathsf{seq\text{-}tran}_1\rangle$ or Rule $\langle\mathsf{seq\text{-}tran}_2\rangle$ and we treat these cases separately.

   If the final rule applied is Rule $\langle\mathsf{seq\text{-}tran}_1\rangle$, then it holds that $\langle r, \sigma\rangle\downarrow$, $\langle \bar{p}, \sigma\rangle \xrightarrow{a} \langle \bar{p}', \sigma'\rangle$ and $t = \bar{p}'$. Since $(r, s) \in R_{pq}$, there exists an $s' \in P$ such that $\langle s, \sigma\rangle \twoheadrightarrow \langle s', \sigma\rangle$, $\langle s', \sigma\rangle\downarrow$ and $(r, s') \in R_{pq}$. So, by Rule 10, $\langle s \cdot \bar{q}, \sigma\rangle \twoheadrightarrow \langle s' \cdot \bar{q}, \sigma\rangle$. Furthermore, since $(\bar{p}, \bar{q})$ satisfies the root condition in $R_{\bar{p}\bar{q}}$, there exists $\bar{q}'$ such that $\langle \bar{q}, \sigma\rangle \xrightarrow{a} \langle \bar{q}', \sigma'\rangle$ and $(\bar{p}', \bar{q}') \in R_{\bar{p}\bar{q}}$. Now, by Rule $\langle\mathsf{seq\text{-}tran}_1\rangle$, $\langle s' \cdot \bar{q}, \sigma\rangle \xrightarrow{a}$

$\langle \bar{q}', \sigma' \rangle$. Finally, since $(r, s') \in R_{pq}$ and $(\bar{p}', \bar{q}') \in R_{\bar{p}\bar{q}} \subseteq R$, according to the definition of R, $(r \cdot \bar{p}, s' \cdot \bar{q}) \in R$ and $(\bar{p}', \bar{q}') \in R$.

If Rule $\langle \mathsf{seq\text{-}tran}_2 \rangle$ is the final rule applied, then it holds that $\langle r, \sigma \rangle \xrightarrow{a} \langle r', \sigma' \rangle$ and $t = r' \cdot \bar{p}$. Since $(r, s) \in R_{pq}$, there exist $s', s'' \in P$ such that $\langle s, \sigma \rangle \twoheadrightarrow \langle s'', \sigma \rangle \xrightarrow{(a)} \langle s', \sigma' \rangle$, $(r, s'') \in R_{pq}$, and $(r', s') \in R_{pq}$. So, by Rule $\langle \mathsf{seq\text{-}tran}_2 \rangle$, we obtain $\langle s \cdot \bar{q}, \sigma \rangle \twoheadrightarrow \langle s'' \cdot \bar{q}, \sigma \rangle \xrightarrow{(a)} \langle s' \cdot \bar{q}, \sigma' \rangle$. From $(r, s'') \in R_{pq}$, $(r', s') \in R_{pq}$ and the definition of R, we have $(r \cdot \bar{p}, s'' \cdot \bar{q}) \in R$ and $(r' \cdot \bar{p}, s' \cdot \bar{q}) \in R$.

Cond. $\langle \mathsf{sl\text{-}div''} \rangle$**:** Suppose that there exist $\sigma \in \Sigma$ and $t_0, t_1, t_2, \ldots \in P$ such that $t_0 = r \cdot \bar{p}$ and $\langle t_0, \sigma \rangle \rightarrow \langle t_1, \sigma \rangle \rightarrow \langle t_2, \sigma \rangle \rightarrow \cdots$,. From Rules $\langle \mathsf{seq\text{-}tran}_1 \rangle$ and $\langle \mathsf{seq\text{-}tran}_1 \rangle$ it easily follows that either there exist $r_0, r_1, r_2, \ldots \in P$ such that $r_0 = r$, $\langle r_0, \sigma \rangle \rightarrow \langle r_1, \sigma \rangle \rightarrow \langle r_2, \sigma \rangle \rightarrow \cdots$ and $t_i = r_i \cdot \bar{p}$ for all $i \geqslant 0$, or there exist $r_0, r_1, \ldots, r_n, \bar{p}' \in P$ such that $r_0 = r$, $\langle r_0, \sigma \rangle \rightarrow \cdots \rightarrow \langle r_n, \sigma \rangle$, $\langle r_n, \sigma \rangle\downarrow$, $\langle \bar{p}, \sigma \rangle \rightarrow \langle \bar{p}', \sigma \rangle$, $t_i = r_i \cdot \bar{p}$ for $0 \leqslant i \leqslant n$ and $t_{n+1} = \bar{p}' \cdot \bar{p}$.

Suppose first that $\langle r_0, \sigma \rangle \rightarrow \langle r_1, \sigma \rangle \rightarrow \langle r_2, \sigma \rangle \rightarrow \cdots$. Since $(r, s) \in R_{pq}$, by Lemma 4.3.6 there exist $s' \in P$ and $k \geqslant 0$ such that $\langle s, \sigma \rangle \rightarrow^{+} \langle s', \sigma \rangle$ and $(r_k, s') \in R_{pq}$. So by Rule $\langle \mathsf{seq\text{-}tran}_2 \rangle$, $\langle s \cdot \bar{q}, \sigma \rangle \rightarrow^{+} \langle s' \cdot \bar{q}, \sigma \rangle$. Since $(r_k, s') \in R_{pq}$, according to the definition of R, $(r_k \cdot \bar{p}, s' \cdot \bar{q}) \in R$.

Suppose now $\langle r_0, \sigma \rangle \rightarrow \cdots \rightarrow \langle r_n, \sigma \rangle$, $\langle r_n, \sigma \rangle\downarrow$, $\langle \bar{p}, \sigma \rangle \rightarrow \langle \bar{p}', \sigma \rangle$, $t_i = r_i \cdot \bar{p}$ for $0 \leqslant i \leqslant n$ and $t_{n+1} \cdot \bar{p}'$. By Lemma 4.3.5, there exists an $s'' \in P$ such that $(r_n, s'') \in R_{pq}$. Now, since $\langle r_n, \sigma \rangle\downarrow$, there exists an $s' \in P$ such that $\langle s'', \sigma \rangle \twoheadrightarrow \langle s', \sigma \rangle$, $\langle s', \sigma \rangle\downarrow$, and $(r_n, s') \in R_{pq}$. By Rule $\langle \mathsf{seq\text{-}tran}_2 \rangle$, $\langle s \cdot \bar{q}, \sigma \rangle \twoheadrightarrow \langle s'' \cdot \bar{q}, \sigma \rangle \twoheadrightarrow \langle s' \cdot \bar{q}, \sigma \rangle$. Furthermore, since $(\bar{p}, \bar{q})$ satisfies the root condition in $R_{\bar{p}\bar{q}}$, there exists a $\bar{q}' \in P$ such that $\langle \bar{q}, \sigma \rangle \rightarrow \langle \bar{q}', \sigma \rangle$ and $(\bar{p}', \bar{q}') \in R_{\bar{p}\bar{q}}$. Hence, by Rule $\langle \mathsf{seq\text{-}tran}_1 \rangle$, $\langle s' \cdot \bar{q}, \sigma \rangle \rightarrow \langle \bar{q}', \sigma \rangle$. Clearly $\langle s \cdot \bar{q}, \sigma \rangle \rightarrow^{+} \langle \bar{q}', \sigma \rangle$. From $(\bar{p}', \bar{q}') \in R_{\bar{p}\bar{q}}$ it follows that $(\bar{p}', \bar{q}') \in R$.

We now show that $(p \cdot \bar{p}, q \cdot \bar{q})$ satisfies the root condition in R.

Cond. $\langle \mathsf{root\text{-}term} \rangle$**:** Suppose $\langle p \cdot \bar{p}, \sigma \rangle\downarrow$ for some $\sigma \in \Sigma$. Since in any derivation with $\langle p \cdot \bar{p}, \sigma \rangle\downarrow$ as conclusion Rule 8 is the final rule applied, it follows that $\langle p, \sigma \rangle\downarrow$ and $\langle \bar{p}, \sigma \rangle\downarrow$. Since (p, q) and $(\bar{p}, \bar{q})$ satisfy the root condition in R_{pq} and $R_{\bar{p}\bar{q}}$ respectively, we obtain $\langle q, \sigma \rangle\downarrow$ and $\langle \bar{q}, \sigma \rangle\downarrow$, and hence, by Rule $\langle \mathsf{seq\text{-}term} \rangle$, $\langle q \cdot \bar{q}, \sigma \rangle\downarrow$.

Cond. $\langle$root-tran$_1\rangle$**:** Suppose $\langle p \cdot \bar{p}, \sigma\rangle \xrightarrow{a} \langle r, \sigma'\rangle$ for some $\sigma, \sigma' \in \Sigma$ and $r \in P$. The final rule of a derivation of this transition is either Rule $\langle$seq-tran$_1\rangle$ or Rule $\langle$seq-tran$_1\rangle$; we treat these cases separately.

If the final rule applied is Rule $\langle$seq-tran$_1\rangle$, then $\langle p, \sigma\rangle\downarrow$, $\langle \bar{p}, \sigma\rangle \xrightarrow{a} \langle \bar{p}', \sigma'\rangle$ and $r = \bar{p}'$. Since (p, q) satisfies the root condition in R_{pq}, we have $\langle q, \sigma\rangle\downarrow$. Moreover, since $(\bar{p}, \bar{q})$ satisfies the root condition in $R_{\bar{p}\bar{q}}$, there exists $\bar{q}' \in P$ such that $\langle \bar{q}, \sigma\rangle \xrightarrow{a} \langle \bar{q}', \sigma'\rangle$ and $(\bar{r}', \bar{s}') \in R_{\bar{p}\bar{q}}$. So, by Rule $\langle$seq-tran$_1\rangle$, $\langle q \cdot \bar{q}, \sigma\rangle \xrightarrow{a} \langle \bar{q}', \sigma'\rangle$.

If the final rule applied is Rule $\langle$seq-tran$_2\rangle$, then $\langle p, \sigma\rangle \xrightarrow{a} \langle p', \sigma'\rangle$ and $r = p' \cdot \bar{p}$. Since (p, q) satisfies the root condition in R_{pq}, there exists $q' \in P$ such that $\langle q, \sigma\rangle \xrightarrow{a} \langle q', \sigma'\rangle$ and $(p', q') \in R_{pq}$. By Rule $\langle$seq-tran$_2\rangle$, $\langle q \cdot \bar{q}, \sigma\rangle \xrightarrow{a} \langle q' \cdot \bar{q}, \sigma'\rangle$. Since $(p', q') \in R_{pq}$, according to the definition of R, $(p' \cdot \bar{p}, q' \cdot \bar{q}) \in R$.

3. Let $R = R' \cup R_{pq} \cup R_{\bar{p}\bar{q}}$ where

$$R' = \left\{ (r + \bar{r}, s + \bar{s}) \mid \begin{array}{l} (r, s) \in R_{pq} \text{ and } (\bar{r}, \bar{s}) \in R_{\bar{p}\bar{q}} \text{ satisfy the root} \\ \text{condition in } R_{pq} \text{ and } R_{\bar{p}\bar{q}} \text{ respectively} \end{array} \right\}.$$

It is enough to show that every pair from R' satisfies the root condition in R.

Cond. $\langle$root-term$\rangle$**:** Suppose $\langle r + \bar{r}, \sigma\rangle\downarrow$ for some $\sigma \in \Sigma$. Because Rule $\langle$alt-term$\rangle$ is the final rule of any derivation of $\langle r + \bar{r}, \sigma\rangle\downarrow$, it holds that $\langle r, \sigma\rangle\downarrow$ or $\langle \bar{r}, \sigma\rangle\downarrow$. We only consider the case when $\langle r, \sigma\rangle\downarrow$; when $\langle \bar{r}, \sigma\rangle\downarrow$ the proof is similar. Since (r, s) satisfies the root condition in R_{pq}, it follows that $\langle s, \sigma\rangle\downarrow$, and hence, by Rule $\langle$alt-term$\rangle$, that $\langle s + \bar{s}, \sigma\rangle\downarrow$.

Cond. $\langle$root-tran$_1\rangle$**:** Suppose $\langle r + \bar{r}, \sigma\rangle \xrightarrow{a} \langle t, \sigma'\rangle$ for some $\sigma, \sigma' \in \Sigma$ and $t \in P$. Since Rule $\langle$alt-tran$\rangle$ must be the final rule of any derivation of this transition, it holds that $\langle r, \sigma\rangle \xrightarrow{a} \langle r', \sigma'\rangle$ and $t = r'$ or that $\langle \bar{r}, \sigma\rangle \xrightarrow{a} \langle \bar{r}', \sigma'\rangle$ and $t = \bar{r}'$. Suppose $\langle r, \sigma\rangle \xrightarrow{a} \langle r', \sigma'\rangle$ (the proof of the other case is similar). Since (r, s) satisfies the root condition in R_{pq}, there exists $s' \in P$ such that $\langle s, \sigma\rangle \xrightarrow{a} \langle s', \sigma'\rangle$ and $(r', s') \in R_{pq}$. Now, by Rule $\langle$alt-tran$\rangle$, $\langle s + \bar{s}, \sigma\rangle \xrightarrow{a} \langle s', \sigma'\rangle$. Since $(r', s') \in R_{pq}$, according to the definition of R, $(r', s') \in R$.

4. Let $R = \{(p^*, q^*), (q^*, p^*)\} \cup R'$ where

$$R' = \{(r \cdot p^*, s \cdot q^*) \mid (r, s) \in R_{pq}\} \cup \{(r \cdot q^*, s \cdot p^*) \mid (r, s) \in R_{pq}\}.$$

For the pair (p^*, q^*) it is enough to show that it satisfies the root condition in $R' \subseteq R$ and thus in R too. By Rule ⟨rep-term⟩, the condition ⟨root-term⟩ holds trivially. For ⟨root-tran$_1$⟩, suppose $\langle p^*, \sigma\rangle \xrightarrow{a} \langle r, \sigma'\rangle$ for some $\sigma, \sigma' \in \Sigma$ and $r \in P$. Since in any derivation with this transition Rule ⟨rep-tran⟩ is the final rule applied, it follows that $\langle p, \sigma\rangle \xrightarrow{a} \langle p', \sigma'\rangle$ and $r = p' \cdot p^*$. Since (p, q) satisfies the root condition in R_{pq}, there exists q' such that $\langle q, \sigma\rangle \xrightarrow{a} \langle q', \sigma'\rangle$ and $(p', q') \in R_{pq}$. By Rule ⟨rep-tran⟩, $\langle q^*, \sigma\rangle \xrightarrow{a} \langle q' \cdot q^*, \sigma'\rangle$. Since $(p', q') \in R_{pq}$, according to the definition of R', $(p' \cdot p^*, q' \cdot q^*) \in R'$. We conclude that (p^*, q^*) satisfies the root condition in R'.

We now show that R' also satisfies the conditions ⟨sl-term⟩, ⟨sl-tran⟩ and ⟨sl-div''⟩.

Cond. ⟨sl-term⟩**:** Suppose $\langle r \cdot p^*, \sigma\rangle\downarrow$ for some $\sigma \in \Sigma$. Rule ⟨seq-term⟩ is the final rule applied in any derivation of $\langle r \cdot p^*, \sigma\rangle\downarrow$, and since $\langle p^*, \sigma\rangle\downarrow$, by Rule ⟨rep-term⟩ it follows that $\langle r, \sigma\rangle\downarrow$. Since $(r, s) \in R_{pq}$, there exists an $s' \in P$ $\langle s, \sigma\rangle \twoheadrightarrow \langle s', \sigma\rangle$, $\langle s', \sigma\rangle\downarrow$, and $(r, s') \in R_{pq}$. By Rule ⟨seq-tran$_2$⟩, we now obtain $\langle s \cdot q^*, \sigma\rangle \twoheadrightarrow \langle s' \cdot q^*, \sigma\rangle$. By Rules ⟨seq-term⟩ and ⟨rep-term⟩, $\langle s' \cdot q^*, \sigma\rangle\downarrow$. Since $(r, s') \in R_{pq}$, according to the definition of R, $(r \cdot p^*, s' \cdot q^*) \in R$.

Cond. ⟨sl-tran⟩**:** Suppose $\langle r \cdot p^*, \sigma\rangle \xrightarrow{a} \langle t, \sigma'\rangle$ for some $\sigma, \sigma' \in \Sigma$ and $t \in P$. Note that the final rule of a derivation with this transition as conclusion is either Rule ⟨seq-tran$_1$⟩ or Rule ⟨seq-tran$_2$⟩; we treat these cases separately.

If the final rule applied is Rule ⟨seq-tran$_1$⟩, then it holds that $\langle r, \sigma\rangle\downarrow$ and $\langle p^*, \sigma\rangle \xrightarrow{a} \langle t, \sigma'\rangle$. Since $(r, s) \in R_{pq}$, there exists an $s' \in P$ such that $\langle s, \sigma\rangle \twoheadrightarrow \langle s', \sigma\rangle$, $\langle s', \sigma\rangle\downarrow$, and $(r, s') \in R_{pq}$. Applying Rule ⟨seq-tran$_2$⟩, $\langle s \cdot q^*, \sigma\rangle \twoheadrightarrow \langle s' \cdot q^*, \sigma\rangle$. Furthermore, since Rule ⟨rep-tran⟩ is the final rule of a derivation with $\langle p^*, \sigma\rangle \xrightarrow{a} \langle t, \sigma'\rangle$ as conclusion, we have $\langle p, \sigma\rangle \xrightarrow{a} \langle p', \sigma'\rangle$ and $t = p' \cdot p^*$. Since (p, q) satisfies the root condition in R_{pq}, there exists q' such that $\langle q, \sigma\rangle \xrightarrow{a} \langle q', \sigma'\rangle$ and $(p', q') \in R_{pq}$. Applying Rule ⟨rep-tran⟩, we obtain $\langle q^*, \sigma\rangle \xrightarrow{a} \langle q' \cdot q^*, \sigma'\rangle$. Since $\langle s', \sigma\rangle\downarrow$, by Rule ⟨seq-tran$_1$⟩, we have $\langle s' \cdot q^*, \sigma\rangle \xrightarrow{a} \langle q' \cdot q^*, \sigma'\rangle$. Finally, because $(r, s') \in R_{pq}$ and $(p', q') \in R_{pq}$, by the definition of R, $(r \cdot p^*, s' \cdot q^*) \in R$ and $(t, q' \cdot q^*) \in R$.

If Rule $\langle\mathsf{seq\text{-}tran_2}\rangle$ is the final rule applied, then $\langle r,\sigma\rangle \xrightarrow{a} \langle r',\sigma'\rangle$ and $t = r'\cdot p^*$. Since $(r,s)\in R_{pq}$, there exist $s',s''\in P$ such that $\langle s,\sigma\rangle \twoheadrightarrow \langle s'',\sigma\rangle \xrightarrow{(a)} \langle s',\sigma'\rangle$, $(r,s'')\in R_{pq}$ and $(r',s')\in R_{pq}$. Now, by Rule $\langle\mathsf{seq\text{-}tran_2}\rangle$, we obtain $\langle s\cdot q^*,\sigma\rangle \twoheadrightarrow \langle s''\cdot q^*,\sigma\rangle \xrightarrow{(a)} \langle s'\cdot q^*,\sigma'\rangle$. Since $(r,s'')\in R_{pq}$ and $(r',s')\in R_{pq}$, according to the definition of R, $(r\cdot p^*, s''\cdot q^*)\in R$ and $(r'\cdot p^*, s'\cdot q^*)\in R$.

Cond. $\langle\mathsf{sl\text{-}div''}\rangle$**:** Since (p^*,q^*) satisfies the root condition in R', the proof that R satisfies $\langle\mathsf{sl\text{-}div''}\rangle$ is essentially the same as in the case of sequential composition.

5. Let $R=\{(r\parallel\bar{r}, s\parallel\bar{s}) \mid (r,s)\in R_{pq}, (\bar{r},\bar{s})\in R_{\bar{p}\bar{q}}\}$. By Lemma 4.2.11, R is a stateless silent bisimulation. We only need to show that $(p\parallel\bar{p}, q\parallel\bar{q})$ satisfies the root condition in it.

Cond. $\langle\mathsf{root\text{-}term}\rangle$**:** Suppose $\langle p\parallel\bar{p},\sigma\rangle\downarrow$ for some $\sigma\in\Sigma$. Because Rule $\langle\mathsf{par\text{-}term}\rangle$ is the final rule of any derivation with $\langle p\parallel\bar{p},\sigma\rangle\downarrow$ as conclusion, we have $\langle p,\sigma\rangle\downarrow$ and $\langle\bar{p},\sigma\rangle\downarrow$. Since (p,q) and $(\bar{p},\bar{q})$ satisfy the root condition in R_{pq} and $R_{\bar{p}\bar{q}}$ respectively, we obtain $\langle q,\sigma\rangle\downarrow$ and $\langle\bar{q},\sigma\rangle\downarrow$, and hence, by Rule $\langle\mathsf{par\text{-}term}\rangle$, $\langle q\parallel\bar{q},\sigma\rangle\downarrow$.

Cond. $\langle\mathsf{root\text{-}term}\rangle$**:** Suppose $\langle p\parallel\bar{p},\sigma\rangle \xrightarrow{a} \langle r,\sigma'\rangle$ for some $\sigma,\sigma'\in\Sigma$ and $r\in P$. The final rule of any derivation with this transition as conclusion is either Rule $\langle\mathsf{par\text{-}tran_1}\rangle$ or Rule $\langle\mathsf{par\text{-}tran_2}\rangle$; we treat these cases separately.

If the final rule applied is Rule $\langle\mathsf{par\text{-}tran_1}\rangle$, then $\langle p,\sigma\rangle \xrightarrow{a} \langle p',\sigma'\rangle$ and $t = p'\parallel\bar{p}$; (or symmetrically $\langle\bar{p},\sigma\rangle \xrightarrow{a} \langle\bar{p}',\sigma'\rangle$ and $t = p\parallel\bar{p}'$). Since (p,q) satisfies the root condition in R_{pq}, there exists a $q'\in P$ such that $\langle q,\sigma\rangle \xrightarrow{a} \langle q',\sigma'\rangle$ and $(p',q')\in R_{pq}$. Hence, by Rule $\langle\mathsf{par\text{-}tran_1}\rangle$, $\langle q\parallel\bar{q},\sigma\rangle \xrightarrow{a} \langle q'\parallel\bar{q},\sigma'\rangle$. Since $(p',q')\in R_{pq}$, according to the definition of R, $(p'\parallel\bar{p}, q'\parallel\bar{q})\in R$.

If the final rule applied is Rule $\langle\mathsf{par\text{-}tran_2}\rangle$, then $\langle p,\sigma\rangle \xrightarrow{b} \langle p',\sigma''\rangle$, $\langle\bar{p},\sigma\rangle \xrightarrow{c} \langle\bar{p}',\sigma'''\rangle$, $r = p'\parallel\bar{p}'$, and $a = \mathsf{act}(\mathsf{comm}(b,c),\sigma)$ for some $b,c\in\mathsf{A}$ and $\sigma' = \mathsf{eff}(\mathsf{comm}(b,c),\sigma) = \sigma''\ll\sigma'''_{/\mathrm{dom}(\sigma''')\setminus\mathrm{dom}(\sigma'')} = \sigma'''\ll\sigma''_{/\mathrm{dom}(\sigma'')\setminus\mathrm{dom}(\sigma''')}$. Since (p,q) and $(\bar{p},\bar{q})$ satisfy the root condition in R_{pq} and $R_{\bar{p}\bar{q}}$ respectively, there exist $q',\bar{q}'\in P$ such that $\langle q,\sigma\rangle \xrightarrow{b} \langle q',\sigma''\rangle$, $\langle\bar{q},\sigma\rangle \xrightarrow{c} \langle\bar{q}',\sigma'''\rangle$, and $(p',q')\in R_{pq}$ and $(\bar{p}',\bar{q}')\in R_{\bar{p}\bar{q}}$. Now, by Rule $\langle\mathsf{par\text{-}tran_2}\rangle$, $\langle q\parallel\bar{q},\sigma\rangle \xrightarrow{a} \langle q'\parallel\bar{q}',\sigma'\rangle$. Since $(p',q')\in R_{pq}$ and $(\bar{p}',\bar{q}')\in R_{\bar{p}\bar{q}}$, according to the definition of R, $(p'\parallel\bar{p}', q'\parallel\bar{q}')\in R$.

6. Let $R = \{(\llbracket\varsigma \mid r\rrbracket, \llbracket\varsigma \mid s\rrbracket) \mid \varsigma \in \Sigma,\ (r,s) \in R_{pq}\}$. From Lemmas 4.1.3 and 4.2.7 it follows that R is a stateless silent bisimulation. We only prove that the pair $(\llbracket\varsigma \mid p\rrbracket, \llbracket\varsigma \mid q\rrbracket)$ satisfies the root condition in it.

   Cond. $\langle$root-term$\rangle$**:** Suppose $\langle\llbracket\varsigma \mid p\rrbracket, \sigma\rangle\downarrow$ for some $\sigma \in \Sigma$. Because Rule $\langle$scp-term$\rangle$ is the final rule of any derivation with $\langle\llbracket\varsigma \mid p\rrbracket, \sigma\rangle\downarrow$ as conclusion, it holds that $\langle p, \sigma \ll \varsigma\rangle\downarrow$. Since (p,q) satisfies the root condition in R_{pq}, $\langle q, \sigma \ll \varsigma\rangle\downarrow$. By Rule $\langle$scp-term$\rangle$, $\langle\llbracket\varsigma \mid q\rrbracket, \sigma\rangle\downarrow$.

   Cond. $\langle$root-tran$_1\rangle$**:** Suppose $\langle\llbracket\varsigma \mid p\rrbracket, \sigma\rangle \xrightarrow{a} \langle t, \sigma'\rangle$ for some $\sigma, \sigma' \in \Sigma$ and $t \in P$. Since Rule $\langle$scp-tran$\rangle$ is the final rule of any derivation with this transition as conclusion, we have that there exist $p' \in P$ and $\sigma'' \in \Sigma$ such that $\langle p, \sigma \ll \varsigma\rangle \xrightarrow{a} \langle p', \sigma''\rangle$, $\sigma' = \sigma''_{/\mathrm{dom}(\sigma)\setminus\mathrm{dom}(\varsigma)}$ and $t = \llbracket\varsigma' \mid p'\rrbracket$ for $\varsigma' = \sigma''_{/\mathrm{dom}(\varsigma)}$. Since (p,q) satisfies the root condition in R_{pq}, there exists a $q' \in P$ such that $\langle q, \sigma \ll \varsigma\rangle \xrightarrow{a} \langle q', \sigma''\rangle$ and $(p',q') \in R_{pq}$. By Rule $\langle$scp-tran$\rangle$, $\langle\llbracket\varsigma \mid q\rrbracket, \sigma\rangle \xrightarrow{a} \langle\llbracket\varsigma' \mid q'\rrbracket, \sigma'\rangle$. Since $(p',q') \in R_{pq}$, according to the definition of R, $(\llbracket\varsigma' \mid p'\rrbracket, \llbracket\varsigma' \mid q'\rrbracket) \in R$.

7. Let $R = \{(\partial_\Xi(r), \partial_\Xi(s)) \mid (r,s) \in R_{pq}\}$. From Lemmas 4.1.4 and 4.2.7 it follows that R is a stateless silent bisimulation. We only prove that the pair $(\partial_\Xi(p), \partial_\Xi(q))$ satisfies the root condition in it.

   Cond. $\langle$root-term$\rangle$**:** Suppose $\langle\partial_\Xi(p), \sigma\rangle\downarrow$ for some $\sigma \in \Sigma$. Because Rule $\langle$enc-term$\rangle$ is the final rule of any derivation with $\langle\partial_\Xi(p), \sigma\rangle\downarrow$ as conclusion, it holds that $\langle p, \sigma\rangle\downarrow$. Since (p,q) satisfies the root condition in R_{pq}, $\langle q, \sigma\rangle\downarrow$. By Rule $\langle$enc-term$\rangle$, $\langle\partial_\Xi(q), \sigma\rangle\downarrow$.

   Cond. $\langle$root-tran$_1\rangle$**:** Suppose $\langle\partial_\Xi(p), \sigma\rangle \xrightarrow{a} \langle t, \sigma'\rangle$ for some $\sigma, \sigma' \in \Sigma$ and $t \in P$. Since Rule $\langle$enc-tran$\rangle$ is the final rule of any derivation with this transition as conclusion, we have that $a \notin \Xi$ and that there exist $p' \in P$ such that $\langle p, \sigma\rangle \xrightarrow{a} \langle p', \sigma'\rangle$ and $t = \partial_\Xi(p')$. Since (p,q) satisfies the root condition in R_{pq}, there exists a $q' \in P$ such that $\langle q, \sigma\rangle \xrightarrow{a} \langle q', \sigma'\rangle$ and $(p',q') \in R_{pq}$. By Rule $\langle$enc-tran$\rangle$, $\langle\partial_\Xi(p), \sigma\rangle \xrightarrow{a} \langle\partial_\Xi(p'), \sigma'\rangle$. Since $(p',q') \in R_{pq}$, according to the definition of R, $(\partial_\Xi(p'), \partial_\Xi(q')) \in R$. ■

8. Let $R = \{(\tau_I(r), \tau_I(s)) \mid (r,s) \in R_{pq}\}$. From Lemmas 4.1.5 and 4.2.7 it follows that R is a stateless silent bisimulation. We only prove that the pair $(\tau_I(p), \tau_I(q))$ satisfies the root condition in it.

Cond. $\langle$root-term$\rangle$**:** Suppose $\langle \tau_I(p), \sigma\rangle\downarrow$ for some $\sigma \in \Sigma$. Because Rule $\langle$hide-term$\rangle$ is the final rule of any derivation with $\langle \tau_I(p), \sigma\rangle\downarrow$ as conclusion, it holds that $\langle p, \sigma\rangle\downarrow$. Since (p, q) satisfies the root condition in R_{pq}, $\langle q, \sigma\rangle\downarrow$. By Rule $\langle$hide-term$\rangle$, $\langle \tau_I(q), \sigma\rangle\downarrow$.

Cond. $\langle$root-tran$_1\rangle$**:** Suppose $\langle \tau_I(p), \sigma\rangle \xrightarrow{a} \langle t, \sigma'\rangle$ for some $\sigma, \sigma' \in \Sigma$ and $t \in P$. Since Rule $\langle$hide-tran$_1\rangle$ or Rule $\langle$hide-tran$_2\rangle$ is the final rule of any derivation with this transition as conclusion, we have that there exist $p' \in P$ such that $\langle p, \sigma\rangle \xrightarrow{a} \langle p', \sigma'\rangle$ and $t = \tau_I(p')$. Since (p, q) satisfies the root condition in R_{pq}, there exists a $q' \in P$ such that $\langle q, \sigma\rangle \xrightarrow{a} \langle q', \sigma'\rangle$ and $(p', q') \in R_{pq}$. By Rule $\langle$hide-tran$_1\rangle$ and $\langle$hide-tran$_2\rangle$, $\langle \tau_I(p), \sigma\rangle \xrightarrow{a} \langle \tau_I(p'), \sigma'\rangle$. Since $(p', q') \in R_{pq}$, according to the definition of R, $(\tau_I(p'), \tau_I(q')) \in R$.

Chapter 5

Timed Silent Congruence

In the introduction we explained the importance of timing in the modeling of dynamic systems. In this chapter we broaden our theory to deal with this aspect. As said before, we take the simplest version of timing, i.e. we take timing to be discrete, relative and independent of actions.

We first extend the notion of doubly-labeled transition systems with a time-transition relation $\mapsto$ that represents the progress of time to the next time slice. Then we incorporate timing into the language κ by following the approach similar to [97] and [20]. Finally, we define timed stateless silent bisimulation and show that it is a congruence relation.

Definition 5.0.8 (Timed doubly-labeled transition system) Let A be a set of *actions* and let Π be a set of atomic propositions. A *timed doubly-labeled transition system* is a quintuple $(S, \rightarrow, \mapsto, \downarrow, \ell)$ where:

- $(S, \rightarrow, \downarrow, \ell) \in \mathcal{T}_{\mathsf{A},\Pi}$, and
- $\mapsto \subseteq S \times S$ is the *time-transition relation*.

We will write $s \overset{\Delta}{\mapsto} s'$ instead of $(s, s') \in \mapsto$. The set of all timed doubly-labeled transition systems with the set of actions A and the set of state labels Π is denoted $\mathcal{T}^{\Delta}_{\mathsf{A},\Pi}$. □

We now add timing to the specification language κ. The new language is called Timed κ and is interpreted over timed doubly-labeled transition systems. We assume that the set D of data values is the set $\mathbb{N}$ of natural numbers (with 0). We also assume that expressions in E are built using standard operators, like $+$, $-$, etc., that are correctly evaluated by a valuation.

We extend the set of atomic processes with the *delay* process Δe where $e \in E$. This process delays n time units, where $n \in \mathbb{N}$ is the value of the expression e in the current valuation, and terminates (c.f. Table 5.1). To obtain full generality, we extend the domain of the eff function to $(\mathsf{A} \cup \{\Delta\}) \times \Sigma$ and consequently allow a tick to change the valuation. We, however, require that $|\mathsf{eff}(\Delta, \sigma)| = 1$. This is to ensure that Timed κ satisfies the so-called *time determinism* property, i.e. the property that time does not make a choice.

$$\frac{\sigma(e) = 0}{\langle \Delta e, \sigma \rangle \downarrow} \langle \mathsf{delay\text{-}term} \rangle \qquad \frac{\sigma(e) = n \in \mathbb{N},\ n \geqslant 1,\ \mathsf{eff}(\Delta, \sigma) = \{\sigma'\}}{\langle \Delta e, \sigma \rangle \overset{\Delta}{\mapsto} \langle \Delta(e-1), \sigma' \rangle} \langle \mathsf{delay\text{-}tick} \rangle$$

Table 5.1: Operational semantics for the delay process in Timed κ

In Table 5.2 we present the operational rules for Timed κ operators to deal with timing. For guarded processes, scopes, and repetition, time transitions are just like action transitions. Also, as expected, the encapsulation and the hiding operator have no effect on timing transitions. The rules for sequential, alternative and parallel composition require more explanation.

We start with the alternative composition operator. Rule $\langle \mathsf{alt\text{-}tick}_2 \rangle$ ensures that Timed κ satisfies the time determinism property by requiring that processes in a non-deterministic choice always delay together. Rule $\langle \mathsf{alt\text{-}tick}_1 \rangle$ describes the case when only one of the processes can delay. In this case time is allowed to make a choice.

Recall that, when action behavior is concerned, if the first process in a sequential composition terminates, then the whole composition can do either an action from the first or from the second process. This is a non-determinism hidden in a sequential composition. Since we also want the time determinism property here, we distinguish four cases [97]. The first case, demonstrated in Rule $\langle \mathsf{seq\text{-}tick}_1 \rangle$, is the case when the first element of a sequential composition cannot terminate. In this case, time transitions behave as action transitions. Rules $\langle \mathsf{seq\text{-}tick}_2 \rangle$, $\langle \mathsf{seq\text{-}tick}_3 \rangle$ and $\langle \mathsf{seq\text{-}tick}_4 \rangle$ describe the situations in which the first process terminates. In Rule $\langle \mathsf{seq\text{-}tick}_2 \rangle$, the first process also ticks but the second one does not, and so its conclusion is based on the same idea described in Rule $\langle \mathsf{alt\text{-}tick}_2 \rangle$. Similar situation appears in Rule $\langle \mathsf{seq\text{-}tick}_3 \rangle$ where the second process ticks but the first does not. In this case, the sequential composition continues as the second process. Finally, in Rule $\langle \mathsf{seq\text{-}tick}_4 \rangle$ we have a hidden non-deterministic choice in which both alternatives delay. Following the time-factorization principle the processes delay together and the actual choice between them is postponed.

Parallel processes are expected to always delay together and so we have Rule $\langle \mathsf{par\text{-}tick}_2 \rangle$. In the case when one component can delay, the other cannot, but it can terminate, we let the parallel composition tick and then proceed as the first component. Rule $\langle \mathsf{par\text{-}tick}_1 \rangle$ appears in [97] and [20] but has been recently dropped in [8]. We keep it here only to show that our results hold for more modeling languages.

5.1 Timed silent congruence

In this section we define the notion of timed silent congruence as an extension of silent congruence with timing. We have already mentioned in the introduction that, for the setting of labeled transition systems without termination, timed branching bisimulation has been defined and shown to satisfy all the desired properties. Therefore, the easiest way to define timed silent bisimulation is to suitably adapt the timing condition from there.

Suppose we define a timed stateless silent bisimulation R as a stateless silent bisimulation such that for all $(p, q) \in R$ and all $\sigma \in \Sigma$ the following holds:

- if $\langle p, \sigma \rangle \overset{\Delta}{\mapsto} \langle p', \sigma' \rangle$, then there exist $q', q'' \in P$ such that

$$\langle q, \sigma \rangle \twoheadrightarrow \langle q'', \sigma \rangle \overset{\Delta}{\mapsto} \langle q', \sigma' \rangle,\ (p, q'') \in R \text{ and } (p', q') \in R.$$

This definition is proposed in [97, 7, 6]. With a suitable root condition it works well in the setting without successful termination. However, in our case, it leads to a bisimilarity that is not a congruence for sequential and parallel composition. The following example illustrates where the problems are.

Example 5.1.1 Assume that $\mathsf{eff}(\boldsymbol{\tau}, \sigma) = \{\sigma\}$ for all $\sigma \in \Sigma$.

a. The processes $\Delta 1 + \varepsilon$ and $\boldsymbol{\tau} \cdot (\Delta 1 + \varepsilon) + \Delta 1$ are timed stateless silently bisimilar. However, when followed in a sequential composition by the process $\Delta 1 \cdot \boldsymbol{a}$ they start exhibiting a different behavior. The left-hand side process does a tick (in any valuation) and then can choose whether to tick again or execute the action a. The right-hand side process can also do a tick initially, but then it has to do another tick without the option to execute a.

   The problem lies in the fact that the right-hand side process does not have an option to terminate initially. This makes it a subject to

$$\frac{\mathsf{check}(\mathsf{b},\sigma) = true,\ \langle p,\sigma\rangle \overset{\Delta}{\mapsto} \langle p',\sigma'\rangle}{\langle \mathsf{b} :\rightarrow p,\sigma\rangle \overset{\Delta}{\mapsto} \langle p',\sigma'\rangle}\ \langle\mathsf{grd\text{-}tick}\rangle$$

$$\frac{\langle p,\sigma\rangle \overset{\Delta}{\mapsto} \langle p',\sigma'\rangle,\ \langle p,\sigma\rangle\not\downarrow}{\langle p\cdot q,\sigma\rangle \overset{\Delta}{\mapsto} \langle p'\cdot q,\sigma'\rangle}\ \langle\mathsf{seq\text{-}tick}_1\rangle$$

$$\frac{\langle p,\sigma\rangle\downarrow,\ \langle p,\sigma\rangle \overset{\Delta}{\mapsto} \langle p',\sigma'\rangle,\ \langle q,\sigma\rangle\overset{\Delta}{\not\mapsto}}{\langle p\cdot q,\sigma\rangle \overset{\Delta}{\mapsto} \langle p'\cdot q,\sigma'\rangle}\ \langle\mathsf{seq\text{-}tick}_2\rangle$$

$$\frac{\langle p,\sigma\rangle\downarrow,\ \langle p,\sigma\rangle\overset{\Delta}{\not\mapsto},\ \langle q,\sigma\rangle \overset{\Delta}{\mapsto} \langle q',\sigma'\rangle}{\langle p\cdot q,\sigma\rangle \overset{\Delta}{\mapsto} \langle q',\sigma'\rangle}\ \langle\mathsf{seq\text{-}tick}_3\rangle$$

$$\frac{\langle p,\sigma\rangle\downarrow,\ \langle p,\sigma\rangle \overset{\Delta}{\mapsto} \langle p',\sigma'\rangle,\ \langle q,\sigma\rangle \overset{\Delta}{\mapsto} \langle q',\sigma'\rangle}{\langle p\cdot q,\sigma\rangle \overset{\Delta}{\mapsto} \langle p'\cdot q + q',\sigma'\rangle}\ \langle\mathsf{seq\text{-}tick}_4\rangle$$

$$\frac{\langle p,\sigma\rangle \overset{\Delta}{\mapsto} \langle p',\sigma'\rangle,\ \langle q,\sigma\rangle\overset{\Delta}{\not\mapsto}}{\langle p+q,\sigma\rangle \overset{\Delta}{\mapsto} \langle p',\sigma'\rangle,\ \langle q+p,\sigma\rangle \overset{\Delta}{\mapsto} \langle p',\sigma'\rangle}\ \langle\mathsf{alt\text{-}tick}_1\rangle$$

$$\frac{\langle p,\sigma\rangle \overset{\Delta}{\mapsto} \langle p',\sigma'\rangle,\ \langle q,\sigma\rangle \overset{\Delta}{\mapsto} \langle q',\sigma'\rangle}{\langle p+q,\sigma\rangle \overset{\Delta}{\mapsto} \langle p'+q',\sigma'\rangle}\ \langle\mathsf{alt\text{-}tick}_2\rangle \qquad \frac{\langle p,\sigma\rangle \overset{\Delta}{\mapsto} \langle p',\sigma'\rangle}{\langle p^*,\sigma\rangle \overset{\Delta}{\mapsto} \langle p'\cdot p^*,\sigma'\rangle}\ \langle\mathsf{rep\text{-}tick}\rangle$$

$$\frac{\langle p,\sigma\rangle \overset{\Delta}{\mapsto} \langle p',\sigma'\rangle,\ \langle q,\sigma\rangle\downarrow,\ \langle q,\sigma\rangle\overset{\Delta}{\not\mapsto}}{\langle p \parallel q,\sigma\rangle \overset{\Delta}{\mapsto} \langle p',\sigma'\rangle,\ \langle q \parallel p,\sigma\rangle \overset{\Delta}{\mapsto} \langle p',\sigma'\rangle}\ \langle\mathsf{par\text{-}tick}_1\rangle$$

$$\frac{\langle p,\sigma\rangle \overset{\Delta}{\mapsto} \langle p',\sigma'\rangle,\ \langle q,\sigma\rangle \overset{\Delta}{\mapsto} \langle q',\sigma'\rangle}{\langle p \parallel q,\sigma\rangle \overset{\Delta}{\mapsto} \langle p' \parallel q',\sigma'\rangle}\ \langle\mathsf{par\text{-}tick}_2\rangle$$

$$\frac{\langle p,\sigma \ll \varsigma\rangle \overset{\Delta}{\mapsto} \langle p',\sigma'\rangle}{\langle [\![\varsigma \mid p]\!],\sigma\rangle \overset{\Delta}{\mapsto} \langle [\![\sigma'_{/\mathrm{dom}(\varsigma)} \mid p']\!],\sigma \ll \sigma'_{/\mathrm{dom}(\sigma')\setminus\mathrm{dom}(\varsigma)}\rangle}\ \langle\mathsf{scp\text{-}tick}\rangle$$

$$\frac{\langle p,\sigma\rangle \overset{\Delta}{\mapsto} \langle p',\sigma'\rangle}{\langle \partial_\Xi(p),\sigma\rangle \overset{\Delta}{\mapsto} \langle \partial_\Xi(p'),\sigma'\rangle}\ \langle\mathsf{enc\text{-}tick}\rangle \qquad \frac{\langle p,\sigma\rangle \overset{\Delta}{\mapsto} \langle p',\sigma'\rangle}{\langle \tau_H(p),\sigma\rangle \overset{\Delta}{\mapsto} \langle \tau_H(p'),\sigma'\rangle}\ \langle\mathsf{hide\text{-}tick}\rangle$$

Table 5.2: Operational semantics for Timed κ – composed processes

Rule $\langle\mathsf{seq\text{-}tick}_1\rangle$, leading to a behavior that the left-hand side cannot simulate.

b. The processes $\Delta 1 + \varepsilon$ and $\boldsymbol{\tau} \cdot (\Delta 1 + \varepsilon) + \varepsilon$ are timed stateless silently bisimilar. As in the previous example, when composed sequentially with the process $\Delta 1 \cdot \boldsymbol{a}$, the two processes behave differently. The left-hand side process must do a *tick* and then choose whether to tick again or execute the action a. The right-hand side process can also do a tick initially but then it has to do another tick without the option to execute a.

   Similarly, when composed in parallel with the process $\Delta 1 \cdot \boldsymbol{a}$, the two processes behave differently. The process $\Delta 1 + \varepsilon \parallel \Delta 1 \cdot \boldsymbol{a}$ can only tick, then do the action a and terminate. The process $(\boldsymbol{\tau} \cdot (\Delta 1 + \varepsilon) + \varepsilon) \parallel \Delta 1 \cdot \boldsymbol{a}$, however, can tick, perform the a, then the action τ and then tick again.

   In both cases the problem lies in the fact that the process $\Delta 1 + \varepsilon$ can initially tick and terminate *at the same time*, while the process $\boldsymbol{\tau} \cdot (\Delta 1 + \varepsilon) + \varepsilon$ can terminate but not tick. In the sequential composition with $\Delta 1 \cdot \boldsymbol{a}$, the left-hand side process is subject to Rule $\langle \mathsf{seq\text{-}tick}_4 \rangle$ and the right-hand side process is not. Similarly, in the parallel composition with $\Delta 1 \cdot \boldsymbol{a}$, the left-hand side process is subject to Rule $\langle \mathsf{par\text{-}tick}_1 \rangle$ while the right-hand side process is not.

c. The processes $\Delta 1 + \varepsilon$ and $\Delta 1 \cdot \boldsymbol{\tau} + \varepsilon$ are timed stateless silently bisimilar. In the sequential composition with $\Delta 1 \cdot \boldsymbol{a}$ the left-hand side process does a tick and then chooses between another tick or the action a while the right-hand side process never has this choice.

   Both processes can tick and terminate at the same time and therefore are subject to Rule $\langle \mathsf{seq\text{-}tick}_4 \rangle$. This rule transforms a sequential composition into an alternative composition and, because alternative composition requests some kind of root condition, the problem emerges.□

The solution to the first two problems is to keep termination and the passage of time together. In other words, instead of simulating ticks and termination separately, we should combine them and distinguish the following three situations: a process terminates and can perform a tick, a process terminates but cannot tick, and a process does not terminate but can do a tick. By requiring in a bisimulation that these three predicates are simulated we solve the problem in the first two cases of the above example. To solve the problem in the third case, we should additionally require that when a tick with termination is simulated, then the resulting pair must satisfy some kind of root condition, i.e. the elements of the pair must simulate each other in the strong sense. However, it does not suffice to add to the root condition of Definition 4.3.3 that $\langle p,\sigma\rangle \overset{\Delta}{\mapsto} \langle p',\sigma'\rangle$, for some $\sigma \in \Sigma$, implies $\langle q,\sigma\rangle \overset{\Delta}{\mapsto} \langle q',\sigma'\rangle$ for some $q' \in P$ (and the symmetric case). This is illustrated in the following example.

Example 5.1.2 As before, we assume that $\mathsf{eff}(\boldsymbol{\tau},\sigma) = \{\sigma\}$ for all $\sigma \in \Sigma$. The processes $\Delta 2$ and $\Delta 1 \cdot \boldsymbol{\tau} \cdot \Delta 1$ would then be rooted timed stateless silent bisimilar. However, $\Delta 2 + \Delta 1 \cdot \boldsymbol{a}$ and $\Delta 1 \cdot \boldsymbol{\tau} \cdot \Delta 1 + \Delta 1 \cdot \boldsymbol{a}$ are not timed stateless silently bisimilar. This is because after a tick the left-hand side process can choose between another tick and the execution of a while the right-hand side process never has this choice. □

This problem is well known from the setting of timed branching bisimulation. The solution is to require that two processes simulate each other's steps in the strong sense until an action is executed (see [97, 7, 6]). For our setting, however, this is still not a suitable solution. We need to use the root condition not only at the root, but also for the pairs in the bisimulation relation that result from matching transitions. The root condition must be used instead of the regular bisimulation conditions and it must be stronger than them. We decided to have a nested relation inside a bisimulation relation that contains the "strong pairs".

Definition 5.1.3 (Relation of Strong Pairs) Let $R \subseteq P \times P$ be a symmetric relation. A subrelation $S \subseteq R$ is said to be a *relation of strong pairs* in R if it is symmetric and if, for all $(p,q) \in S$, the following holds:

$\langle\mathsf{str\text{-}term}^{\Delta}\rangle$ if $\langle p,\sigma\rangle\downarrow$, then $\langle q,\sigma\rangle\downarrow$,

$\langle\mathsf{str\text{-}tran}^{\Delta}\rangle$ if $\langle p,\sigma\rangle \xrightarrow{a} \langle p',\sigma'\rangle$, then there exists $q' \in P$ such that $\langle q,\sigma\rangle \xrightarrow{a} \langle q',\sigma'\rangle$ and $(p',q') \in R$, and

$\langle\mathsf{str\text{-}tick}^{\Delta}\rangle$ if $\langle p,\sigma\rangle \overset{\Delta}{\mapsto} \langle p',\sigma'\rangle$, then there exists $q' \in P$ such that $\langle q,\sigma\rangle \overset{\Delta}{\mapsto} \langle q',\sigma'\rangle$ and $(p',q') \in S$. □

Now, using the notion of strong pairs, and treating termination and ticks together, we define timed silent bisimulation.

Definition 5.1.4 (Timed Stateless Silent Bisimulation) A pair (S,R), where $R \subseteq P \times P$ is a symmetric relation and $S \subseteq R$ is a relation of strong pairs in R, is a *timed stateless silent bisimulation* if it satisfies $\langle\mathsf{sl\text{-}tran}\rangle$ and $\langle\mathsf{sl\text{-}div}\rangle$ and iff, for all $(p,q) \in R$, the following holds:

$\langle\mathsf{sl\text{-}term}^{\Delta}\rangle$ if $\langle p,\sigma\rangle\downarrow$, then there exists a $q' \in P$ such that $\langle q,\sigma\rangle \twoheadrightarrow \langle q',\sigma\rangle$, $\langle q',\sigma\rangle\downarrow$, $(p,q') \in R$ and such that $\langle p,\sigma\rangle \overset{\Delta}{\not\mapsto}$ implies $\langle q',\sigma\rangle \overset{\Delta}{\not\mapsto}$.

$\langle \textsf{sl-tick}^{\Delta} \rangle$ if $\langle p, \sigma \rangle \overset{\Delta}{\mapsto} \langle p', \sigma' \rangle$, then there exist $q', q'' \in P$ such that $\langle q, \sigma \rangle \twoheadrightarrow \langle q'', \sigma \rangle \overset{\Delta}{\mapsto} \langle q', \sigma' \rangle$, $(p, q'') \in R$, $(p', q') \in R$, and

(a) if $\langle p, \sigma \rangle \downarrow$, then $\langle q'', \sigma \rangle \downarrow$ and $(p', q') \in S$,

(b) if $\langle p, \sigma \rangle \not\downarrow$, then $\langle q'', \sigma \rangle \not\downarrow$.

Two processes p and q are *timed silently congruent*, denoted $p \approx_s^{\Delta} q$, if there exists a timed stateless silent bisimulation relation (S, R) such that $(p, q) \in S$. □

Note that, we could have alternatively defined timed stateless silent bisimulation relation (S, R) by requiring that R satisfies $\langle \textsf{sl-div}'' \rangle$ instead of $\langle \textsf{sl-div} \rangle$. This follows from Lemma 4.3.6.

The notion of a timed stateless silent bisimulation is a conservative extension of both, stateless silent bisimulation from Chapter 4, and timed branching bisimulation from [97, 7, 6]. If the timing part and the relation S are ignored, then timed stateless silent bisimulation coincides with stateless silent bisimulation. If the termination, the divergence, and the valuation part is ignored, the bisimulation coincides with timed branching bisimulation.

We now prove that $\approx_s^{\Delta}$ is indeed a congruence. First we prove that it is an equivalence relation. For that we need some lemmas. Some of the lemmas are given without a proof. This is either because they are straightforward or to avoid repeating the technicalities from Chapter 2.

Lemma 5.1.5 Let $R_i \in P \times P$ and $S_i \subseteq R_i$ for $i \in I$. Let $R = \bigcup_{i \in I} R_i$ and $S = \bigcup_{i \in I} S_i$. Let $\textsf{con} \in \{\langle \textsf{str-term}^{\Delta} \rangle, \langle \textsf{str-tran}^{\Delta} \rangle, \langle \textsf{str-tick}^{\Delta} \rangle\}$. Then if all (S_i, R_i) for $i \in I$ satisfy $\textsf{con}$, also (S, R) satisfies $\textsf{con}$. □

Lemma 5.1.6 Let $R_i \in P \times P$ and $S_i \subseteq R_i$ for $i = 1, 2$. Let $R = R_1 \circ R_2$ and $S = S_1 \circ S_2$. Let $\textsf{con} \in \{\langle \textsf{str-term}^{\Delta} \rangle, \langle \textsf{str-tran}^{\Delta} \rangle, \langle \textsf{str-tick}^{\Delta} \rangle\}$. Then if (S_1, R_1) and (S_2, R_2) satisfy $\textsf{con}$, also (S, R) satisfies $\textsf{con}$. □

Corollary 5.1.7 If $S_i \subseteq R_i$ is a relation of strong pairs in R_i, for $i = 1, 2$, then $S_1 \circ S_2$ is a relation of strong pairs in $R_1 \circ R_2$. □

Lemma 5.1.8 Let $R_i \in P \times P$ and let S_i be a relation of strong pairs in R_i, for $i = 1, 2$. Let $R = R_1 \circ R_2$ and $S = S_1 \circ S_2$. Suppose that R_2 satisfies $\langle \textsf{sl-tran} \rangle$. Then if (S_1, R_1) and (S_2, R_2) satisfy $\langle \textsf{sl-term}^{\Delta} \rangle$, resp. $\langle \textsf{sl-tick}^{\Delta} \rangle$, then (S, R) also satisfies $\langle \textsf{sl-term}^{\Delta} \rangle$, resp. $\langle \textsf{sl-tick}^{\Delta} \rangle$. □

Proof Let $(p,r) \in R$. Then there exists a $q \in P$ such that $(p,q) \in R_1$ and $(q,r) \in R_2$.

Suppose first that $\langle p,\sigma\rangle{\downarrow}$ for some $\sigma \in \Sigma$. Then there exists a $q' \in P$ such that $\langle q,\sigma\rangle \twoheadrightarrow \langle q',\sigma\rangle$, $\langle q',\sigma\rangle{\downarrow}$ and $(p,q') \in R_1$. Also, if $\langle p,\sigma\rangle \not\stackrel{\Delta}{\mapsto}$, then $\langle q',\sigma\rangle \not\stackrel{\Delta}{\mapsto}$. Since R_2 satisfies $\langle \mathsf{sl\text{-}tran}\rangle$, from Lemma 4.3.5 it follows that there exists an $r'' \in P$ such that $\langle r,\sigma\rangle \twoheadrightarrow \langle r'',\sigma\rangle$ and $(q',r'') \in R_2$. Since $\langle q',\sigma\rangle{\downarrow}$, there exists an $r' \in P$ such that $\langle r'',\sigma\rangle \twoheadrightarrow \langle r',\sigma\rangle$, $\langle r',\sigma\rangle{\downarrow}$ and $(q',r') \in R_2$, and that $\langle q',\sigma\rangle \not\stackrel{\Delta}{\mapsto}$ implies $\langle r',\sigma\rangle \not\stackrel{\Delta}{\mapsto}$. From $(p,q') \in R_1$ and $(q',r') \in R_2$ we have $(p,r') \in R$. Clearly, if $\langle p,\sigma\rangle \not\stackrel{\Delta}{\mapsto}$, then $\langle r',\sigma\rangle \not\stackrel{\Delta}{\mapsto}$.

Suppose $\langle p,\sigma\rangle \stackrel{\Delta}{\mapsto} \langle p',\sigma'\rangle$ for some $\sigma,\sigma' \in \Sigma$. Then there exist $q',q'' \in P$ such that $\langle q,\sigma\rangle \twoheadrightarrow \langle q'',\sigma\rangle \stackrel{\Delta}{\mapsto} \langle q',\sigma'\rangle$, $(p,q'') \in R_1$, $(p',q') \in R_1$, and if if $\langle p,\sigma\rangle{\downarrow}$, then $\langle q'',\sigma\rangle{\downarrow}$ and $(p',q') \in S_1$, and if $\langle p,\sigma\rangle \not\downarrow$, then $\langle q'',\sigma\rangle \not\downarrow$. By Lemma 4.3.5 there is an $r'' \in P$, such that $\langle r,\sigma\rangle \twoheadrightarrow \langle r'',\sigma\rangle$ and $(q'',r'') \in R_2$. From the latter it follows that there exist $r',r''' \in P$ such that $\langle r'',\sigma\rangle \twoheadrightarrow \langle r''',\sigma\rangle \stackrel{\Delta}{\mapsto} \langle r',\sigma'\rangle$, $(q'',r''') \in R_2$, $(q',r') \in R_2$, and if $\langle q'',\sigma\rangle{\downarrow}$, then $\langle r'',\sigma\rangle{\downarrow}$ and $(q',r') \in S_2$, and if $\langle q'',\sigma\rangle \not\downarrow$, then $\langle r'',\sigma\rangle \not\downarrow$.

From $(p,q'') \in R_1$ and $(q'',r'') \in R_2$ we obtain $(p,r'') \in R$, and from $(p',q') \in R_1$ and $(q',r') \in R_2$, we obtain $(p',r') \in R$. If $\langle p,\sigma\rangle{\downarrow}$, then $\langle q'',\sigma\rangle{\downarrow}$, and from this $\langle r'',\sigma\rangle{\downarrow}$. In this case, we also have $(p',q') \in S_1$ and $(q',r') \in S_2$, so $(p',r') \in S$. If $\langle p,\sigma\rangle \not\downarrow$, then $\langle q'',\sigma\rangle \not\downarrow$, and from this $\langle r'',\sigma\rangle \not\downarrow$. ■

The following theorem now easily follows.

Theorem 5.1.9 The relation $\approx_s^\Delta$ is an equivalence relation. □

We now show that timed silent congruence is compositional with respect to the operators of Timed κ.

Theorem 5.1.10 For all $p,q,\bar{p},\bar{q} \in P$, if $p \approx_s^\Delta q$ and $\bar{p} \approx_s^\Delta \bar{q}$, then

1. $\mathsf{b} :\to p \approx_s^\Delta \mathsf{b} :\to q$ for all $\mathsf{b} \in \mathsf{B}$,
2. $p \cdot \bar{p} \approx_s^\Delta q \cdot \bar{q}$,
3. $p + \bar{p} \approx_s^\Delta q + \bar{q}$,
4. $p^* \approx_s^\Delta q^*$,
5. $p \parallel \bar{p} \approx_s^\Delta q \parallel \bar{q}$,
6. $[\![\varsigma \mid p]\!] \approx_s^\Delta [\![\varsigma \mid q]\!]$ for all valuations $\varsigma \in \Sigma$,

7. $\partial_\Xi(p) \approx_s^\Delta \partial_\Xi(q)$ for all $\Xi \subseteq \mathsf{A} \setminus \{\tau\}$.
8. $\tau_I(p) \approx_s^\Delta \tau_I(q)$ for all $I \subseteq \mathsf{A}$. □

Proof All cases are proven in the same fashion. We let (S_{pq}, R_{pq}) and $(S_{\bar{p}\bar{q}}, R_{\bar{p}\bar{q}})$ be two timed stateless silent bisimulation that witness the $p \approx_s^\Delta q$ and $\bar{p} \approx_s^\Delta \bar{q}$ respectively. Using these relations we construct a pair (S, R), show that it is a timed stateless silent bisimulation, and that the desired pair is in S. When checking the conditions for a timed stateless silent bisimulation we ignore symmetric cases. The pairs from R that were already treated in the corresponding cases of Theorem 4.3.7 are not checked against ⟨sl-tran⟩ and ⟨sl-div⟩ (or ⟨sl-div"⟩). The pairs in R that are also in S are not checked against Conditions ⟨sl-term$^\Delta$⟩, ⟨sl-tran⟩, ⟨sl-div"⟩ and ⟨sl-tick$^\Delta$⟩. These is because these conditions are directly implied by the conditions of Definition 5.1.3.

1. Define $R = R_{pq} \cup S$ with

$$S = S_{pq} \cup \{(\mathsf{b} :\rightarrow p, \mathsf{b} :\rightarrow q), (\mathsf{b} :\rightarrow q, \mathsf{b} :\rightarrow p)\}.$$

It is enough to show that S is a relation of strong pairs in R. We only check the conditions of Definition 5.1.3 for the pair $(\mathsf{b} :\rightarrow p, \mathsf{b} :\rightarrow q)$.

Cond. ⟨str-term$^\Delta$⟩**:** Suppose $\langle \mathsf{b} :\rightarrow p, \sigma\rangle\downarrow$ for a $\sigma \in \Sigma$. Rule ⟨grd-term⟩ is the final rule of any derivation with $\langle \mathsf{b} :\rightarrow p, \sigma\rangle\downarrow$ as conclusion, so $\mathsf{check}(\mathsf{b}, \sigma) = \mathit{true}$ and $\langle p, \sigma\rangle\downarrow$. Since $(p, q) \in S_{pq}$, we have $\langle q, \sigma\rangle\downarrow$. By Rule ⟨grd-term⟩, $\langle \mathsf{b} :\rightarrow q, \sigma\rangle\downarrow$.

Cond. ⟨str-tran$^\Delta$⟩**:** Suppose $\langle \mathsf{b} :\rightarrow p, \sigma\rangle \xrightarrow{a} \langle r, \sigma'\rangle$ for some $\sigma, \sigma' \in \Sigma$. Since Rule ⟨grd-tran⟩ must be the final rule in any derivation of this transition, it holds that $\mathsf{check}(\mathsf{b}, \sigma) = \mathit{true}$, $\langle p, \sigma\rangle \xrightarrow{a} \langle p', \sigma'\rangle$ and $r = p'$. Since $(p, q) \in S_{pq}$, there exists a $q' \in P$ such that $\langle q, \sigma\rangle \xrightarrow{a} \langle q', \sigma'\rangle$ and $(p', q') \in R_{pq}$. Hence, by Rule ⟨grd-tran⟩, $\langle \mathsf{b} :\rightarrow q, \sigma\rangle \xrightarrow{a} \langle q', \sigma'\rangle$. From $(p', q') \in R_{pq}$, according to the definition of R, $(p', q') \in R$.

Cond. ⟨str-tick$^\Delta$⟩**:** Suppose $\langle \mathsf{b} :\rightarrow p, \sigma\rangle \overset{\Delta}{\mapsto} \langle p', \sigma'\rangle$ for some $p' \in P$ and $\sigma, \sigma' \in \Sigma$. Then by Rule ⟨grd-tick⟩ $\mathsf{check}(\mathsf{b}, \sigma) = \mathit{true}$ and $\langle p, \sigma\rangle \overset{\Delta}{\mapsto} \langle p', \sigma'\rangle$. Since $(p, q) \in S_{pq}$, there exists a $q' \in P$ such that $\langle q, \sigma\rangle \overset{\Delta}{\mapsto} \langle q', \sigma'\rangle$ and $(p', q') \in S_{pq}$. By Rule ⟨grd-tick⟩ again, $\langle \mathsf{b} :\rightarrow q, \sigma\rangle \overset{\Delta}{\mapsto} \langle q', \sigma'\rangle$. From $(p', q') \in S_{pq}$ and the definition of S it follows that $(p', q') \in S$.

2. Let

$$R = S \cup \{(r \cdot \bar{p}, s \cdot \bar{q}) \mid (r,s) \in R_{pq}\} \cup \{(r \cdot \bar{q}, s \cdot \bar{p}) \mid (r,s) \in R_{pq}\} \cup R_{\bar{p}\bar{q}},$$

where $S = \bigcup_{i=1}^{\infty} S_i$ with S_i, $i \geqslant 1$, defined by

$$S_1 = \{(r \cdot \bar{p}, s \cdot \bar{q}) \mid (r,s) \in S_{pq}\} \cup \{(r \cdot \bar{q}, s \cdot \bar{p}) \mid (r,s) \in S_{pq}\} \cup S_{\bar{p}\bar{q}},$$

and

$$S_{n+1} = \{(r + \bar{r}, s + \bar{s}) \mid (r,s) \in S_n \text{ and } (\bar{r}, \bar{s}) \in S_{\bar{p}\bar{q}}\}.$$

First we show that S is a relation of strong pairs in R. Suppose $(x,y) \in S$. Then $(x,y) \in S_n$, for some $n \geqslant 1$.

Cond. $\langle \textsf{str-term}^{\Delta} \rangle$**:** Suppose $\langle x, \sigma \rangle \downarrow$. We prove, by induction on n, that $\langle y, \sigma \rangle \downarrow$. Suppose first that $n = 1$. Then $x = r \cdot \bar{p}$ and $y = s \cdot \bar{q}$ with $(r,s) \in S_{pq}$, or $(x,y) \in S_{\bar{p}\bar{q}}$. In the first case, by Rule $\langle \textsf{seq-term} \rangle$ we have $\langle r, \sigma \rangle \downarrow$ and $\langle \bar{p}, \sigma \rangle \downarrow$. Since $(r,s) \in S_{pq}$ and $(\bar{p}, \bar{q}) \in S_{\bar{p}\bar{q}}$, we have $\langle s, \sigma \rangle \downarrow$ and $\langle \bar{q}, \sigma \rangle \downarrow$. By Rule $\langle \textsf{seq-term} \rangle$, $\langle s \cdot \bar{q}, \sigma \rangle \downarrow$. In the second case, from $(x,y) \in S_{\bar{p}\bar{q}}$ we obtain $\langle y, \sigma \rangle \downarrow$.
Suppose the statement holds for $k \leqslant n$ and suppose $(x,y) \in S_{n+1}$. Then $x = r + \bar{r}$ and $y = s + \bar{s}$ where $(r,s) \in S_n$ and $(\bar{r}, \bar{s}) \in S_{\bar{p}\bar{q}}$. Rule $\langle \textsf{alt-term} \rangle$ is the final rule in any derivation with $\langle r + \bar{r}, \sigma \rangle \downarrow$ as conclusion, so either $\langle r, \sigma \rangle \downarrow$ or $\langle \bar{r}, \sigma \rangle \downarrow$. If $\langle r, \sigma \rangle \downarrow$, then since $(r,s) \in S_n$, by the inductive hypothesis, $\langle s, \sigma \rangle \downarrow$. If $\langle \bar{r}, \sigma \rangle \downarrow$, then because $(\bar{r}, \bar{s}) \in S_{\bar{p}\bar{q}}$, we have $\langle \bar{s}, \sigma \rangle \downarrow$. In both cases, we obtain $\langle s + \bar{s}, \sigma \rangle \downarrow$ by Rule $\langle \textsf{alt-term} \rangle$.

Cond. $\langle \textsf{str-tran}^{\Delta} \rangle$**:** Suppose now $\langle x, \sigma \rangle \xrightarrow{a} \langle x', \sigma' \rangle$. We prove, by induction on n, that $\langle y, \sigma \rangle \xrightarrow{a} \langle y', \sigma' \rangle$ for some $y' \in P$ such that $(x', y') \in R$. Suppose $n = 1$. Then $x = r \cdot \bar{p}$ and $y = s \cdot \bar{q}$ with $(r,s) \in S_{pq}$, or $x = \bar{r}$ and $y = \bar{s}$ with $(\bar{r}, \bar{s}) \in S_{\bar{p}\bar{q}}$.
Suppose first $x = r \cdot \bar{p}$ and $y = s \cdot \bar{q}$ for $(r,s) \in S_{pq}$. The final rule in any derivation with $\langle r \cdot \bar{p}, \sigma \rangle \xrightarrow{a} \langle x', \sigma' \rangle$ as conclusion is either Rule $\langle \textsf{seq-tran}_1 \rangle$ or Rule $\langle \textsf{seq-tran}_2 \rangle$. If Rule $\langle \textsf{seq-tran}_1 \rangle$ is the final rule applied, we get $\langle r, \sigma \rangle \downarrow$, $\langle \bar{p}, \sigma \rangle \xrightarrow{a} \langle \bar{p}', \sigma' \rangle$ and $x' = \bar{p}'$ for some $\bar{p}' \in P$. Since $(r,s) \in S_{pq}$, we have $\langle s, \sigma \rangle \downarrow$. Since $(\bar{r}, \bar{s}) \in S_{\bar{p}\bar{q}}$, we have $\langle \bar{q}, \sigma \rangle \xrightarrow{a} \langle \bar{q}', \sigma' \rangle$ and $(\bar{p}', \bar{q}') \in R_{\bar{p}\bar{q}}$ for some $\bar{q}' \in P$. By Rule $\langle \textsf{seq-tran}_1 \rangle$, $\langle s \cdot \bar{q}, \sigma \rangle \xrightarrow{a} \langle \bar{q}', \sigma' \rangle$. From $(\bar{p}', \bar{q}') \in R_{\bar{p}\bar{q}}$, according to the definition of R, we have $(\bar{p}', \bar{q}') \in R$. If Rule $\langle \textsf{seq-tran}_2 \rangle$ is the final rule applied, then $\langle r, \sigma \rangle \xrightarrow{a} \langle r', \sigma' \rangle$ and $x' = r' \cdot \bar{p}$.

Since $(r,s) \in R_{pq}$, $\langle s,\sigma\rangle \xrightarrow{a} \langle s',\sigma'\rangle$ and $(r',s') \in R_{pq}$ for some $s' \in P$. By Rule $\langle \mathsf{seq\text{-}tran}_2\rangle$, we obtain $\langle s\cdot\bar{q},\sigma\rangle \xrightarrow{a} \langle s'\cdot\bar{q},\sigma'\rangle$. That $(r'\cdot\bar{p}, s'\cdot\bar{q}) \in R$ follows from $(r',s') \in R_{pq}$ and the definition of R.

Suppose now that $x = \bar{r}$ and $y = \bar{s}$ with $(\bar{r},\bar{s}) \in S_{\bar{p}\bar{q}}$. From $\langle \bar{r},\sigma\rangle \xrightarrow{a} \langle \bar{r}',\sigma'\rangle$ we obtain $\langle \bar{s},\sigma\rangle \xrightarrow{a} \langle \bar{s}',\sigma'\rangle$ and $(\bar{r}',\bar{s}') \in R_{\bar{p}\bar{q}}$ for some $\bar{s}' \in P$. According to the definition of R, $(\bar{r}',\bar{s}') \in R$.

Suppose the statement holds for $k \leqslant n$ and suppose $(x,y) \in S_{n+1}$. Then $x = r+\bar{r}$ and $y = s+\bar{s}$ with $(r,s) \in S_n$ and $(\bar{r},\bar{s}) \in S_{\bar{p}\bar{q}}$. Since Rule $\langle \mathsf{alt\text{-}tran}\rangle$ must be the final rule in any derivation with $\langle r+\bar{r},\sigma\rangle \xrightarrow{a} \langle x',\sigma'\rangle$ as conclusion, we obtain that either $\langle r,\sigma\rangle \xrightarrow{a} \langle r',\sigma'\rangle$ and $x' = r'$ for some $r' \in P$, or that $\langle \bar{r},\sigma\rangle \xrightarrow{a} \langle \bar{r}',\sigma'\rangle$ and $x' = \bar{r}'$ for some $\bar{s}' \in P$. In the first case, by the inductive hypothesis, there exists an $s' \in P$ such that $\langle s,\sigma\rangle \xrightarrow{a} \langle s',\sigma'\rangle$ and $(r',s') \in R$. By Rule $\langle \mathsf{alt\text{-}tran}\rangle$, $\langle s+\bar{s},\sigma\rangle \xrightarrow{a} \langle s',\sigma'\rangle$. In the second case, $(\bar{r},\bar{s}) \in S_{\bar{p}\bar{q}}$, there exists an $\bar{s}' \in P$ such that $\langle \bar{s},\sigma\rangle \xrightarrow{a} \langle \bar{s}',\sigma'\rangle$ and $(\bar{r}',\bar{s}') \in R_{\bar{p}\bar{q}}$. By Rule $\langle \mathsf{alt\text{-}tran}\rangle$, $\langle s+\bar{s},\sigma\rangle \xrightarrow{a} \langle \bar{s}',\sigma'\rangle$. Since $(\bar{r}',\bar{s}') \in R_{\bar{p}\bar{q}}$, according to the definition of R it follows that $(\bar{r}',\bar{s}') \in R$.

Cond. $\langle \mathsf{str\text{-}tick}^{\Delta}\rangle$**:** Suppose now $\langle x,\sigma\rangle \overset{\Delta}{\mapsto} \langle x',\sigma'\rangle$. We prove, by induction on n, that $\langle y,\sigma\rangle \overset{\Delta}{\mapsto} \langle y',\sigma'\rangle$ for some $y' \in P$ such that $(x',y') \in S$. Suppose $n = 1$. Then $x = r\cdot\bar{p}$ and $y = s\cdot\bar{q}$ with $(r,s) \in S_{pq}$, or $x = \bar{r}$ and $y = \bar{s}$ with $(\bar{r},\bar{s}) \in S_{\bar{p}\bar{q}}$.

Suppose first that $x = \bar{r}$ and $y = \bar{s}$ where $(\bar{r},\bar{s}) \in S_{\bar{p}\bar{q}}$. Then there exists an $\bar{s}' \in P$ such that $\langle \bar{s},\sigma\rangle \overset{\Delta}{\mapsto} \langle \bar{s}',\sigma'\rangle$ and $(\bar{r}',\bar{s}') \in R_{\bar{p}\bar{q}}$ and $(\bar{r}',\bar{s}') \in R_{\bar{p}\bar{q}}$. According to the definition of S_1, $(\bar{r}',\bar{s}') \in S_1 \subseteq S$.

Suppose now that $x = r\cdot\bar{p}$ and $y = s\cdot\bar{q}$ with $(r,s) \in S_{pq}$. Any of the rules $\langle \mathsf{seq\text{-}tick}_1\rangle$, $\langle \mathsf{seq\text{-}tick}_2\rangle$, $\langle \mathsf{seq\text{-}tick}_3\rangle$ or $\langle \mathsf{seq\text{-}tick}_4\rangle$ can be the final rule in a derivation with $\langle r\cdot\bar{p},\sigma\rangle \overset{\Delta}{\mapsto} \langle x',\sigma'\rangle$ as conclusion.

If Rule $\langle \mathsf{seq\text{-}tick}_1\rangle$ is the final rule applied, then $\langle r,\sigma\rangle \overset{\Delta}{\mapsto} \langle r',\sigma'\rangle$, $\langle r,\sigma\rangle\not\downarrow$ and $x' = r'\cdot\bar{p}$ for some $r' \in P$. Since $(r,s) \in S_{pq}$, we have that $\langle s,\sigma\rangle\not\downarrow$ and that there exists an $s' \in P$ such that $\langle s,\sigma\rangle \overset{\Delta}{\mapsto} \langle s',\sigma'\rangle$ and $(r',s) \in S_{pq}$. By Rule $\langle \mathsf{seq\text{-}tick}_1\rangle$, $\langle s\cdot\bar{q},\sigma\rangle \overset{\Delta}{\mapsto} \langle s'\cdot\bar{q},\sigma'\rangle$. Since $(r',s') \in S_{pq}$, according to the definition of S, $(r'\cdot\bar{p}, s'\cdot\bar{q}) \in S$.

If Rule $\langle \mathsf{seq\text{-}tick}_2\rangle$ is the final rule applied, then $\langle r,\sigma\rangle \overset{\Delta}{\mapsto} \langle r',\sigma'\rangle$,

$\langle \bar{p}, \sigma\rangle \not\overset{\Delta}{\mapsto}$ and $x' = r' \cdot \bar{p}$ for some $r' \in P$. Since $(r, s) \in S_{pq}$, there exists an $s' \in P$ such that $\langle s, \sigma\rangle \overset{\Delta}{\mapsto} \langle s', \sigma'\rangle$ and $(r', s') \in S_{pq}$. Since $(\bar{r}, \bar{s}) \in S_{\bar{p}\bar{q}}$, we have $\langle \bar{q}, \sigma\rangle \not\overset{\Delta}{\mapsto}$. Now, by Rule $\langle \mathsf{seq\text{-}tick}_2\rangle$, $\langle s \cdot \bar{q}, \sigma\rangle \overset{\Delta}{\mapsto} \langle s' \cdot \bar{q}, \sigma'\rangle$. From $(r', s') \in S_{pq}$, by the definition of S, $(r' \cdot \bar{p}, s' \cdot \bar{q}) \in S$.

If Rule $\langle \mathsf{seq\text{-}tick}_3\rangle$ is the final rule applied, then $\langle r, \sigma\rangle\downarrow$, $\langle r, \sigma\rangle \not\overset{\Delta}{\mapsto}$, $\langle \bar{p}, \sigma\rangle \overset{\Delta}{\mapsto} \langle \bar{p}', \sigma'\rangle$ and $x' = \bar{p}'$ for some $\bar{p}' \in P$. Since $(r, s) \in S_{pq}$, we have $\langle s, \sigma\rangle\downarrow$ and $\langle s, \sigma\rangle \not\overset{\Delta}{\mapsto}$. Since $(\bar{p}, \bar{q}) \in S_{\bar{p}\bar{q}}$, there exists a $\bar{q}' \in P$ such that $\langle \bar{q}, \sigma\rangle \overset{\Delta}{\mapsto} \langle \bar{q}', \sigma'\rangle$ and $(\bar{p}', \bar{q}') \in S_{\bar{p}\bar{q}}$. By Rule $\langle \mathsf{seq\text{-}tick}_3\rangle$, $\langle s \cdot \bar{q}, \sigma\rangle \overset{\Delta}{\mapsto} \langle \bar{q}', \sigma'\rangle$ and from the definition of S_1, $(\bar{p}', \bar{q}') \in S_1 \subseteq S$.

If Rule $\langle \mathsf{seq\text{-}tick}_4\rangle$ is the final rule applied, then $\langle r, \sigma\rangle\downarrow$, $\langle r, \sigma\rangle \overset{\Delta}{\mapsto} \langle r', \sigma'\rangle$, $\langle \bar{p}, \sigma\rangle \overset{\Delta}{\mapsto} \langle \bar{p}', \sigma'\rangle$ and $x' = r \cdot \bar{p} + \bar{p}'$ for some $r', \bar{p}' \in P$. Since $(r, s) \in S_{pq}$, we have that $\langle s, \sigma\rangle\downarrow$ and that there exists an $s' \in P$ such that $\langle s, \sigma\rangle \overset{\Delta}{\mapsto} \langle s', \sigma'\rangle$, $(r', s') \in S_{pq}$. Since $(\bar{p}, \bar{q}) \in S_{\bar{p}\bar{q}}$, there exists a $\bar{q}' \in P$ such that $\langle \bar{q}, \sigma\rangle \overset{\Delta}{\mapsto} \langle \bar{q}', \sigma'\rangle$ and $(\bar{p}', \bar{q}') \in S_{\bar{p}\bar{q}}$. By Rule $\langle \mathsf{seq\text{-}tick}_4\rangle$, $\langle s \cdot \bar{q}, \sigma\rangle \overset{\Delta}{\mapsto} \langle s' \cdot \bar{q} + \bar{q}', \sigma'\rangle$. Since $(\bar{r}', \bar{s}') \in S_{\bar{p}\bar{q}}$, by the definition of S_1, $(r' \cdot \bar{p}, s' \cdot \bar{q}) \in S_1$. Since $(\bar{p}', \bar{q}') \in S_{\bar{p}\bar{q}}$, according to the definition of S_2, $(r' \cdot \bar{p} + \bar{p}', s' \cdot \bar{q} + \bar{q}') \in S_2 \subseteq S$.

Suppose the statement holds for $k \leqslant n$ and suppose $(x, y) \in S_{n+1}$. Then $x = r + \bar{r}$ and $y = s + \bar{s}$ where $(r, s) \in S_n$ and $(\bar{r}, \bar{s}) \in S_{\bar{p}\bar{q}}$. Either Rule $\langle \mathsf{alt\text{-}tick}_1\rangle$ or Rule $\langle \mathsf{alt\text{-}tick}_2\rangle$ is the final rule in any derivation with $\langle r + \bar{r}, \sigma\rangle \overset{\Delta}{\mapsto} \langle x', \sigma'\rangle$ as conclusion. We treat the two cases separately.

If Rule $\langle \mathsf{alt\text{-}tick}_1\rangle$ is the final rule applied, then either $\langle r, \sigma\rangle \overset{\Delta}{\mapsto} \langle r', \sigma'\rangle$, $\langle \bar{r}, \sigma\rangle \not\overset{\Delta}{\mapsto}$ and $x' = r'$ for some $r' \in P$, or $\langle \bar{r}, \sigma\rangle \overset{\Delta}{\mapsto} \langle \bar{r}', \sigma'\rangle$, $\langle r, \sigma\rangle \not\overset{\Delta}{\mapsto}$ and $x' = \bar{r}'$ for some $\bar{r}' \in P$. In the first case, by the inductive hypothesis, there exists an $s' \in P$ such that $\langle s, \sigma\rangle \overset{\Delta}{\mapsto} \langle s', \sigma'\rangle$ and $(r', s') \in S$. Since $(\bar{r}, \bar{s}) \in S_{\bar{p}\bar{q}}$, from the inductive hypothesis and a simple contradiction it follows that $\langle \bar{s}, \sigma\rangle \not\overset{\Delta}{\mapsto}$. By Rule $\langle \mathsf{alt\text{-}tick}_1\rangle$, $\langle s + \bar{s}, \sigma\rangle \overset{\Delta}{\mapsto} \langle s', \sigma'\rangle$. In the second case, by the inductive hypothesis, $\langle s, \sigma\rangle \not\overset{\Delta}{\mapsto}$ and there exists an $\bar{s}' \in P$ such that $\langle \bar{s}, \sigma\rangle \overset{\Delta}{\mapsto} \langle \bar{s}', \sigma'\rangle$ and $(\bar{r}', \bar{s}') \in S_{\bar{p}\bar{q}}$. By Rule $\langle \mathsf{alt\text{-}tick}_1\rangle$, $\langle s + \bar{s}, \sigma\rangle \overset{\Delta}{\mapsto} \langle \bar{s}', \sigma'\rangle$. From $(\bar{r}', \bar{s}') \in S_{\bar{p}\bar{q}}$, we have $(\bar{r}', \bar{s}') \in S_1 \subseteq S$.

If Rule $\langle \mathsf{alt\text{-}tick}_2\rangle$ is the final rule applied, then $\langle r, \sigma\rangle \overset{\Delta}{\mapsto} \langle r', \sigma'\rangle$,

$\langle \bar{r}, \sigma\rangle \overset{\Delta}{\mapsto} \langle \bar{r}', \sigma'\rangle$ and $x' = r' + \bar{r}'$ for some $r', \bar{r}' \in P$. By the inductive hypothesis, there exists an $s' \in P$ such that $\langle s, \sigma\rangle \overset{\Delta}{\mapsto} \langle s', \sigma'\rangle$ and $(r', s') \in S$. Since $(\bar{r}, \bar{s}) \in S_{\bar{p}\bar{q}}$, there exists an $\bar{s}' \in P$ such that $\langle \bar{s}, \sigma\rangle \overset{\Delta}{\mapsto} \langle \bar{s}', \sigma'\rangle$ and $(\bar{r}', \bar{s}') \in S_{\bar{p}\bar{q}}$. By Rule $\langle\mathsf{alt\text{-}tick}_1\rangle$, $\langle s + \bar{s}, \sigma\rangle \overset{\Delta}{\mapsto} \langle s' + \bar{s}', \sigma'\rangle$. From $(r', s') \in S$ we have $(r', s') \in S_m$ for some $m \geqslant 1$. Since also $(\bar{r}', \bar{s}') \in S_{\bar{p}\bar{q}}$, according to the definition of S, we have $(r' + \bar{r}', s' + \bar{s}') \in S_{m+1} \subseteq S$.

This completes the proof that S is a relation of strong pairs in R.

We now show that (S, R) satisfies the conditions of Definition 5.1.4. It is enough to show that the pairs in $R' = \{(r \cdot \bar{p}, s \cdot \bar{q}) \mid (r, s) \in R_{pq}\} \cup \{(r \cdot \bar{q}, s \cdot \bar{p}) \mid (r, s) \in R_{pq}\}$ satisfy $\langle\mathsf{sl\text{-}term}^{\Delta}\rangle$ and $\langle\mathsf{sl\text{-}tick}^{\Delta}\rangle$.

Cond. $\langle\mathsf{sl\text{-}term}^{\Delta}\rangle$**:** Suppose $\langle r \cdot \bar{p}, \sigma\rangle\!\downarrow$. By Rule $\langle\mathsf{seq\text{-}term}\rangle$, we have $\langle r, \sigma\rangle\!\downarrow$ and $\langle \bar{p}, \sigma\rangle\!\downarrow$. Since $(\bar{p}, \bar{q}) \in S_{\bar{p}\bar{q}}$, we have $\langle \bar{q}, \sigma\rangle\!\downarrow$. Since $(r, s) \in R_{pq}$, there is an $s' \in P$ such that $\langle s, \sigma\rangle \twoheadrightarrow \langle s', \sigma\rangle$, $\langle s', \sigma\rangle\!\downarrow$ and $(r, s') \in R_{pq}$. In addition, if $\langle r, \sigma\rangle \overset{\Delta}{\not\mapsto}$, then also $\langle s', \sigma\rangle \overset{\Delta}{\not\mapsto}$. By Rules $\langle\mathsf{seq\text{-}tran}_2\rangle$ and $\langle\mathsf{seq\text{-}term}\rangle$ we have $\langle s \cdot \bar{q}, \sigma\rangle \twoheadrightarrow \langle s' \cdot \bar{q}, \sigma\rangle$ and $\langle s' \cdot \bar{q}, \sigma\rangle\!\downarrow$. Suppose that $\langle r \cdot \bar{p}, \sigma\rangle \overset{\Delta}{\not\mapsto}$. Because $\langle r, \sigma\rangle\!\downarrow$, this implies $\langle r, \sigma\rangle \overset{\Delta}{\not\mapsto}$ and $\langle \bar{p}, \sigma\rangle \overset{\Delta}{\not\mapsto}$. From $\langle r, \sigma\rangle \overset{\Delta}{\not\mapsto}$ we obtain $\langle s', \sigma\rangle \overset{\Delta}{\not\mapsto}$. Since $(\bar{p}, \bar{q}) \in S_{\bar{p}\bar{q}}$, we have $\langle \bar{q}, \sigma\rangle \overset{\Delta}{\not\mapsto}$. From $\langle s', \sigma\rangle \overset{\Delta}{\not\mapsto}$ and $\langle \bar{p}, \sigma\rangle \overset{\Delta}{\not\mapsto}$ it follows that $\langle s' \cdot \bar{q}, \sigma\rangle \overset{\Delta}{\not\mapsto}$.

Cond. $\langle\mathsf{sl\text{-}tick}^{\Delta}\rangle$**:** Suppose $\langle r \cdot \bar{p}, \sigma\rangle \overset{\Delta}{\mapsto} \langle t, \sigma'\rangle$ for some $t \in P$. There are four rules with this transition as conclusion, namely $\langle\mathsf{seq\text{-}tick}_1\rangle$, $\langle\mathsf{seq\text{-}tick}_2\rangle$, $\langle\mathsf{seq\text{-}tick}_3\rangle$, $\langle\mathsf{seq\text{-}tick}_4\rangle$. We treat them separately.

If the final rule applied is Rule $\langle\mathsf{seq\text{-}tick}_1\rangle$, then $\langle r, \sigma\rangle\!\not\downarrow$, $\langle r, \sigma\rangle \overset{\Delta}{\mapsto} \langle r', \sigma'\rangle$ for some $r' \in P$, and $t = r' \cdot \bar{p}$. There exist $s', s'' \in P$ such that $\langle s, \sigma\rangle \twoheadrightarrow \langle s'', \sigma\rangle \overset{\Delta}{\mapsto} \langle s', \sigma'\rangle$, $(r, s'') \in R_{pq}$ and $(r', s') \in R_{pq}$. Since $\langle r, \sigma\rangle\!\not\downarrow$, we also have that $\langle s'', \sigma\rangle\!\not\downarrow$. By Rules $\langle\mathsf{seq\text{-}tran}_2\rangle$ and Rule $\langle\mathsf{seq\text{-}tick}_1\rangle$, $\langle s \cdot \bar{q}, \sigma\rangle \twoheadrightarrow \langle s'' \cdot \bar{q}, \sigma\rangle \overset{\Delta}{\mapsto} \langle s' \cdot \bar{q}, \sigma'\rangle$. Since $(r, s'') \in R_{pq}$ and $(r', s') \in R_{pq}$, by the definition of R, we have $(r \cdot \bar{p}, s'' \cdot \bar{q}) \in R$ and $(r' \cdot \bar{p}, s' \cdot \bar{q}) \in R$. Note that $\langle r \cdot \bar{p}, \sigma\rangle\!\not\downarrow$, because otherwise we would have by Rule $\langle\mathsf{seq\text{-}term}\rangle$ that $\langle r, \sigma\rangle\!\downarrow$. So, so we only need to show that $\langle s'' \cdot \bar{q}, \sigma\rangle\!\not\downarrow$. This however follows directly since $\langle s'', \sigma\rangle\!\not\downarrow$.

If the final rule applied is Rule $\langle\mathsf{seq\text{-}tick}_2\rangle$, then $\langle r, \sigma\rangle\!\downarrow$, $\langle r, \sigma\rangle \overset{\Delta}{\mapsto} \langle r', \sigma'\rangle$ for some $r' \in P$, and $\langle \bar{p}, \sigma\rangle \overset{\Delta}{\not\mapsto}$. Also $t = r' \cdot \bar{p}$. Then $\langle s, \sigma\rangle \twoheadrightarrow$

$\langle s'', \sigma\rangle \overset{\Delta}{\mapsto} \langle s', \sigma'\rangle$, $(r, s'') \in R_{pq}$ and $(r', s') \in R_{pq}$ for some $s', s'' \in P$. Since $\langle r, \sigma\rangle\downarrow$, we also have $\langle s'', \sigma\rangle\downarrow$ and $(r', s') \in S_{pq}$. Since $(\bar{p}, \bar{q}) \in S_{\bar{p}\bar{q}}$, we have $\langle \bar{q}, \sigma\rangle \overset{\Delta}{\not\mapsto}$. By Rules $\langle$seq-tran$_2\rangle$ and $\langle$seq-tick$_2\rangle$, we have $\langle s \cdot \bar{q}, \sigma\rangle \twoheadrightarrow \langle s'' \cdot \bar{q}, \sigma\rangle \overset{\Delta}{\mapsto} \langle s' \cdot \bar{q}, \sigma'\rangle$. From $(r, s'') \in R_{pq}$ and $(r', s') \in S_{pq}$, according to the definition of R, it follows that $(r \cdot \bar{p}, s'' \cdot \bar{q}) \in R$ and $(r' \cdot \bar{p}, s' \cdot \bar{q}) \in S_1 \subseteq S \subseteq R$.

Now, suppose first that $\langle r \cdot \bar{p}, \sigma\rangle\downarrow$. Then, by Rule $\langle$seq-term$\rangle$, $\langle \bar{p}, \sigma\rangle\downarrow$. Since $(\bar{p}, \bar{q}) \in S_{\bar{p}\bar{q}}$, we have $\langle \bar{q}, \sigma\rangle\downarrow$. Now from $\langle s'', \sigma\rangle\downarrow$ and $\langle \bar{q}, \sigma\rangle\downarrow$, by Rule $\langle$seq-term$\rangle$, we obtain $\langle s'' \cdot \bar{q}, \sigma\rangle\downarrow$. Suppose now that $\langle r \cdot \bar{p}, \sigma\rangle\not\downarrow$. Since $\langle r, \sigma\rangle\downarrow$, we have $\langle \bar{p}, \sigma\rangle\not\downarrow$. From $(\bar{p}, \bar{q}) \in S_{\bar{p}\bar{q}}$, it follows that $\langle \bar{q}, \sigma\rangle\not\downarrow$, and therefore that $\langle s'' \cdot \bar{q}, \sigma\rangle\not\downarrow$.

If the final rule applied is Rule $\langle$seq-tick$_3\rangle$, then $\langle r, \sigma\rangle\downarrow$, $\langle r, \sigma\rangle \overset{\Delta}{\not\mapsto}$ and $\langle \bar{p}, \sigma\rangle \overset{\Delta}{\mapsto} \langle \bar{p}', \sigma'\rangle$ for some $\bar{p}' \in P$. Then also $t = \bar{p}'$. Since $(\bar{p}, \bar{q}) \in S_{\bar{p}\bar{q}}$, there is a $\bar{q}' \in P$ such that $\langle \bar{q}, \sigma\rangle \overset{\Delta}{\mapsto} \langle \bar{q}', \sigma'\rangle$ and $(\bar{p}', \bar{q}') \in S_{\bar{p}\bar{q}}$. From $\langle r, \sigma\rangle\downarrow$ it follows that there exists an $s' \in P$ such that $\langle s, \sigma\rangle \twoheadrightarrow \langle s', \sigma\rangle$, $\langle s', \sigma\rangle\downarrow$ and $(r, s) \in R_{pq}$. Since $\langle r, \sigma\rangle \overset{\Delta}{\not\mapsto}$, it also follows that $\langle s', \sigma\rangle \overset{\Delta}{\not\mapsto}$. By Rules $\langle$seq-tran$_2\rangle$ and $\langle$seq-tick$_3\rangle$, $\langle s \cdot \bar{q}, \sigma\rangle \twoheadrightarrow \langle s' \cdot \bar{q}, \sigma\rangle \overset{\Delta}{\mapsto} \langle \bar{q}', \sigma'\rangle$. According to the definition of R, from $(r, s') \in R_{pq}$ it follows that $(r \cdot \bar{p}, s' \cdot \bar{q}) \in R$. From $(\bar{p}', \bar{q}') \in S_{\bar{p}\bar{q}}$ it follows that $(\bar{p}', \bar{q}') \in S \subseteq R$. Suppose first that $\langle r \cdot \bar{p}, \sigma\rangle\downarrow$. Then, by Rule $\langle$seq-term$\rangle$, $\langle \bar{p}, \sigma\rangle\downarrow$. Since $(\bar{p}, \bar{q}) \in S_{\bar{p}\bar{q}}$, we have $\langle \bar{q}, \sigma\rangle\downarrow$. By Rule $\langle$seq-term$\rangle$, from $\langle s', \sigma\rangle\downarrow$ and $\langle \bar{q}, \sigma\rangle\downarrow$, we obtain $\langle s' \cdot \bar{q}, \sigma\rangle\downarrow$. Suppose now that $\langle r \cdot \bar{p}, \sigma\rangle\not\downarrow$. Since $\langle r, \sigma\rangle\downarrow$, we have $\langle \bar{p}, \sigma\rangle\not\downarrow$. From $(\bar{p}, \bar{q}) \in S_{\bar{p}\bar{q}}$, it follows that $\langle \bar{q}, \sigma\rangle\not\downarrow$, and thus $\langle s' \cdot \bar{q}, \sigma\rangle\not\downarrow$.

If the final rule applied is Rule $\langle$seq-tick$_4\rangle$, then $\langle r, \sigma\rangle\downarrow$, $\langle r, \sigma\rangle \overset{\Delta}{\mapsto} \langle r', \sigma'\rangle$ for some $r' \in P$, and $\langle \bar{p}, \sigma\rangle \overset{\Delta}{\mapsto} \langle \bar{p}', \sigma'\rangle$ for some $\bar{p}' \in P$. In addition, $t = r' \cdot \bar{p} + \bar{p}'$. Since $(\bar{p}, \bar{q}) \in S_{\bar{p}\bar{q}}$, there is a $\bar{q}' \in P$ such that $\langle \bar{q}, \sigma\rangle \overset{\Delta}{\mapsto} \langle \bar{q}', \sigma'\rangle$ and $(\bar{p}', \bar{q}') \in S_{\bar{p}\bar{q}}$. Since $(r, s) \in R_{pq}$, there exist a $s', s'' \in P$ such that $\langle s, \sigma\rangle \twoheadrightarrow \langle s'', \sigma\rangle \overset{\Delta}{\mapsto} \langle s', \sigma'\rangle$, $(r, s'') \in R_{pq}$ and $(r', s') \in R_{pq}$. From $\langle r, \sigma\rangle\downarrow$ we have that $\langle s'', \sigma\rangle\downarrow$ and that $(r', s') \in S_{pq}$. By Rule $\langle$seq-tran$_2\rangle$ and Rule $\langle$seq-tick$_4\rangle$, $\langle s \cdot \bar{q}, \sigma\rangle \twoheadrightarrow \langle s'' \cdot \bar{q}, \sigma\rangle \overset{\Delta}{\mapsto} \langle s' \cdot \bar{q} + \bar{q}', \sigma'\rangle$. Since $(r, s'') \in R_{pq}$, according to the definition of R, $(r \cdot \bar{p}, s'' \cdot \bar{q}) \in R$. Since $(r', s') \in S_{pq}$, from the definition of S_1, $(r' \cdot \bar{p}, s' \cdot \bar{q}) \in S_1$. Because $(\bar{p}', \bar{q}') \in S_{\bar{p}\bar{q}}$, from the definition of S_2 it follows that

$(r' \cdot \bar{p} + \bar{p}', s' \cdot \bar{q} + \bar{q}') \in S_2 \subseteq S \subseteq R$. Suppose first that $\langle r \cdot \bar{p}, \sigma\rangle\downarrow$. Then, by Rule $\langle\mathsf{seq\text{-}term}\rangle$, $\langle\bar{p}, \sigma\rangle\downarrow$. Since $(\bar{p}, \bar{q}) \in S_{\bar{p}\bar{q}}$, we have $\langle\bar{q}, \sigma\rangle\downarrow$. From $\langle s', \sigma\rangle\downarrow$ and $\langle\bar{q}, \sigma\rangle\downarrow$, by Rule $\langle\mathsf{seq\text{-}term}\rangle$, we obtain $\langle s' \cdot \bar{q}, \sigma\rangle\downarrow$. Suppose now that $\langle r \cdot \bar{p}, \sigma\rangle\not\downarrow$. Since $\langle r, \sigma\rangle\downarrow$, we have $\langle\bar{p}, \sigma\rangle\not\downarrow$. From $(\bar{p}, \bar{q}) \in S_{\bar{p}\bar{q}}$, it follows that $\langle\bar{q}, \sigma\rangle\not\downarrow$, and thus $\langle s' \cdot \bar{q}, \sigma\rangle\not\downarrow$.

3. Let $R = S \cup R_{pq} \cup R_{\bar{p}\bar{q}}$ with

$$S = S_{pq} \cup S_{\bar{p}\bar{q}} \cup \{(r + \bar{r}, s + \bar{s}) \mid (r, s) \in S_{pq}, (\bar{r}, \bar{s}) \in S_{\bar{p}\bar{q}}\}\,.$$

We show that (S, R) is a timed stateless silent bisimulation. It is enough to show that S is a relation of strong pairs in R. We only need to check the conditions of Definition 5.1.3 for the pairs in the set $\{(r + \bar{r}, s + \bar{s}) \mid (r, s) \in S_{pq}, (\bar{r}, \bar{s}) \in S_{\bar{p}\bar{q}}\}$.

Cond. $\langle\mathsf{str\text{-}term}^{\Delta}\rangle$**:** Suppose $\langle r + \bar{r}, \sigma\rangle\downarrow$. Rule $\langle\mathsf{alt\text{-}term}\rangle$ must be the final rule in a derivation with this as conclusion so $\langle r, \sigma\rangle\downarrow$ or $\langle\bar{r}, \sigma\rangle\downarrow$. Since $(r, s) \in S_{pq}$, we have that either $\langle s, \sigma\rangle\downarrow$ or $\langle\bar{s}, \sigma\rangle\downarrow$. By Rule $\langle\mathsf{alt\text{-}term}\rangle$, $\langle s + \bar{s}, \sigma\rangle\downarrow$.

Cond. $\langle\mathsf{str\text{-}tran}^{\Delta}\rangle$**:** Suppose $\langle r + \bar{r}, \sigma\rangle \xrightarrow{a} \langle t, \sigma'\rangle$ for some $t \in P$. Since Rule $\langle\mathsf{alt\text{-}tran}\rangle$ must be the final rule of any derivation of this transition, it holds that $\langle r, \sigma\rangle \xrightarrow{a} \langle r', \sigma'\rangle$ and $t = r'$ or that $\langle\bar{r}, \sigma\rangle \xrightarrow{a} \langle\bar{r}', \sigma'\rangle$ and $t = \bar{r}'$. Suppose $\langle r, \sigma\rangle \xrightarrow{a} \langle r', \sigma'\rangle$ (the proof in the other case is similar). Since $(r, s) \in S_{pq}$, there exists $s' \in P$ such that $\langle s, \sigma\rangle \xrightarrow{a} \langle s', \sigma'\rangle$ and $(r', s') \in R_{pq}$. Now, by Rule $\langle\mathsf{alt\text{-}tran}\rangle$, $\langle s + \bar{s}, \sigma\rangle \xrightarrow{a} \langle s', \sigma'\rangle$. According to the definition of R, from $(r', s') \in R_{pq}$ we have $(r', s') \in R$.

Cond. $\langle\mathsf{str\text{-}tick}^{\Delta}\rangle$**:** Suppose $\langle r + \bar{r}, \sigma\rangle \overset{\Delta}{\mapsto} \langle t, \sigma'\rangle$ for some $t \in P$. The final rule in any derivation with this transition as conclusion is either Rule $\langle\mathsf{alt\text{-}tick_1}\rangle$ or Rule $\langle\mathsf{alt\text{-}tick_2}\rangle$. We treat these cases separately.

If Rule $\langle\mathsf{alt\text{-}tick_1}\rangle$ is the last rule applied, then $\langle r, \sigma\rangle \overset{\Delta}{\mapsto} \langle r', \sigma'\rangle$ for some $r' \in P$, $\langle\bar{r}, \sigma\rangle \overset{\Delta}{\not\mapsto}$ and $t = r'$ (or the symmetric case). Since $(r, s) \in S_{pq}$, there exists an $s' \in P$ such that $\langle s, \sigma\rangle \overset{\Delta}{\mapsto} \langle s', \sigma'\rangle$ and $(r', s') \in S_{pq}$. Since $(\bar{r}, \bar{s}) \in S_{\bar{p}\bar{q}}$, we have $\langle\bar{s}, \sigma\rangle \overset{\Delta}{\not\mapsto}$. By Rule $\langle\mathsf{alt\text{-}tick_1}\rangle$, $\langle s + \bar{s}, \sigma\rangle \overset{\Delta}{\mapsto} \langle s', \sigma'\rangle$. From $(r', s') \in S_{pq}$ and the definition of S, it follows that $(r', s') \in S$.

If Rule $\langle\textsf{alt-tick}_2\rangle$ is the last rule applied, then there exist $r', \bar{r}' \in P$ such that $\langle r, \sigma\rangle \overset{\Delta}{\mapsto} \langle r', \sigma'\rangle$, $\langle \bar{r}, \sigma\rangle \overset{\Delta}{\mapsto} \langle \bar{r}', \sigma'\rangle$ and $t = r' + \bar{r}'$ (or symmetrically $\langle s, \sigma\rangle \overset{\Delta}{\mapsto} \langle s', \sigma'\rangle$, $\langle \bar{s}, \sigma\rangle \overset{\Delta}{\mapsto} \langle \bar{s}', \sigma'\rangle$ and $t = s' + \bar{s}'$ for some $s', \bar{s}' \in P$. Since $(r, s) \in S_{pq}$, there exists an $s' \in P$ such that $\langle s, \sigma\rangle \overset{\Delta}{\mapsto} \langle s', \sigma'\rangle$ and $(r', s') \in S_{pq}$. Similarly, there exists an $\bar{s}' \in P$ such that $\langle \bar{s}, \sigma\rangle \overset{\Delta}{\mapsto} \langle \bar{s}', \sigma'\rangle$ and $(\bar{r}', \bar{s}') \in S_{\bar{p}\bar{q}}$. By Rule $\langle\textsf{alt-tick}_1\rangle$, $\langle s + \bar{s}, \sigma\rangle \overset{\Delta}{\mapsto} \langle s' + \bar{s}', \sigma'\rangle$. According to the definition of S, $(r' + \bar{r}', s' + \bar{s}') \in S$.

4. Let

$$R = S \cup \{(r \cdot p^*, s \cdot q^*) \mid (r, s) \in R_{pq}\} \cup \{(r \cdot q^*, s \cdot p^*) \mid (r, s) \in R_{pq}\},$$

where $S = \bigcup_{i=1}^{\infty} S_i$ with S_i, $i \geqslant 1$, defined by

$$\begin{array}{rcl} S & = & \{(p^*, q^*), (q^*, p^*)\} \\ & & \cup\ \{(r \cdot p^*, s \cdot q^*) \mid (r, s) \in S_{pq}\} \\ & & \cup\ \{(r \cdot q^*, s \cdot p^*) \mid (r, s) \in S_{pq}\}, \text{ and} \\ S_{n+1} & = & S_n \cup \{(r + \bar{r}, s + \bar{s}) \mid (r, s) \in S_n, (\bar{r}, \bar{s}) \in S_n\}. \end{array}$$

First we show that S is a relation of strong pairs in R. Suppose $(x, y) \in S$. Then $(x, y) \in S_n$, for some $n \geqslant 1$.

Cond. $\langle\textsf{str-term}^{\Delta}\rangle$**:** Suppose $\langle x, \sigma\rangle\!\downarrow$. We prove, by induction on n, that $\langle y, \sigma\rangle\!\downarrow$. Suppose first that $n = 1$. Since $\langle p^*, \sigma\rangle\!\downarrow$ for all $\sigma \in \Sigma$, we only need to observe the case when $x = r \cdot p^*$ and $y = s \cdot q^*$ with $(r, s) \in S_{pq}$. By Rule $\langle\textsf{seq-term}\rangle$, $\langle r, \sigma\rangle\!\downarrow$. Since $(r, s) \in S_{pq}$, we have $\langle s, \sigma\rangle\!\downarrow$. By Rule $\langle\textsf{seq-term}\rangle$, $\langle s \cdot q^*, \sigma\rangle\!\downarrow$.

Suppose the statement holds for $k \leqslant n$ and suppose $(x, y) \in S_{n+1}$. Then $x = r + \bar{r}$ and $y = s + \bar{s}$ where $(r, s) \in S_n$ and $(\bar{r}, \bar{s}) \in S_n$. Rule $\langle\textsf{alt-term}\rangle$ is the final rule in any derivation with $\langle r + \bar{r}, \sigma\rangle\!\downarrow$ as conclusion, so either $\langle r, \sigma\rangle\!\downarrow$ or $\langle \bar{r}, \sigma\rangle\!\downarrow$. If $\langle r, \sigma\rangle\!\downarrow$, then since $(r, s) \in S_n$, by the inductive hypothesis, $\langle s, \sigma\rangle\!\downarrow$. Similarly, if $\langle \bar{r}, \sigma\rangle\!\downarrow$, then $\langle \bar{s}, \sigma\rangle\!\downarrow$. In both cases, by Rule $\langle\textsf{alt-term}\rangle$, we obtain $\langle s + \bar{s}, \sigma\rangle\!\downarrow$.

Cond. $\langle\textsf{str-tran}^{\Delta}\rangle$**:** Suppose now $\langle x, \sigma\rangle \xrightarrow{a} \langle x', \sigma'\rangle$. We prove, by induction on n, that $\langle y, \sigma\rangle \xrightarrow{a} \langle y', \sigma'\rangle$ for some $y' \in P$ and that $(x', y') \in R$. Suppose $n = 1$. Then either $x = p^*$ and $y = q^*$, or

$x = r \cdot p^*$ and $y = s \cdot q^*$ with $(r, s) \in S_{pq}$. We treat these case separately.

The final rule in any derivation with $\langle p^*, \sigma\rangle \xrightarrow{a} \langle x', \sigma'\rangle$ as conclusion must be Rule $\langle\mathsf{rep\text{-}tran}\rangle$ and so, we obtain that $\langle p, \sigma\rangle \xrightarrow{a} \langle p', \sigma'\rangle$ and $x' = p' \cdot p^*$. Since $(p, q) \in S_{pq}$, there exists a $q' \in P$ such that $\langle q, \sigma\rangle \xrightarrow{a} \langle q', \sigma'\rangle$ and $(p', q') \in R_{pq}$. By Rule $\langle\mathsf{rep\text{-}tran}\rangle$ we have $\langle q^*, \sigma\rangle \xrightarrow{a} \langle q' \cdot q^*, \sigma'\rangle$. Since $(p', q') \in R_{pq}$, according to the definition of R, we have $(p' \cdot p^*, q' \cdot q^*) \in R$.

The final rule in any derivation with $\langle r \cdot p^*, \sigma\rangle \xrightarrow{a} \langle x', \sigma'\rangle$ as conclusion is either Rule $\langle\mathsf{seq\text{-}tran}_1\rangle$ or Rule $\langle\mathsf{seq\text{-}tran}_2\rangle$.

If Rule $\langle\mathsf{seq\text{-}tran}_1\rangle$ is the final rule applied, we get $\langle r, \sigma\rangle\downarrow$, $\langle p^*, \sigma\rangle \xrightarrow{a} \langle t, \sigma'\rangle$ and $x' = t$ for some $t \in P$. From this, by Rule $\langle\mathsf{rep\text{-}tran}\rangle$, $\langle p, \sigma\rangle \xrightarrow{a} \langle p', \sigma'\rangle$ and $t = p' \cdot p^*$ for some $p' \in P$. Since $(r, s) \in S_{pq}$, we have $\langle s, \sigma\rangle\downarrow$. Since $(p, q) \in S_{pq}$, there exists a $q' \in P$ such that $\langle q, \sigma\rangle \xrightarrow{a} \langle q', \sigma'\rangle$ and $(p', q') \in R_{pq}$. By Rule $\langle\mathsf{rep\text{-}tran}\rangle$, $\langle q^*, \sigma\rangle \xrightarrow{a} \langle q' \cdot q^*, \sigma'\rangle$. Finally, by Rule $\langle\mathsf{seq\text{-}tran}_1\rangle$, $\langle s \cdot q^*, \sigma\rangle \xrightarrow{a} \langle q' \cdot q^*, \sigma'\rangle$. From $(p', q') \in R_{pq}$ and the definition of R, $(p' \cdot p^*, q' \cdot q^*) \in R$.

If Rule $\langle\mathsf{seq\text{-}tran}_2\rangle$ is the final rule applied, then $\langle r, \sigma\rangle \xrightarrow{a} \langle r', \sigma'\rangle$ and $x' = r' \cdot p^*$. Since $(r, s) \in R_{pq}$, we have that $\langle s, \sigma\rangle \xrightarrow{a} \langle s', \sigma'\rangle$ and $(r', s') \in R_{pq}$ for some $s' \in P$. By Rule $\langle\mathsf{seq\text{-}tran}_2\rangle$, we obtain $\langle s \cdot q^*, \sigma\rangle \xrightarrow{a} \langle s' \cdot q^*, \sigma'\rangle$. That $(r' \cdot p^*, s' \cdot q^*) \in R$ follows from $(r', s') \in R_{pq}$ and the definition of R.

Suppose the statement holds for $k \leqslant n$ and suppose $(x, y) \in S_{n+1}$. We can assume that $x = r + \bar{r}$ and $y = s + \bar{s}$ where $(r, s) \in S_n$ and $(\bar{r}, \bar{s}) \in S_n$. Since Rule $\langle\mathsf{alt\text{-}tran}\rangle$ must be the final rule in any derivation with $\langle r + \bar{r}, \sigma\rangle \xrightarrow{a} \langle x', \sigma'\rangle$ as conclusion, we obtain that either $\langle r, \sigma\rangle \xrightarrow{a} \langle r', \sigma'\rangle$ and $x' = r'$ for some $r' \in P$, or that $\langle \bar{r}, \sigma\rangle \xrightarrow{a} \langle \bar{r}', \sigma'\rangle$ and $x' = \bar{r}'$ for some $\bar{s}' \in P$. We only treat the first case; the second one is symmetric. By the inductive hypothesis, there exists an $s' \in P$ such that $\langle s, \sigma\rangle \xrightarrow{a} \langle s', \sigma'\rangle$ and $(r', s') \in R$. By Rule $\langle\mathsf{alt\text{-}tran}\rangle$, $\langle s + \bar{s}, \sigma\rangle \xrightarrow{a} \langle s', \sigma'\rangle$.

Cond. $\langle\mathsf{str\text{-}tick}^{\Delta}\rangle$**:** Suppose now $\langle x, \sigma\rangle \overset{\Delta}{\mapsto} \langle x', \sigma'\rangle$. We prove, by induction on n, that $\langle y, \sigma\rangle \overset{\Delta}{\mapsto} \langle y', \sigma'\rangle$ for some $y' \in P$ such that $(x', y') \in S$. Suppose $n = 1$. Then either $x = p^*$ and $y = q^*$, or $x = r \cdot p^*$ and $y = s \cdot q^*$ with $(r, s) \in S_{pq}$. We treat these case separately.

The final rule in any derivation with $\langle p^*, \sigma\rangle \overset{\Delta}{\mapsto} \langle x', \sigma'\rangle$ as conclusion must be Rule $\langle\mathsf{rep\text{-}tick}\rangle$ and so, we obtain that $\langle p, \sigma\rangle \overset{\Delta}{\mapsto}$

$\langle p', \sigma' \rangle$ and $x' = p' \cdot p^*$. Since $(p, q) \in S_{pq}$, there exists a $q' \in P$ such that $\langle q, \sigma \rangle \overset{\Delta}{\mapsto} \langle q', \sigma' \rangle$ and $(p', q') \in S_{pq}$. By Rule $\langle \textsf{rep-tick} \rangle$ we have $\langle q^*, \sigma \rangle \overset{\Delta}{\mapsto} \langle q' \cdot q^*, \sigma' \rangle$. Since $(p', q') \in S_{pq}$, according to the definition of S, we have $(p' \cdot p^*, q' \cdot q^*) \in S$.

For the second case note that only Rules $\langle \textsf{seq-tick}_1 \rangle$, $\langle \textsf{seq-tick}_2 \rangle$ $\langle \textsf{seq-tick}_3 \rangle$ or $\langle \textsf{seq-tick}_4 \rangle$ can be the final rules in a derivation with $\langle r \cdot p^*, \sigma \rangle \overset{\Delta}{\mapsto} \langle x', \sigma' \rangle$ as conclusion. We treat them separately.

If Rule $\langle \textsf{seq-tick}_1 \rangle$ is the final rule applied, then $\langle r, \sigma \rangle \overset{\Delta}{\mapsto} \langle r', \sigma' \rangle$, $\langle r, \sigma \rangle \not\downarrow$ and $x' = r' \cdot p^*$ for some $r' \in P$. Since $(r, s) \in S_{pq}$, we have $\langle s, \sigma \rangle \not\downarrow$ and that there exists an $s' \in P$ such that $\langle s, \sigma \rangle \overset{\Delta}{\mapsto} \langle s', \sigma' \rangle$ and $(r', s') \in S_{pq}$. By Rule $\langle \textsf{seq-tick}_1 \rangle$, $\langle s \cdot q^*, \sigma \rangle \overset{\Delta}{\mapsto} \langle s' \cdot q^*, \sigma' \rangle$. Since $(r', s') \in S_{pq}$, by the definition of S, $(r' \cdot p^*, s' \cdot q^*) \in S$.

If Rule $\langle \textsf{seq-tick}_2 \rangle$ is the final rule applied, then $\langle r, \sigma \rangle \overset{\Delta}{\mapsto} \langle r', \sigma' \rangle$, $\langle p^*, \sigma \rangle \overset{\Delta}{\not\mapsto}$ and $x' = r' \cdot p^*$ for some $r' \in P$. Since $(r, s) \in S_{pq}$, there exists an $s' \in P$ such that $\langle s, \sigma \rangle \overset{\Delta}{\mapsto} \langle s', \sigma' \rangle$ and $(r', s') \in S_{pq}$. By Rule $\langle \textsf{rep-tick} \rangle$, from $\langle p^*, \sigma \rangle \overset{\Delta}{\not\mapsto}$ we have $\langle p, \sigma \rangle \overset{\Delta}{\not\mapsto}$. Since $(p, q) \in S_{pq}$, we have $\langle q, \sigma \rangle \overset{\Delta}{\not\mapsto}$ and then, by Rule $\langle \textsf{rep-tick} \rangle$, that $\langle q^*, \sigma \rangle \overset{\Delta}{\not\mapsto}$. By Rule $\langle \textsf{seq-tick}_2 \rangle$, $\langle s \cdot q^*, \sigma \rangle \overset{\Delta}{\mapsto} \langle s' \cdot q^*, \sigma' \rangle$. From $(r', s') \in S_{pq}$, according to the definition of S, $(r' \cdot p^*, s' \cdot q^*) \in S$.

If Rule $\langle \textsf{seq-tick}_3 \rangle$ is the final rule applied, then we have $\langle r, \sigma \rangle \downarrow$, $\langle r, \sigma \rangle \overset{\Delta}{\not\mapsto}$, $\langle p^*, \sigma \rangle \overset{\Delta}{\mapsto} \langle u, \sigma' \rangle$ and $x' = t$ for some $t \in P$. By Rule $\langle \textsf{rep-tick} \rangle$, $\langle p, \sigma \rangle \overset{\Delta}{\mapsto} \langle p', \sigma' \rangle$ and $t = p' \cdot p^*$ for some $p' \in P$. Since $(r, s) \in S_{pq}$, we have $\langle s, \sigma \rangle \downarrow$ and $\langle s, \sigma \rangle \overset{\Delta}{\not\mapsto}$. Since $(p, q) \in S_{pq}$, there exists a $q' \in P$ such that $\langle q, \sigma \rangle \overset{\Delta}{\mapsto} \langle q', \sigma' \rangle$ and $(p', q') \in S_{pq}$. By Rule $\langle \textsf{rep-tick} \rangle$, $\langle q^*, \sigma \rangle \overset{\Delta}{\mapsto} \langle q' \cdot q^*, \sigma' \rangle$. By Rule $\langle \textsf{seq-tick}_3 \rangle$, $\langle s \cdot q^*, \sigma \rangle \overset{\Delta}{\mapsto} \langle q' \cdot q^*, \sigma' \rangle$. According to the definition of S, from $(p', q') \in S_{pq}$ we have $(p', q') \in S$.

If Rule $\langle \textsf{seq-tick}_4 \rangle$ is the final rule applied, then $\langle r, \sigma \rangle \downarrow$, $\langle r, \sigma \rangle \overset{\Delta}{\mapsto} \langle r', \sigma' \rangle$, $\langle p^*, \sigma \rangle \overset{\Delta}{\mapsto} \langle t, \sigma' \rangle$ and $x' = r \cdot p^* + t$ for some $r', t \in P$. By Rule $\langle \textsf{rep-tick} \rangle$, $\langle p, \sigma \rangle \overset{\Delta}{\mapsto} \langle p', \sigma' \rangle$ and $u = p' \cdot p^*$ for some $p' \in P$. Since $(r, s) \in S_{pq}$, we have that $\langle s, \sigma \rangle \downarrow$ and that there exists an $s' \in P$ such that $\langle s, \sigma \rangle \overset{\Delta}{\mapsto} \langle s', \sigma' \rangle$, $(r', s') \in S_{pq}$. Since $(p, q) \in S_{pq}$, there exists a $q' \in P$ such that $\langle q, \sigma \rangle \overset{\Delta}{\mapsto} \langle q', \sigma' \rangle$ and $(p', q') \in S_{pq}$. By Rule $\langle \textsf{rep-tick} \rangle$, $\langle q^*, \sigma \rangle \overset{\Delta}{\mapsto} \langle q' \cdot q^*, \sigma' \rangle$. By Rule $\langle \textsf{seq-tick}_4 \rangle$,

$\langle s \cdot q^*, \sigma\rangle \overset{\Delta}{\mapsto} \langle s' \cdot q^* + q' \cdot q^*, \sigma'\rangle$. Since $(r', s') \in S_{pq}$, we have $(r' \cdot p^*, s' \cdot q^*) \in S_1$. Since $(p', q') \in S_{pq}$, $(p' \cdot p^*, q' \cdot q^*) \in S_1$. By the definition of S, $(r' \cdot p^* + p' \cdot p^*, s' \cdot q^* + q' \cdot q^*) \in S_2 \subseteq S$.

Suppose the statement holds for $k \leqslant n$ and suppose $(x, y) \in S_{n+1}$. Then $x = r + \bar{r}$ and $y = s + \bar{s}$ where $(r, s) \in S_n$ and $(\bar{r}, \bar{s}) \in S_n$. Rules $\langle\mathsf{alt\text{-}tick}_1\rangle$ and $\langle\mathsf{alt\text{-}tick}_2\rangle$ must be the final rules in any derivation with $\langle r + \bar{r}, \sigma\rangle \overset{\Delta}{\mapsto} \langle x', \sigma'\rangle$ as conclusion.

If Rule $\langle\mathsf{alt\text{-}tick}_1\rangle$ is the final rule applied, then $\langle r, \sigma\rangle \overset{\Delta}{\mapsto} \langle r', \sigma'\rangle$, $\langle \bar{r}, \sigma\rangle \overset{\Delta}{\not\mapsto}$ and $x' = r'$ for some $r' \in P$ (or the symmetric case). By the inductive hypothesis, there exist a $s' \in P$ such that $\langle s, \sigma\rangle \overset{\Delta}{\mapsto} \langle s', \sigma'\rangle$ and $(r', s') \in S$. Since $(\bar{r}, \bar{s}) \in S_n$, from the inductive hypothesis and a simple contradiction it follows that $\langle \bar{s}, \sigma\rangle \overset{\Delta}{\not\mapsto}$. By Rule $\langle\mathsf{alt\text{-}tick}_1\rangle$, $\langle s + \bar{s}, \sigma\rangle \overset{\Delta}{\mapsto} \langle s', \sigma'\rangle$.

If Rule $\langle\mathsf{alt\text{-}tick}_2\rangle$ is the final rule applied, then $\langle r, \sigma\rangle \overset{\Delta}{\mapsto} \langle r', \sigma'\rangle$, $\langle \bar{r}, \sigma\rangle \overset{\Delta}{\mapsto} \langle \bar{r}', \sigma'\rangle$ and $x' = r' + \bar{r}'$ for some $r', \bar{r}' \in P$. By the inductive hypothesis, there exist a $s' \in P$ such that $\langle s, \sigma\rangle \overset{\Delta}{\mapsto} \langle s', \sigma'\rangle$ and $(r', s') \in S$. Similarly, there exists an $\bar{s}' \in P$ such that $\langle \bar{s}, \sigma\rangle \overset{\Delta}{\mapsto} \langle \bar{s}', \sigma'\rangle$ and $(\bar{r}', \bar{s}') \in S$. By Rule $\langle\mathsf{alt\text{-}tick}_1\rangle$, $\langle s + \bar{s}, \sigma\rangle \overset{\Delta}{\mapsto} \langle s' + \bar{s}', \sigma'\rangle$. Suppose $(r', s') \in S_m$ and $(\bar{r}', \bar{s}') \in S_{\bar{m}}$ for some $m \geqslant 1$, $\bar{m} \geqslant 1$. By the definition of S, we have $(\bar{r}', \bar{s}') \in S_{\max(m,\bar{m})} \subseteq S$.

We now prove that R satisfies the conditions of Definition 5.1.4. It is enough to show that the pairs in $\{(r \cdot p^*, s \cdot q^*) \mid (r, s) \in R_{pq}\} \cup \{(r \cdot q^*, s \cdot p^*) \mid (r, s) \in R_{pq}\}$ satisfy $\langle\mathsf{sl\text{-}term}^\Delta\rangle$ and $\langle\mathsf{sl\text{-}tick}^\Delta\rangle$.

Cond. $\langle\mathsf{sl\text{-}term}^\Delta\rangle$**:** Suppose $\langle r \cdot p^*, \sigma\rangle\downarrow$. By Rules $\langle\mathsf{seq\text{-}term}\rangle$, $\langle r, \sigma\rangle\downarrow$. Since $(r, s) \in R_{pq}$, there exists an $s' \in P$ such that $\langle s, \sigma\rangle \twoheadrightarrow \langle s', \sigma\rangle$, $\langle s', \sigma\rangle\downarrow$ and $(r, s') \in R_{pq}$. In addition, if $\langle r, \sigma\rangle \overset{\Delta}{\not\mapsto}$, then $\langle s', \sigma\rangle \overset{\Delta}{\not\mapsto}$. Using Rules $\langle\mathsf{seq\text{-}tran}_2\rangle$, $\langle\mathsf{seq\text{-}term}\rangle$ and $\langle\mathsf{rep\text{-}term}\rangle$, we have $\langle s \cdot q^*, \sigma\rangle \twoheadrightarrow \langle s' \cdot q^*, \sigma\rangle$ and $\langle s' \cdot q^*, \sigma\rangle\downarrow$. Suppose now that $\langle r \cdot p^*, \sigma\rangle \overset{\Delta}{\not\mapsto}$. Because $\langle r, \sigma\rangle\downarrow$, this is equivalent to $\langle r, \sigma\rangle \overset{\Delta}{\not\mapsto}$ and $\langle p^*, \sigma\rangle \overset{\Delta}{\not\mapsto}$. From $\langle r, \sigma\rangle \overset{\Delta}{\not\mapsto}$ we obtain $\langle s', \sigma\rangle \overset{\Delta}{\not\mapsto}$. It easily follows, by Rule $\langle\mathsf{rep\text{-}tick}\rangle$, that $\langle p, \sigma\rangle \overset{\Delta}{\not\mapsto}$. Since $(p, q) \in S_{pq}$, we have $\langle q, \sigma\rangle \overset{\Delta}{\not\mapsto}$. From this $\langle q^*, \sigma\rangle \overset{\Delta}{\not\mapsto}$. From $\langle s', \sigma\rangle \overset{\Delta}{\not\mapsto}$ and $\langle p^*, \sigma\rangle \overset{\Delta}{\not\mapsto}$ it follows that $\langle s' \cdot q^*, \sigma\rangle \overset{\Delta}{\not\mapsto}$.

Cond. $\langle \mathsf{sl\text{-}tick}^{\Delta} \rangle$**:** Suppose $\langle r \cdot p^*, \sigma\rangle \overset{\Delta}{\mapsto} \langle t, \sigma'\rangle$ for some $t \in P$. There are four rules with this transition as conclusion, namely $\langle \mathsf{seq\text{-}tick}_1\rangle$, $\langle \mathsf{seq\text{-}tick}_2\rangle$, $\langle \mathsf{seq\text{-}tick}_3\rangle$, $\langle \mathsf{seq\text{-}tick}_4\rangle$. We treat them separately.

If the final rule applied is Rule $\langle \mathsf{seq\text{-}tick}_1\rangle$, then $\langle r, \sigma\rangle \not\downarrow$, $\langle r, \sigma\rangle \overset{\Delta}{\mapsto} \langle r', \sigma'\rangle$ for some $r' \in P$, and $t = r' \cdot p^*$. Then there exist $s', s'' \in P$ such that $\langle s, \sigma\rangle \twoheadrightarrow \langle s'', \sigma\rangle \overset{\Delta}{\mapsto} \langle s', \sigma'\rangle$, $(r, s'') \in R_{pq}$ and $(r', s') \in R_{pq}$. Since $\langle r, \sigma\rangle \not\downarrow$, we also have that $\langle s'', \sigma\rangle \not\downarrow$. By Rules $\langle \mathsf{seq\text{-}tran}_2\rangle$ and Rule $\langle \mathsf{seq\text{-}tick}_1\rangle$, we obtain $\langle s \cdot q^*, \sigma\rangle \twoheadrightarrow \langle s'' \cdot q^*, \sigma\rangle \overset{\Delta}{\mapsto} \langle s' \cdot q^*, \sigma'\rangle$. Since $(r, s'') \in R_{pq}$ and $(r', s') \in R_{pq}$, by the definition of R, $(r \cdot p^*, s'' \cdot q^*) \in R$ and $(r' \cdot p^*, s' \cdot q^*) \in R$. Note that, since $\langle r, \sigma\rangle \not\downarrow$, by Rule $\langle \mathsf{seq\text{-}term}\rangle$ we have $\langle r \cdot p^*, \sigma\rangle \not\downarrow$, and so we only need to show that $\langle s \cdot q^*, \sigma\rangle \not\downarrow$. This however follows directly from $\langle s, \sigma\rangle \not\downarrow$.

If the final rule applied is Rule $\langle \mathsf{seq\text{-}tick}_2\rangle$, then $\langle r, \sigma\rangle\downarrow$, $\langle r, \sigma\rangle \overset{\Delta}{\mapsto} \langle r', \sigma'\rangle$ for some $r' \in P$, and $\langle p^*, \sigma\rangle \overset{\Delta}{\not\mapsto}$. Also $t = r' \cdot p^*$. Then $\langle s, \sigma\rangle \twoheadrightarrow \langle s'', \sigma\rangle \overset{\Delta}{\mapsto} \langle s', \sigma'\rangle$, $(r, s'') \in R_{pq}$ and $(r', s') \in R_{pq}$ for some $s', s'' \in P$. Since $\langle r, \sigma\rangle\downarrow$, we have $\langle s'', \sigma\rangle\downarrow$ and $(r', s') \in S_{pq}$. By Rule $\langle \mathsf{rep\text{-}tick}\rangle$, $\langle p, \sigma\rangle \overset{\Delta}{\not\mapsto}$. Since $(p, q) \in S_{pq}$, we obtain $\langle q, \sigma\rangle \overset{\Delta}{\not\mapsto}$ which implies $\langle q^*, \sigma\rangle \overset{\Delta}{\not\mapsto}$. By Rule $\langle \mathsf{seq\text{-}tran}_2\rangle$ and Rule $\langle \mathsf{seq\text{-}tick}_2\rangle$, $\langle s \cdot q^*, \sigma\rangle \twoheadrightarrow \langle s'' \cdot q^*, \sigma\rangle \overset{\Delta}{\mapsto} \langle s' \cdot q^*, \sigma'\rangle$. Since $(r, s'') \in R_{pq}$, by the definition of R, $(r \cdot p^*, s'' \cdot q^*) \in R$. Since $(r', s') \in S_{pq}$, we have that $(r' \cdot p^*, s' \cdot q^*) \in S_1 \subseteq S$. Now, suppose first that $\langle r \cdot p^*, \sigma\rangle\downarrow$. Note that, since $\langle r, \sigma\rangle\downarrow$, and since, by Rule $\langle \mathsf{rep\text{-}term}\rangle$, $\langle p^*, \sigma\rangle\downarrow$, we have $\langle r \cdot p^*, \sigma\rangle\downarrow$. From $\langle s'', \sigma\rangle\downarrow$ and $\langle q^*, \sigma\rangle\downarrow$, by Rule $\langle \mathsf{seq\text{-}term}\rangle$, we obtain $\langle s'' \cdot q^*, \sigma\rangle\downarrow$.

If the final rule applied is Rule $\langle \mathsf{seq\text{-}tick}_3\rangle$, then $\langle r, \sigma\rangle\downarrow$, $\langle r, \sigma\rangle \overset{\Delta}{\not\mapsto}$ and $\langle p^*, \sigma\rangle \overset{\Delta}{\mapsto} \langle u, \sigma'\rangle$ for some $u \in P$. Then also $t = u$. By Rule $\langle \mathsf{rep\text{-}tick}\rangle$, we have that there exists a $p' \in P$ such that $\langle p^*, \sigma\rangle \overset{\Delta}{\mapsto} \langle p', \sigma'\rangle$ and $u = p' \cdot p^*$. Since $(p, q) \in S_{pq}$, there is a $q' \in P$ such that $\langle q, \sigma\rangle \overset{\Delta}{\mapsto} \langle q', \sigma'\rangle$ and $(p', q') \in S_{pq}$. By Rule $\langle \mathsf{rep\text{-}tick}\rangle$, $\langle q^*, \sigma\rangle \overset{\Delta}{\mapsto} \langle q' \cdot q^*, \sigma'\rangle$. From $\langle r, \sigma\rangle\downarrow$ it follows that there exists an $s' \in P$ such that $\langle s, \sigma\rangle \twoheadrightarrow \langle s', \sigma\rangle$, $\langle s', \sigma\rangle\downarrow$ and $(r, s) \in R_{pq}$. From $\langle r, \sigma\rangle \overset{\Delta}{\not\mapsto}$ it follows that $\langle s', \sigma\rangle \overset{\Delta}{\not\mapsto}$. By Rules $\langle \mathsf{seq\text{-}tran}_2\rangle$ and $\langle \mathsf{seq\text{-}tick}_3\rangle$, $\langle s \cdot q^*, \sigma\rangle \twoheadrightarrow \langle s' \cdot q^*, \sigma\rangle \overset{\Delta}{\mapsto} \langle q' \cdot q^*, \sigma'\rangle$. By the definition of R, from $(r, s') \in R_{pq}$ it follows that $(r \cdot p^*, s' \cdot q^*) \in R$. Since $(p, q) \in S_{pq}$, by the definition of S, $(p' \cdot p^*, q' \cdot q^*) \in S$. As in the previous case we have that $\langle r \cdot p^*, \sigma\rangle\downarrow$ and that $\langle s' \cdot q^*, \sigma\rangle\downarrow$. If the final rule applied is Rule $\langle\textsf{seq-tick}_4\rangle$, then $\langle r, \sigma\rangle\downarrow$, $\langle r, \sigma\rangle \overset{\Delta}{\mapsto} \langle r', \sigma'\rangle$ for some $r' \in P$, and $\langle p^*, \sigma\rangle \overset{\Delta}{\mapsto} \langle u, \sigma'\rangle$ for some $u \in P$. In addition, $t = r' \cdot p^* + u$. By Rule $\langle\textsf{rep-tick}\rangle$, we have that there exists a $p' \in P$ such that $\langle p^*, \sigma\rangle \overset{\Delta}{\mapsto} \langle p', \sigma'\rangle$ and $u = p' \cdot p^*$. Since $(p, q) \in S_{pq}$, there is a $q' \in P$ such that $\langle q, \sigma\rangle \overset{\Delta}{\mapsto} \langle q', \sigma'\rangle$ and $(p', q') \in S_{pq}$. By Rule $\langle\textsf{rep-tick}\rangle$, $\langle q^*, \sigma\rangle \overset{\Delta}{\mapsto} \langle q' \cdot q^*, \sigma'\rangle$. Since $(r, s) \in R_{pq}$, there exist a $s', s'' \in P$ such that $\langle s, \sigma\rangle \twoheadrightarrow \langle s'', \sigma\rangle \overset{\Delta}{\mapsto} \langle s', \sigma'\rangle$, $(r, s'') \in R_{pq}$ and $(r', s') \in R_{pq}$. From $\langle r, \sigma\rangle\downarrow$ we have that $\langle s'', \sigma\rangle\downarrow$ and that $(r', s') \in S_{pq}$. By Rule $\langle\textsf{seq-tran}_2\rangle$ and Rule $\langle\textsf{seq-tick}_4\rangle$, $\langle s \cdot q^*, \sigma\rangle \twoheadrightarrow \langle s'' \cdot q^*, \sigma\rangle \overset{\Delta}{\mapsto} \langle s' \cdot q^* + q' \cdot q^*, \sigma'\rangle$. Since $(r, s'') \in R_{pq}$, by the definition of R, $(r \cdot p^*, s'' \cdot q^*) \in R$. Since $(r', s') \in S_{pq}$ and $(p', q') \in S_{pq}$, by the definition of S_2, that $(r' \cdot p^* + p' \cdot p^*, s' \cdot q^* + q' \cdot q^*) \in S_2 \subseteq S$. Like in the previous two cases we have $\langle r \cdot p^*, \sigma\rangle\downarrow$ and $\langle s' \cdot q^*, \sigma\rangle\downarrow$.

5. Let

$$R = R_{pq} \cup R_{\bar{p}\bar{q}} \cup \{(r \parallel \bar{r}, s \parallel \bar{s}) \mid (r, s) \in R_{pq}, (\bar{r}, \bar{s}) \in R_{\bar{p}\bar{q}}\}$$

and

$$S = S_{pq} \cup S_{\bar{p}\bar{q}} \cup \{(r \parallel \bar{r}, s \parallel \bar{s}) \mid (r, s) \in S_{pq}, (\bar{r}, \bar{s}) \in S_{\bar{p}\bar{q}}\}\,.$$

Note that in the proof of Theorem 4.3.7(5) we did not need to include the relations R_{pq} and $R_{\bar{p}\bar{q}}$ in the definition of R. Here, however, we have to because of Rule $\langle\textsf{par-tick}_1\rangle$.

First we show that S is a relation of strong pairs in R. It is enough to consider only the pairs from $\{(r \parallel \bar{r}, s \parallel \bar{s}) \mid (r, s) \in S_{pq}, (\bar{r}, \bar{s}) \in S_{\bar{p}\bar{q}}\}$.

Cond. $\langle\textsf{str-term}^\Delta\rangle$**:** Suppose $\langle r \parallel \bar{r}, \sigma\rangle\downarrow$ for some $\sigma \in \Sigma$. Because Rule $\langle\textsf{par-term}\rangle$ is the final rule with this as conclusion, we have $\langle r, \sigma\rangle\downarrow$ and $\langle \bar{r}, \sigma\rangle\downarrow$. Since $(r, s) \in S_{pq}$ and $(\bar{r}, \bar{s}) \in S_{\bar{p}\bar{q}}$, we obtain $\langle s, \sigma\rangle\downarrow$ and $\langle \bar{s}, \sigma\rangle\downarrow$, and hence, by Rule $\langle\textsf{par-term}\rangle$ again, $\langle s \parallel \bar{s}, \sigma\rangle\downarrow$.

Cond. $\langle\textsf{str-tran}^\Delta\rangle$**:** Suppose $\langle r \parallel \bar{r}, \sigma\rangle \xrightarrow{a} \langle t, \sigma'\rangle$ for some $\sigma, \sigma' \in \Sigma$ and $t \in P$. The final rule of a derivation with this transition as conclusion is either Rule $\langle\textsf{par-tran}_1\rangle$ or Rule $\langle\textsf{par-tran}_2\rangle$; we treat these cases separately.

If the final rule applied is Rule $\langle\mathsf{par\text{-}tran}_1\rangle$, then $\langle r,\sigma\rangle \xrightarrow{a} \langle r',\sigma'\rangle$ and $t = r' \parallel \bar{r}$; (or symmetrically $\langle \bar{r},\sigma\rangle \xrightarrow{a} \langle \bar{r}',\sigma'\rangle$ and $t = r \parallel \bar{r}'$). Since $(r,s) \in S_{pq}$, there exists $s' \in P$ such that $\langle s,\sigma\rangle \xrightarrow{a} \langle s',\sigma'\rangle$ and $(r',s') \in R_{pq}$. Hence, by Rule $\langle\mathsf{par\text{-}tran}_1\rangle$, $\langle s \parallel \bar{s},\sigma\rangle \xrightarrow{a} \langle s' \parallel \bar{s},\sigma'\rangle$. By the definition of R, since $(r',s') \in R_{pq}$, we have $(r' \parallel \bar{r}, s' \parallel \bar{s}) \in R$.

If the final rule applied is Rule $\langle\mathsf{par\text{-}tran}_2\rangle$, we have that $\langle r,\sigma\rangle \xrightarrow{b} \langle r',\sigma''\rangle$ and $\langle \bar{r},\sigma\rangle \xrightarrow{c} \langle \bar{r}',\sigma''\rangle$, for some $b,c \in \mathsf{A}$, $\sigma'',\sigma''' \in \Sigma$ with $a = \mathsf{act}(\mathsf{comm}(b,c),\sigma)$, $\sigma' = \mathsf{eff}(a,\sigma) = \sigma'' \ll \sigma''' = \sigma''' \ll \sigma''$ and $t = r' \parallel \bar{r}'$. Since $(r,s) \in S_{pq}$ and $(\bar{r},\bar{s}) \in S_{\bar{p}\bar{q}}$, there exist $s',\bar{s}' \in P$ such that $\langle s,\sigma\rangle \xrightarrow{b} \langle s',\sigma''\rangle$ and $\langle \bar{s},\sigma\rangle \xrightarrow{c} \langle \bar{s}',\sigma\rangle\sigma'''$, $(r',s') \in R_{pq}$ and $(\bar{r}',\bar{s}') \in R_{\bar{p}\bar{q}}$. By Rule $\langle\mathsf{par\text{-}tran}_2\rangle$, $\langle s \parallel \bar{s},\sigma\rangle \xrightarrow{a} \langle s' \parallel \bar{s}',\sigma'\rangle$. By the definition of R, from $(r',s') \in R_{pq}$ and $(\bar{r}',\bar{s}') \in R_{\bar{p}\bar{q}}$ we have $(r' \parallel \bar{r}', s' \parallel \bar{s}') \in R$.

Cond. $\langle\mathsf{str\text{-}tick}^{\Delta}\rangle$**:** Suppose $\langle r \parallel \bar{r},\sigma\rangle \overset{\Delta}{\mapsto} \langle t,\sigma'\rangle$ for some $\sigma,\sigma' \in \Sigma$ and $t \in P$. The final rule of any derivation with this transition as conclusion is either Rule $\langle\mathsf{par\text{-}tick}_1\rangle$ or Rule $\langle\mathsf{par\text{-}tick}_2\rangle$; we treat the cases separately.

If Rule $\langle\mathsf{par\text{-}tick}_1\rangle$ is the final rule applied, then $\langle r,\sigma\rangle \overset{\Delta}{\mapsto} \langle r',\sigma'\rangle$, $\langle \bar{r},\sigma\rangle\!\downarrow$, $\langle \bar{r},\sigma\rangle \overset{\Delta}{\not\mapsto}$ and $t = r'$ (or the symmetric case). Because $(r,s) \in S_{pq}$, there exists $s' \in P$ such that $\langle s,\sigma\rangle \overset{\Delta}{\mapsto} \langle s',\sigma'\rangle$ and $(r',s') \in S_{pq}$. Since $(\bar{r},\bar{s}) \in S_{\bar{p}\bar{q}}$, we have $\langle \bar{s},\sigma\rangle\!\downarrow$ and $\langle \bar{s},\sigma\rangle \overset{\Delta}{\not\mapsto}$. By Rule $\langle\mathsf{par\text{-}tick}_1\rangle$, we have $\langle s \parallel \bar{s},\sigma\rangle \overset{\Delta}{\mapsto} \langle s',\sigma'\rangle$. Since $(r',s') \in S_{pq}$, according to the definition of S, $(r',s') \in S$.

If Rule $\langle\mathsf{par\text{-}tick}_2\rangle$ is the final rule applied, then $\langle r,\sigma\rangle \overset{\Delta}{\mapsto} \langle r',\sigma'\rangle$, $\langle \bar{r},\sigma\rangle \overset{\Delta}{\mapsto} \langle \bar{r}',\sigma'\rangle$ and $t = r' \parallel \bar{r}'$; (or the symmetric case). Since $(r,s) \in S_{pq}$, there exists an $s' \in P$ such that $\langle s,\sigma\rangle \overset{\Delta}{\mapsto} \langle s',\sigma'\rangle$ and $(r',s') \in S_{pq}$. Since $(\bar{r},\bar{s}) \in S_{\bar{p}\bar{q}}$, there exists an $\bar{s}' \in P$ such that $\langle \bar{s},\sigma\rangle \overset{\Delta}{\mapsto} \langle \bar{s}',\sigma'\rangle$ and $(\bar{r}',\bar{s}') \in S_{\bar{p}\bar{q}}$. By Rule $\langle\mathsf{par\text{-}tick}_2\rangle$, $\langle s \parallel \bar{s},\sigma\rangle \overset{\Delta}{\mapsto} \langle s' \parallel \bar{s},\sigma'\rangle$. Since $(r',s') \in S_{pq}$ and $(\bar{r}',\bar{s}') \in S_{\bar{p}\bar{q}}$, according to the definition of S, we have $(r' \parallel \bar{r}, s' \parallel \bar{s}) \in S$.

We now prove that R satisfies $\langle\mathsf{sl\text{-}term}^{\Delta}\rangle$ and $\langle\mathsf{sl\text{-}tick}^{\Delta}\rangle$. We check these conditions only for the pairs in $\{(r \parallel \bar{r}, s \parallel \bar{s}) \mid (r,s) \in R_{pq}, (\bar{r},\bar{s}) \in R_{\bar{p}\bar{q}}$.

Cond. $\langle\mathsf{sl\text{-}term}^{\Delta}\rangle$**:** Suppose $\langle r \parallel \bar{r},\sigma\rangle\!\downarrow$ for some $\sigma \in \Sigma$. Rule $\langle\mathsf{par\text{-}term}\rangle$ must be the final rule with $\langle r \parallel \bar{r},\sigma\rangle\!\downarrow$ as conclusion, so we have

$\langle r, \sigma \rangle \downarrow$ and $\langle \bar{r}, \sigma \rangle \downarrow$. Since $(r, s) \in S_{pq}$ and $(\bar{r}, \bar{s}) \in S_{\bar{p}\bar{q}}$, there exist $s', \bar{s}' \in P$ such that $\langle s, \sigma \rangle \twoheadrightarrow \langle s', \sigma \rangle$, $\langle s', \sigma \rangle \downarrow$, $\langle \bar{s}, \sigma \rangle \twoheadrightarrow \langle \bar{s}', \sigma \rangle$, $\langle \bar{s}', \sigma \rangle \downarrow$, $(r, s') \in R_{pq}$ and $(\bar{r}, \bar{s}') \in R_{\bar{p}\bar{q}}$. In addition, $\langle r, \sigma \rangle \overset{\Delta}{\not\mapsto}$ and $\langle \bar{r}, \sigma \rangle \overset{\Delta}{\not\mapsto}$ imply $\langle s', \sigma \rangle \overset{\Delta}{\not\mapsto}$ and $\langle \bar{s}', \sigma \rangle \overset{\Delta}{\not\mapsto}$. By Rules $\langle \textsf{par-tran}_1 \rangle$ and $\langle \textsf{par-term} \rangle$, $\langle s \parallel \bar{s}, \sigma \rangle \twoheadrightarrow \langle s' \parallel \bar{s}, \sigma \rangle \twoheadrightarrow \langle s' \parallel \bar{s}', \sigma \rangle$ and $\langle s' \parallel \bar{s}', \sigma \rangle \downarrow$. Since $(r, s') \in R_{pq}$ and $(\bar{r}, \bar{s}') \in R_{\bar{p}\bar{q}}$, according to the definition of R, $(r \parallel \bar{r}, s' \parallel \bar{s}') \in R$. Suppose $\langle r \parallel \bar{r}, \sigma \rangle \overset{\Delta}{\not\mapsto}$. Since $\langle r, \sigma \rangle \downarrow$ and $\langle \bar{r}, \sigma \rangle \downarrow$, from Rules $\langle \textsf{par-tick}_1 \rangle$ and $\langle \textsf{par-tick}_2 \rangle$ we have that $\langle r, \sigma \rangle \overset{\Delta}{\not\mapsto}$ and $\langle \bar{r}, \sigma \rangle \overset{\Delta}{\not\mapsto}$. From this, we obtain $\langle s, \sigma \rangle \overset{\Delta}{\not\mapsto}$ and $\langle \bar{s}, \sigma \rangle \overset{\Delta}{\not\mapsto}$which implies $\langle s \parallel \bar{s}, \sigma \rangle \overset{\Delta}{\not\mapsto}$.

Cond. $\langle \textsf{sl-tick}^{\Delta} \rangle$**:** Suppose $\langle r \parallel \bar{r}, \sigma \rangle \overset{\Delta}{\mapsto} \langle t, \sigma' \rangle$ for some $\sigma, \sigma' \in \Sigma$ and $t \in P$. The final rule of any derivation with this transition as conclusion is either Rule $\langle \textsf{par-tick}_1 \rangle$ or Rule $\langle \textsf{par-tick}_2 \rangle$; we treat these cases separately.

If Rule $\langle \textsf{par-tick}_1 \rangle$ is the final rule applied, then $\langle r, \sigma \rangle \overset{\Delta}{\mapsto} \langle r', \sigma' \rangle$, $\langle \bar{r}, \sigma \rangle \downarrow$, $\langle \bar{r}, \sigma \rangle \overset{\Delta}{\not\mapsto}$ and $t = r'$; (or the symmetric case). Since $(r, s) \in R_{pq}$, there exist $s', s'' \in P$ such that $\langle s, \sigma \rangle \twoheadrightarrow \langle s'', \sigma \rangle \overset{\Delta}{\mapsto} \langle s', \sigma' \rangle$, $\langle s', \sigma \rangle \downarrow$, $(r, s'') \in R_{pq}$ and $(r', s') \in R_{pq}$. In addition, if $\langle r, \sigma \rangle \downarrow$, then $\langle s'', \sigma \rangle \downarrow$ and $(r', s') \in S_{pq}$, and if $\langle r, \sigma \rangle \not\downarrow$, then $\langle s'', \sigma \rangle \not\downarrow$. From $\langle \bar{r}, \sigma \rangle \downarrow$ and $\langle \bar{r}, \sigma \rangle \overset{\Delta}{\not\mapsto}$ we have that there exists an $\bar{s}' \in P$ such that $\langle \bar{s}, \sigma \rangle \twoheadrightarrow \langle \bar{s}', \sigma \rangle$, $\langle \bar{s}', \sigma \rangle \downarrow$, $\langle \bar{s}', \sigma \rangle \overset{\Delta}{\not\mapsto}$ and $(\bar{r}, \bar{s}') \in R_{\bar{p}\bar{q}}$. By Rule $\langle \textsf{par-tran}_1 \rangle$ and Rule $\langle \textsf{par-tick}_1 \rangle$, $\langle s \parallel \bar{s}, \sigma \rangle \twoheadrightarrow \langle s'' \parallel \bar{s}, \sigma \rangle \twoheadrightarrow \langle s'' \parallel \bar{s}', \sigma \rangle \overset{\Delta}{\mapsto} \langle s', \sigma' \rangle$. Now, since $(r, s'') \in R_{pq}$, $(\bar{r}, \bar{s}') \in R_{\bar{p}\bar{q}}$ and $(r', s') \in R_{pq}$, by the definition of R, we have $(r \parallel \bar{r}, s'' \parallel \bar{s}') \in R$ and $(r', s') \in R$. Suppose first that $\langle r \parallel \bar{r}, \sigma \rangle \downarrow$. By Rule $\langle \textsf{par-term} \rangle$, $\langle r, \sigma \rangle \downarrow$. This implies that $\langle s'', \sigma \rangle \downarrow$ and $(r', s') \in S_{pq}$. Since $\langle \bar{s}', \sigma \rangle \downarrow$, by Rule $\langle \textsf{par-term} \rangle$, $\langle s'' \parallel \bar{s}', \sigma \rangle \downarrow$. From $(r', s') \in S_{pq}$ it follows that $(r', s') \in S$. Suppose now that $\langle r \parallel \bar{r}, \sigma \rangle \not\downarrow$. Then $\langle r, \sigma \rangle \not\downarrow$. This implies that $\langle s'', \sigma \rangle \not\downarrow$. By Rule $\langle \textsf{par-term} \rangle$, $\langle s'' \parallel \bar{s}', \sigma \rangle \not\downarrow$.

If Rule $\langle \textsf{par-tick}_2 \rangle$ is the final rule applied, then $\langle r, \sigma \rangle \overset{\Delta}{\mapsto} \langle r', \sigma' \rangle$, $\langle \bar{r}, \sigma \rangle \overset{\Delta}{\mapsto} \langle \bar{r}', \sigma' \rangle$ and $t = r' \parallel \bar{r}'$. From $\langle r, \sigma \rangle \overset{\Delta}{\mapsto} \langle r', \sigma' \rangle$ it follows that there exists $s', s'' \in P$ such that $\langle s, \sigma \rangle \twoheadrightarrow \langle s'', \sigma \rangle \overset{\Delta}{\mapsto} \langle s', \sigma' \rangle$, $\langle s', \sigma \rangle \downarrow$, $(r, s'') \in R_{pq}$ and $(r', s') \in R_{pq}$. Moreover, it also follows that if $\langle r, \sigma \rangle \downarrow$, then $\langle s'', \sigma \rangle \downarrow$ and $(r', s') \in S_{pq}$, and that if $\langle r, \sigma \rangle \not\downarrow$, then $\langle s'', \sigma \rangle \not\downarrow$. Similarly, from $\langle \bar{r}, \sigma \rangle \overset{\Delta}{\mapsto} \langle \bar{r}', \sigma' \rangle$, there exist $\bar{s}', \bar{s}'' \in$

P such that $\langle \bar{s}, \sigma\rangle \twoheadrightarrow \langle \bar{s}'', \sigma\rangle \overset{\Delta}{\mapsto} \langle \bar{s}', \sigma'\rangle$, $\langle \bar{s}'', \sigma\rangle\downarrow$, $(\bar{r}, \bar{s}'') \in R_{\bar{p}\bar{q}}$ and $(\bar{r}', \bar{s}') \in R_{\bar{p}\bar{q}}$. Also, $\langle \bar{r}, \sigma\rangle\downarrow$ implies $\langle \bar{s}'', \sigma\rangle\downarrow$ and $(\bar{r}', \bar{s}') \in S_{\bar{p}\bar{q}}$, and $\langle \bar{r}, \sigma\rangle\not\downarrow$ implies $\langle \bar{s}'', \sigma\rangle\not\downarrow$. By Rule $\langle\mathsf{par\text{-}tran}_1\rangle$ and Rule $\langle\mathsf{par\text{-}tick}_1\rangle$, $\langle s \parallel \bar{s}, \sigma\rangle \twoheadrightarrow \langle s'' \parallel \bar{s}, \sigma\rangle \twoheadrightarrow \langle s'' \parallel \bar{s}'', \sigma\rangle \overset{\Delta}{\mapsto} \langle s' \parallel \bar{s}', \sigma'\rangle$. According to the definition of R, from $(r, s'') \in R_{pq}$ and $(\bar{r}, \bar{s}'') \in R_{\bar{p}\bar{q}}$ it follows that $(r \parallel \bar{r}, s'' \parallel \bar{s}'') \in R$; from $(r', s') \in R_{pq}$ and $(\bar{r}', \bar{s}') \in R_{\bar{p}\bar{q}}$, it follows that $(r' \parallel \bar{r}', s' \parallel \bar{s}') \in R$. Suppose now that $\langle r \parallel \bar{r}, \sigma\rangle\downarrow$. By Rule $\langle\mathsf{par\text{-}term}\rangle$, $\langle r, \sigma\rangle\downarrow$ and $\langle \bar{r}, \sigma\rangle\downarrow$. From this, $\langle s'', \sigma\rangle\downarrow$, $\langle \bar{s}'', \sigma\rangle\downarrow$, $(r', s') \in S_{pq}$, and $(\bar{r}', \bar{s}') \in S_{\bar{p}\bar{q}}$. From $\langle s'', \sigma\rangle\downarrow$ and $\langle \bar{s}'', \sigma\rangle\downarrow$, by Rule $\langle\mathsf{par\text{-}term}\rangle$ we have $\langle s'' \parallel \bar{s}'', \sigma\rangle\downarrow$. Since $(r', s') \in S_{pq}$ and $(\bar{r}', \bar{s}') \in S_{\bar{p}\bar{q}}$, by the definition of S, we have $(r' \parallel \bar{r}', s' \parallel \bar{s}') \in S$. Suppose $\langle r \parallel \bar{r}, \sigma\rangle\not\downarrow$. By Rule $\langle\mathsf{par\text{-}term}\rangle$, either $\langle r, \sigma\rangle\not\downarrow$ or $\langle \bar{r}, \sigma\rangle\not\downarrow$. We only treat the first case; the second one is symmetric. From $\langle r, \sigma\rangle\not\downarrow$ we have $\langle s'', \sigma\rangle\not\downarrow$. By Rule $\langle\mathsf{par\text{-}term}\rangle$, $\langle s'' \parallel \bar{s}'', \sigma\rangle\not\downarrow$.

6. Let

$$R = \{([\![\varsigma \mid r]\!], [\![\varsigma \mid s]\!]) \mid (r, s) \in R_{pq}, \varsigma \in \Sigma\}$$

and

$$S = \{([\![\varsigma \mid r]\!], [\![\varsigma \mid s]\!]) \mid (r, s) \in S_{pq}, \varsigma \in \Sigma\}.$$

First we show that S is a relation of strong pairs in R. Suppose $([\![\varsigma \mid r]\!], [\![\varsigma \mid r]\!]) \in S$.

Cond. $\langle\mathsf{str\text{-}term}^{\Delta}\rangle$**:** Suppose $\langle [\![\varsigma \mid r]\!], \sigma\rangle\downarrow$ for some $\sigma \in \Sigma$. Because Rule $\langle\mathsf{scp\text{-}term}\rangle$ must be the final rule with this as conclusion, we have $\langle r, \varsigma \ll \sigma\rangle\downarrow$. Since $(r, s) \in S_{pq}$, we have $\langle s, \varsigma \ll \sigma\rangle\downarrow$, and hence, by Rule $\langle\mathsf{scp\text{-}term}\rangle$, that $\langle [\![\varsigma \mid s]\!], \sigma\rangle\downarrow$.

Cond. $\langle\mathsf{str\text{-}tran}^{\Delta}\rangle$**:** Suppose $\langle [\![\varsigma \mid r]\!], \sigma\rangle \xrightarrow{a} \langle t, \sigma'\rangle$ for some $\sigma, \sigma' \in \Sigma$, $a \in A$ and $t \in P$. By Rule $\langle\mathsf{scp\text{-}tran}\rangle$, as the final rule with this transition as conclusion, we obtain $\langle r, \varsigma \ll \sigma\rangle \xrightarrow{a} \langle r', \sigma''\rangle$, $t = [\![\varsigma \mid r']\!]$ and $\sigma' = \sigma \ll \sigma''_{/\mathrm{dom}(\sigma)\setminus\mathrm{dom}(\varsigma)}$. Since $(r, s) \in S_{pq}$, there exists an $s' \in P$ such that $\langle s, \varsigma \ll \sigma\rangle \xrightarrow{a} \langle s', \sigma''\rangle$ and $(r', s') \in R_{pq}$. By Rule $\langle\mathsf{scp\text{-}tran}\rangle$, $\langle [\![\varsigma \mid s]\!], \sigma\rangle \xrightarrow{a} \langle [\![\varsigma \mid s']\!], \sigma'\rangle$. Since $(r', s') \in R_{pq}$, according to the definition of R, $([\![\varsigma \mid r']\!], [\![\varsigma \mid s']\!]) \in R$.

Cond. $\langle\mathsf{str\text{-}tick}^{\Delta}\rangle$**:** Suppose $\langle [\![\varsigma \mid r]\!], \sigma\rangle \overset{\Delta}{\mapsto} \langle t, \sigma'\rangle$ for some $\sigma, \sigma' \in \Sigma$ and $t \in P$. By Rule $\langle\mathsf{scp\text{-}tick}\rangle$, as the final rule with this transition as conclusion, we obtain $\langle r, \varsigma \ll \sigma\rangle \overset{\Delta}{\mapsto} \langle r', \sigma''\rangle$, $t = [\![\varsigma \mid r']\!]$ and $\sigma' = \sigma \ll \sigma''_{/\mathrm{dom}(\sigma)\setminus\mathrm{dom}(\varsigma)}$. Since $(r, s) \in S_{pq}$, there exists an

$s' \in P$ such that $\langle s, \varsigma \ll \sigma\rangle \overset{\Delta}{\mapsto} \langle s', \sigma''\rangle$ and $(r', s') \in S_{pq}$. By Rule ⟨scp-tick⟩ again, $\langle [\![\varsigma \mid s]\!], \sigma\rangle \overset{\Delta}{\mapsto} \langle [\![\varsigma \mid s']\!], \sigma'\rangle$. From $(r, s) \in S_{pq}$ and the definition of S, $([\![\varsigma \mid r']\!], [\![\varsigma \mid s']\!]) \in S$.

We now show that R satisfies Conditions ⟨sl-term$^\Delta$⟩ and ⟨sl-tick$^\Delta$⟩. Let $([\![\varsigma \mid r]\!], [\![\varsigma \mid s]\!])$ be some pair from R.

Cond. ⟨sl-term$^\Delta$⟩**:** Suppose $\langle [\![\varsigma \mid r]\!], \sigma\rangle\downarrow$ for some $\sigma \in \Sigma$. Because Rule ⟨scp-term⟩ must be the final rule with this conclusion, we have $\langle r, \sigma \ll \varsigma\rangle\downarrow$. This implies that there is an $s' \in P$ such that $\langle s, \sigma \ll \varsigma\rangle \twoheadrightarrow \langle s', \sigma \ll \varsigma\rangle$, $\langle s', \sigma \ll \varsigma\rangle\downarrow$ and $(r, s') \in R_{pq}$. We also have that $\langle r, \sigma \ll \varsigma\rangle \overset{\Delta}{\not\mapsto}$ implies $\langle s', \sigma \ll \varsigma\rangle \overset{\Delta}{\not\mapsto}$. By Rules ⟨scp-term⟩ and ⟨scp-tran⟩, using that $(\sigma \ll \varsigma)_{/\mathrm{dom}(\varsigma)} = \varsigma$ and

$$\sigma \ll (\sigma \ll \varsigma)_{/(\mathrm{dom}(\sigma)\cup\mathrm{dom}(\varsigma))\setminus\mathrm{dom}(\varsigma)} = \sigma,$$

we have $\langle [\![\varsigma \mid s]\!], \sigma\rangle \twoheadrightarrow \langle [\![\varsigma \mid s']\!], \sigma\rangle$ and $\langle [\![\varsigma \mid s']\!], \sigma\rangle\downarrow$. Now, from $(r, s') \in R_{pq}$, by the definition of R, we have $([\![\varsigma \mid r]\!], [\![\varsigma \mid s']\!]) \in R$. Suppose now that $\langle [\![\varsigma \mid r]\!], \sigma\rangle \overset{\Delta}{\not\mapsto}$. By Rule ⟨scp-tick⟩, $\langle r, \sigma \ll \varsigma\rangle \overset{\Delta}{\not\mapsto}$. This implies $\langle s, \sigma \ll \varsigma\rangle \overset{\Delta}{\not\mapsto}$, and that, by Rule ⟨scp-tick⟩, implies $\langle [\![\varsigma \mid s]\!], \sigma\rangle \overset{\Delta}{\not\mapsto}$.

Cond. ⟨sl-tick$^\Delta$⟩**:** Suppose that $\langle [\![\varsigma \mid r]\!], \sigma\rangle \overset{\Delta}{\mapsto} \langle t, \sigma'\rangle$ for some $\sigma, \sigma' \in \Sigma$ and $t \in P$. By Rule ⟨scp-tick⟩, as the final rule with this transition as conclusion, we obtain $\langle r, \sigma \ll \varsigma\rangle \overset{\Delta}{\mapsto} \langle r', \sigma''\rangle$, $\sigma' = \sigma \ll \sigma''_{/\mathrm{dom}(\sigma)\setminus\mathrm{dom}(\varsigma)}$ and $t = [\![\varsigma \mid r']\!]$ for some $r' \in P$. Then there exist $s', s'' \in \Sigma$ such that $\langle s, \sigma \ll \varsigma\rangle \twoheadrightarrow \langle s'', \sigma \ll \varsigma\rangle \overset{\Delta}{\mapsto} \langle s', \sigma''\rangle$, $(r, s'') \in R_{pq}$ and $(r', s') \in R_{pq}$. We also have that $\langle r, \sigma \ll \varsigma\rangle\downarrow$ implies $\langle s'', \sigma \ll \varsigma\rangle\downarrow$ and $(r', s') \in S_{pq}$, and that $\langle r, \sigma \ll \varsigma\rangle\not\downarrow$ implies $\langle s'', \sigma \ll \varsigma\rangle\not\downarrow$. By Rules ⟨scp-tran⟩ and ⟨scp-tick⟩, using that $(\sigma \ll \varsigma)_{/\mathrm{dom}(\varsigma)} = \varsigma$ and $\sigma \ll (\sigma \ll \varsigma)_{/(\mathrm{dom}(\sigma)\cup\mathrm{dom}(\varsigma))\setminus\mathrm{dom}(\varsigma)} = \sigma$, we have $\langle [\![\varsigma \mid s]\!], \sigma\rangle \twoheadrightarrow \langle [\![\varsigma \mid s'']\!], \sigma\rangle \overset{\Delta}{\mapsto} \langle [\![\varsigma \mid s']\!], \sigma'\rangle$. Since $(r, s'') \in R_{pq}$ and $(r', s') \in R_{pq}$, by the definition of R, we have $(r, s'') \in R$ and $(r', s') \in R$. Suppose first that $\langle [\![\varsigma \mid r]\!], \sigma\rangle\downarrow$. By Rule ⟨scp-term⟩, we obtain $\langle r, \sigma \ll \varsigma\rangle\downarrow$. This implies $\langle s'', \sigma \ll \varsigma\rangle\downarrow$ and $(r', s') \in S_{pq}$. By Rule ⟨scp-term⟩, $\langle [\![\varsigma \mid s]\!], \sigma\rangle\downarrow$. According to the definition of S, we have $([\![\varsigma \mid r']\!], [\![\varsigma \mid s']\!]) \in S$. Suppose now that $\langle [\![\varsigma \mid r]\!], \sigma\rangle\not\downarrow$. By Rule ⟨scp-term⟩, $\langle r, \sigma \ll \varsigma\rangle\not\downarrow$. This implies $\langle s'', \sigma \ll \varsigma\rangle\not\downarrow$, which in turn, by Rule ⟨scp-term⟩, implies $\langle [\![\varsigma \mid s]\!], \sigma\rangle\not\downarrow$.

7. Define
$$R = \{(\partial_\Xi(r), \partial_\Xi(s)) \mid (r, s) \in R_{pq}\}$$
and
$$S = \{(\partial_\Xi(r), \partial_\Xi(s)) \mid (r, s) \in S_{pq}\}.$$

First we show that S is a relation of strong pairs in R. Suppose $(\partial_\Xi(r), \partial_\Xi(s)) \in S$.

Cond. $\langle \textsf{str-term}^\Delta \rangle$**:** Suppose $\langle \partial_\Xi(r), \sigma \rangle\downarrow$ for some $\sigma \in \Sigma$. Because Rule $\langle \textsf{enc-term} \rangle$ must be the final rule with this predicate as conclusion, we have $\langle r, \sigma \rangle\downarrow$. Since $(r, s) \in S_{pq}$, we have that $\langle s, \sigma \rangle\downarrow$, and hence by Rule $\langle \textsf{enc-term} \rangle$, that $\langle \partial_\Xi(q), \sigma \rangle\downarrow$.

Cond. $\langle \textsf{str-tran}^\Delta \rangle$**:** Suppose $\langle \partial_\Xi(r), \sigma \rangle \xrightarrow{a} \langle t, \sigma' \rangle$ for some $a \in A$ and $t \in P$. By Rule $\langle \textsf{enc-tran} \rangle$, as the final rule with this transition as conclusion, we obtain $\langle r, \sigma \rangle \xrightarrow{a} \langle r', \sigma' \rangle$, $t = \partial_\Xi(r')$ and $a \notin \Xi$. Since $(r, s) \in S_{pq}$, there exists $s' \in P$ such that $\langle s, \sigma \rangle \xrightarrow{a} \langle s', \sigma' \rangle$ and $(r', s') \in R_{pq}$. By Rule $\langle \textsf{enc-tran} \rangle$, $\langle \partial_\Xi(s), \sigma \rangle \xrightarrow{a} \langle \partial_\Xi(s'), \sigma' \rangle$. Since $(r', s') \in R_{pq}$, by the definition of R, $(\partial_\Xi(r'), \partial_\Xi(s')) \in R$.

Cond. $\langle \textsf{str-tick}^\Delta \rangle$**:** Suppose $\langle \partial_\Xi(r), \sigma \rangle \overset{\Delta}{\mapsto} \langle t, \sigma' \rangle$ for some $t \in P$. By Rule $\langle \textsf{enc-tick} \rangle$, as the final rule with this transition as conclusion, we obtain $\langle r, \sigma \rangle \overset{\Delta}{\mapsto} \langle r', \sigma' \rangle$ and $t = \partial_\Xi(r')$. Because $(r, s) \in S_{pq}$, there exists an $s' \in P$ such that $\langle s, \sigma \rangle \overset{\Delta}{\mapsto} \langle s', \sigma' \rangle$ and $(r', s') \in S_{pq}$. By Rule $\langle \textsf{enc-tick} \rangle$, $\langle \partial_\Xi(s), \sigma \rangle \overset{\Delta}{\mapsto} \langle \partial_\Xi(s'), \sigma' \rangle$. Since $(r', s') \in S_{pq}$, according to the definition of S, $(\partial_\Xi(r'), \partial_\Xi(s')) \in S$.

We now show that R satisfies Conditions $\langle \textsf{sl-term}^\Delta \rangle$ and $\langle \textsf{sl-tick}^\Delta \rangle$. Let $(\partial_\Xi(r), \partial_\Xi(s))$ be some pair from R.

Cond. $\langle \textsf{sl-term}^\Delta \rangle$**:** Suppose $\langle \partial_\Xi(r), \sigma \rangle\downarrow$. By Rule $\langle \textsf{enc-term} \rangle$, as the final rule with $\langle \partial_\Xi(r), \sigma \rangle\downarrow$ as conclusion, we obtain $\langle r, \sigma \rangle\downarrow$. It follows that there exist $s', s'' \in P$ such that $\langle s, \sigma \rangle \twoheadrightarrow \langle s', \sigma \rangle$, $\langle s', \sigma \rangle\downarrow$, $(r, s') \in R_{pq}$ and, if $\langle r, \sigma \rangle \overset{\Delta}{\not\mapsto}$, then also $\langle s', \sigma \rangle \overset{\Delta}{\not\mapsto}$. By Rules $\langle \textsf{enc-tran} \rangle$ and $\langle \textsf{enc-term} \rangle$, $\langle \partial_\Xi(s), \sigma \rangle \twoheadrightarrow \langle \partial_\Xi(s'), \sigma \rangle$ and $\langle \partial_\Xi(s'), \sigma \rangle\downarrow$. Suppose $\langle \partial_\Xi(r), \sigma \rangle \overset{\Delta}{\not\mapsto}$. Then by Rule $\langle \textsf{enc-tick} \rangle$, $\langle r, \sigma \rangle \overset{\Delta}{\not\mapsto}$. From this we have $\langle s', \sigma \rangle \overset{\Delta}{\not\mapsto}$ and then, by Rule $\langle \textsf{enc-tick} \rangle$ again, that $\langle \partial_\Xi(s'), \sigma \rangle \overset{\Delta}{\not\mapsto}$.

Cond. $\langle \mathsf{sl\text{-}tick}^{\Delta} \rangle$**:** Suppose now that $\langle \partial_\Xi(r), \sigma\rangle \overset{\Delta}{\mapsto} \langle t, \sigma'\rangle$ for some $t \in P$. By Rule $\langle \mathsf{enc\text{-}tick}\rangle$, $\langle r, \sigma\rangle \overset{\Delta}{\mapsto} \langle r', \sigma'\rangle$ and $t = \partial_\Xi(r')$ for some $r' \in P$. It follows that there exist $s', s'' \in P$ such that $\langle s, \sigma\rangle \twoheadrightarrow \langle s'', \sigma\rangle \overset{\Delta}{\mapsto} \langle s', \sigma'\rangle$, $(r, s'') \in R_{pq}$ and $(r', s') \in R_{pq}$. Also, if $\langle r, \sigma\rangle\downarrow$, then $\langle s'', \sigma\rangle\downarrow$ and $(r', s') \in S_{pq}$, and if $\langle r, \sigma\rangle\not\downarrow$, then $\langle s'', \sigma\rangle\not\downarrow$. By Rule $\langle \mathsf{enc\text{-}tran}\rangle$ and Rule $\langle \mathsf{enc\text{-}tick}\rangle$, $\langle \partial_\Xi(s), \sigma\rangle \twoheadrightarrow \langle \partial_\Xi(s''), \sigma\rangle \overset{\Delta}{\mapsto} \langle \partial_\Xi(s'), \sigma'\rangle$. Since $(r, s'') \in R_{pq}$ and $(r', s') \in R_{pq}$, by the definition of R, $(r, s'') \in R$ and $(r', s') \in R$. Now, suppose first that $\langle \partial_\Xi(r), \sigma\rangle\downarrow$. By Rule $\langle \mathsf{enc\text{-}term}\rangle$, we have $\langle r, \sigma\rangle\downarrow$. This implies that $\langle r, \sigma\rangle\downarrow$ and $(r', s') \in S_{pq}$. From $\langle r, \sigma\rangle\downarrow$, by Rule $\langle \mathsf{enc\text{-}term}\rangle$, we have $\langle \partial_\Xi(s), \sigma\rangle\downarrow$. From $(r', s') \in S_{pq}$, according to the definition of S, we obtain $(\partial_\Xi(r'), \partial_\Xi(s')) \in S$. Suppose that $\langle \partial_\Xi(r), \sigma\rangle\not\downarrow$. It easily follows that then $\langle r, \sigma\rangle\not\downarrow$. This implies $\langle s, \sigma\rangle\not\downarrow$, and that $\langle \partial_\Xi(s), \sigma\rangle\not\downarrow$.

8. Define

$$R = \{(\tau_I(r), \tau_I(s)) \mid (r, s) \in R_{pq}\}$$

and

$$S = \{(\tau_I(r), \tau_I(s)) \mid (r, s) \in S_{pq}\}.$$

First we show that S is a relation of strong pairs in R. Suppose $(\tau_I(r), \tau_I(s)) \in S$.

Cond. $\langle \mathsf{str\text{-}term}^{\Delta}\rangle$**:** Suppose $\langle \tau_I(r), \sigma\rangle\downarrow$ for some $\sigma \in \Sigma$. Because Rule $\langle \mathsf{hide\text{-}term}\rangle$ must be the final rule with this as conclusion, it follows that $\langle r, \sigma\rangle\downarrow$. Since $(r, s) \in S_{pq}$, we have $\langle s, \sigma\rangle\downarrow$, and hence by Rule $\langle \mathsf{hide\text{-}term}\rangle$, $\langle \tau_I(q), \sigma\rangle\downarrow$.

Cond. $\langle \mathsf{str\text{-}tran}^{\Delta}\rangle$**:** Suppose $\langle \tau_I(r), \sigma\rangle \xrightarrow{a} \langle t, \sigma'\rangle$ for some $a \in A$ and $t \in P$. Since Rules $\langle \mathsf{hide\text{-}tran}_1\rangle$ and $\langle \mathsf{hide\text{-}tran}_2\rangle$ are the final rules with this transition as conclusion, we have $\langle r, \sigma\rangle \xrightarrow{b} \langle r', \sigma'\rangle$ and $t = \tau_I(r')$, for some $b \in \mathsf{A}$ such that either $b \in I$ and $a = \tau$ or $b \notin I$ and $a = b$. Since $(r, s) \in S_{pq}$, there exists $s' \in P$ such that $\langle s, \sigma\rangle \xrightarrow{b} \langle s', \sigma'\rangle$ and $(r', s') \in R_{pq}$. By Rules $\langle \mathsf{hide\text{-}tran}_1\rangle$ and $\langle \mathsf{hide\text{-}tran}_2\rangle$, $\langle \tau_I(s), \sigma\rangle \xrightarrow{a} \langle \tau_I(s'), \sigma'\rangle$. Since $(r', s') \in R_{pq}$, according to the definition of R, we have $(\tau_I(r'), \tau_I(s')) \in R$.

Cond. $\langle \mathsf{str\text{-}tick}^{\Delta}\rangle$**:** Suppose $\langle \tau_I(r), \sigma\rangle \overset{\Delta}{\mapsto} \langle t, \sigma'\rangle$ for some $t \in P$. By Rule $\langle \mathsf{hide\text{-}tick}\rangle$, as the final rule with this transition as conclusion, we obtain $\langle r, \sigma\rangle \overset{\Delta}{\mapsto} \langle r', \sigma'\rangle$ and $t = \tau_I(r')$. Because

$(r, s) \in S_{pq}$, there exists an $s' \in P$ such that $\langle s, \sigma \rangle \overset{\Delta}{\mapsto} \langle s', \sigma' \rangle$ and $(r', s') \in S_{pq}$. By Rule $\langle \mathsf{hide\text{-}tick} \rangle$, $\langle \tau_I(s), \sigma \rangle \overset{\Delta}{\mapsto} \langle \tau_I(s'), \sigma' \rangle$. Since $(r', s') \in S_{pq}$, according to the definition of S, $(\tau_I(r'), \tau_I(s')) \in S$.

We now show that R satisfies Conditions $\langle \mathsf{sl\text{-}term}^{\Delta} \rangle$ and $\langle \mathsf{sl\text{-}tick}^{\Delta} \rangle$. Let $(\tau_I(r), \tau_I(s))$ be some pair from R.

Cond. $\langle \mathsf{sl\text{-}term}^{\Delta} \rangle$**:** Suppose $\langle \tau_I(r), \sigma \rangle\!\downarrow$. By Rule $\langle \mathsf{hide\text{-}term} \rangle$, as the final rule with $\langle \tau_I(r), \sigma \rangle\!\downarrow$ as conclusion, we obtain $\langle r, \sigma \rangle\!\downarrow$. From $(r, s) \in R_{pq}$ it follows that there exist $s', s'' \in P$ such that $\langle s, \sigma \rangle \twoheadrightarrow \langle s', \sigma \rangle$, $\langle s', \sigma \rangle\!\downarrow$, $(r, s') \in R_{pq}$ and, if $\langle r, \sigma \rangle \overset{\Delta}{\not\mapsto}$, then also $\langle s', \sigma \rangle \overset{\Delta}{\not\mapsto}$. By Rules $\langle \mathsf{hide\text{-}tran}_1 \rangle$, $\langle \mathsf{hide\text{-}tran}_2 \rangle$ and $\langle \mathsf{hide\text{-}term} \rangle$, we have that $\langle \tau_I(s), \sigma \rangle \twoheadrightarrow \langle \tau_I(s'), \sigma \rangle$ and $\langle \tau_I(s'), \sigma \rangle\!\downarrow$. Suppose $\langle \tau_I(r), \sigma \rangle \overset{\Delta}{\not\mapsto}$. Then by Rule $\langle \mathsf{hide\text{-}tick} \rangle$, $\langle r, \sigma \rangle \overset{\Delta}{\not\mapsto}$. From this we have $\langle s', \sigma \rangle \overset{\Delta}{\not\mapsto}$ and then, by Rule $\langle \mathsf{hide\text{-}tick} \rangle$ again, that $\langle \tau_I(s'), \sigma \rangle \overset{\Delta}{\not\mapsto}$.

Cond. $\langle \mathsf{sl\text{-}tick}^{\Delta} \rangle$**:** Suppose now that $\langle \tau_I(r), \sigma \rangle \overset{\Delta}{\mapsto} \langle t, \sigma' \rangle$ for some $t \in P$. By Rule $\langle \mathsf{hide\text{-}tick} \rangle$, $\langle r, \sigma \rangle \overset{\Delta}{\mapsto} \langle r', \sigma' \rangle$ and $t = \tau_I(r')$ for some $r' \in P$. It follows that there exist $s', s'' \in P$ such that $\langle s, \sigma \rangle \twoheadrightarrow \langle s'', \sigma \rangle \overset{\Delta}{\mapsto} \langle s', \sigma' \rangle$, $(r, s'') \in R_{pq}$ and $(r', s') \in R_{pq}$. Also, if $\langle r, \sigma \rangle\!\downarrow$, then $\langle s'', \sigma \rangle\!\downarrow$ and $(r', s') \in S_{pq}$, and if $\langle r, \sigma \rangle\!\not\downarrow$, then $\langle s'', \sigma \rangle\!\not\downarrow$. By Rules $\langle \mathsf{hide\text{-}tran}_1 \rangle$, $\langle \mathsf{hide\text{-}tran}_2 \rangle$ and $\langle \mathsf{hide\text{-}tick} \rangle$, we have $\langle \tau_I(s), \sigma \rangle \twoheadrightarrow \langle \tau_I(s''), \sigma \rangle \overset{\Delta}{\mapsto} \langle \tau_I(s'), \sigma' \rangle$. Since $(r, s'') \in R_{pq}$ and $(r', s') \in R_{pq}$, by the definition of R, we have $(r, s'') \in R$ and $(r', s') \in R$. Suppose first that $\langle \tau_I(r), \sigma \rangle\!\downarrow$. By Rule $\langle \mathsf{hide\text{-}term} \rangle$, we have $\langle r, \sigma \rangle\!\downarrow$. This implies that $\langle r, \sigma \rangle\!\downarrow$ and that $(r', s') \in S_{pq}$. From $\langle r, \sigma \rangle\!\downarrow$, by Rule $\langle \mathsf{hide\text{-}term} \rangle$, we have $\langle \tau_I(s), \sigma \rangle\!\downarrow$. Since $(r', s') \in S_{pq}$, by the definition of S, $(\tau_I(r'), \tau_I(s')) \in S$. Suppose that $\langle \tau_I(r), \sigma \rangle\!\not\downarrow$. It easily follows that then $\langle r, \sigma \rangle\!\not\downarrow$. This implies $\langle s, \sigma \rangle\!\not\downarrow$, and that $\langle \tau_I(s), \sigma \rangle\!\not\downarrow$. ■

5.2 Adding maximal progress

Maximal progress in a system is the property that events must happen as soon as they are enabled. In a model this means that action execution has priority over the passage of time. The property is usually enforced by the maximal progress operator π_M where the parameter set $M \subseteq \mathsf{A}$ is the set of actions that are given priority over delaying. In this section we introduce a variant of the maximal progress operator and show that $\approx_s^\Delta$ is a congruence for it.

The standard semantics of the operator π_M says that $\pi_M(p)$ ticks only if p ticks and p cannot execute an action from M. In the setting with termination it is additionally required that $\pi_M(p)$ terminates iff p terminates. The first condition is to weak to be used in our setting for two different reasons as illustrated by the following example.

Example 5.2.1 Assume that $\mathsf{eff}(\boldsymbol{\tau}, \sigma) = \{\sigma\}$ for all $\sigma \in \Sigma$.

a. We have $\Delta 1 + \varepsilon \approx_s^\Delta \Delta 1 + \boldsymbol{\tau} \cdot (\Delta 1 + \varepsilon) + \varepsilon$. However, putting the maximal progress operator around these terms, with $\tau \in M$ and with the semantics from above, the left-hand side process behaves the same but the right-hand side process looses the option to delay initially. This gives us the situation from Example 5.1.1a and so, the two processes are not timed silently congruent.

b. Clearly $\boldsymbol{a} + \Delta 1 \approx_s^\Delta \Delta 1 + \boldsymbol{\tau} \cdot (\boldsymbol{a} + \Delta 1)$. By putting the maximal progress operator π_M, with $a \in M$ and $\tau \notin M$, around we get that the left-hand side process cannot delay while the right-hand side process can. $\square$

The problem in the first case is again due to the inseparability of termination and delaying. The solution is to forbid the maximal progress operator to cut the tick transitions whenever the process can also terminate. In the second case, the problem is that the execution of a is postponed by the silent step. Since this would always impose problems, the only solution is to require that M always contains the internal action τ.

Remark 5.2.2 It is, in general, possible to keep the standard semantics of the maximal progress operator and to adapt timed silent congruence so that it becomes compositional with respect to it. However, to solve the first problem these adaptations are very complex, and for the second problem they lead to a much stronger equivalence (see the definition of orthogonal bisimulation from [16]). That is why we have decided to change the definition of maximal progress itself. We believe that our decision does not have a big practical disadvantage. $\square$

The operational rules for the maximal progress operator are now given in Table 5.2. We always assume that $M \subseteq \mathsf{A}$ and $\tau \in M$.

Note that, so far, the introduction of timing to the theory was independent of divergence. For the maximal progress operator however, it is crucial

$$\frac{\langle p,\sigma\rangle\downarrow}{\langle \pi_M(p),\sigma\rangle\downarrow}\ \langle\mathsf{mp\text{-}term}\rangle \qquad \frac{\langle p,\sigma\rangle \xrightarrow{a} \langle p',\sigma'\rangle}{\langle \pi_M(p),\sigma\rangle \xrightarrow{a} \langle \pi_M(p'),\sigma'\rangle}\ \langle\mathsf{mp\text{-}tran}\rangle$$

$$\frac{\langle p,\sigma\rangle \overset{\Delta}{\mapsto} \langle p',\sigma'\rangle,\ \langle p,\sigma\rangle\downarrow}{\langle \pi_M(p),\sigma\rangle \overset{\Delta}{\mapsto} \langle \pi_M(p'),\sigma'\rangle}\ \langle\mathsf{mp\text{-}tick_1}\rangle$$

$$\frac{\langle p,\sigma\rangle \overset{\Delta}{\mapsto} \langle p',\sigma'\rangle,\ \langle p,\sigma\rangle\not\downarrow,\ \langle p,\sigma\rangle \not\xrightarrow{a} \text{ for } a\in M}{\langle \pi_M(p),\sigma\rangle \overset{\Delta}{\mapsto} \langle \pi_M(p'),\sigma'\rangle}\ \langle\mathsf{mp\text{-}tick_2}\rangle$$

Table 5.3: Operational semantics for the maximal progress operator

that the divergence condition is imposed if $\approx_s^\Delta$ is to be a congruence. If we ignore divergence, then $\Delta 1$ becomes timed silent congruent to $\tau^* \cdot \Delta 1$, but $\pi_M(\Delta 1)$ and $\pi_M(\tau^* \cdot \Delta 1)$ cannot be timed silent congruent. This is because $\pi_M(\Delta 1)$ can still do a tick while $\pi_M(\tau^* \cdot \Delta 1)$ cannot.

We now prove that $\approx_s^\Delta$ is a congruence for maximal progress.

Theorem 5.2.3 For all $p, q \in P$, if $p \approx_s^\Delta q$, then $\pi_M(p) \approx_s^\Delta \pi_M(q)$ for any $M \subseteq \mathsf{A}$ such that $\tau \in M$. □

Proof Let $p \approx_s^\Delta q$ be witnessed by the timed stateless silent bisimulation (S_{pq}, R_{pq}). Define

$$S = \{(\pi_M(r), \pi_M(s) \mid (r,s) \in S_{pq}\} \text{ and } R = \{(\pi_M(r), \pi_M(s) \mid (r,s) \in R_{pq}\}.$$

We show that (S, R) is a timed stateless silent bisimulation. First, we show that S is a relation of strong pairs in R.

Cond. $\langle\mathsf{str\text{-}term}^\Delta\rangle$**:** Suppose $\langle \pi_M(r),\sigma\rangle\downarrow$ for some $\sigma \in \Sigma$. Rule $\langle\mathsf{mp\text{-}term}\rangle$ must be the final rule with this as conclusion, so it follows that $\langle r,\sigma\rangle\downarrow$. Since $(r,s) \in S_{pq}$, we have $\langle s,\sigma\rangle\downarrow$, and hence, by Rule $\langle\mathsf{mp\text{-}term}\rangle$, we have $\langle \pi_M(q),\sigma\rangle\downarrow$.

Cond. $\langle\mathsf{str\text{-}tran}^\Delta\rangle$**:** Suppose $\langle \pi_M(r),\sigma\rangle \xrightarrow{a} \langle t,\sigma'\rangle$ for some $a \in A$ and $t \in P$. By Rule $\langle\mathsf{mp\text{-}tran}\rangle$, as the final rule with this transition as conclusion, we obtain $\langle r,\sigma\rangle \xrightarrow{a} \langle r',\sigma'\rangle$ and $t = \pi_M(r')$. Since $(r,s) \in S_{pq}$, there exists $s' \in P$ such that $\langle s,\sigma\rangle \xrightarrow{a} \langle s',\sigma'\rangle$ and $(r',s') \in R_{pq}$. By Rule $\langle\mathsf{mp\text{-}tran}\rangle$, $\langle \pi_M(s),\sigma\rangle \xrightarrow{a} \langle \pi_M(s'),\sigma'\rangle$. Since $(r',s') \in R_{pq}$, according to the definition of R, $(\pi_M(r'), \pi_M(s')) \in R$.

Cond. $\langle \mathsf{str\text{-}tick}^{\Delta} \rangle$**:** Suppose $\langle \pi_M(r), \sigma \rangle \overset{\Delta}{\mapsto} \langle t, \sigma' \rangle$ for some $t \in P$. The final rule in a derivation with this as conclusion is either Rule $\langle \mathsf{mp\text{-}tick}_1 \rangle$ or Rule $\langle \mathsf{mp\text{-}tick}_2 \rangle$.

If Rule $\langle \mathsf{mp\text{-}tick}_1 \rangle$ is the final rule applied, we have $\langle r, \sigma \rangle\downarrow$ and $\langle r, \sigma \rangle \overset{\Delta}{\mapsto} \langle r', \sigma' \rangle$ for some $r' \in P$, and $t = \pi_M(r')$. Since $(r, s) \in S_{pq}$, we have that $\langle s, \sigma \rangle\downarrow$ and that there exists an $s' \in P$ such that $\langle s, \sigma \rangle \overset{\Delta}{\mapsto} \langle s', \sigma' \rangle$ and $(r', s') \in S_{pq}$. By Rule $\langle \mathsf{mp\text{-}tick}_1 \rangle$, $\langle \pi_M(s), \sigma \rangle \overset{\Delta}{\mapsto} \langle \pi_M(s'), \sigma' \rangle$. According to S, from $(r', s') \in S_{pq}$ we obtain $(\pi_M(r'), \pi_M(s')) \in S$.

If Rule $\langle \mathsf{mp\text{-}tick}_2 \rangle$ is the final rule applied, then we have $\langle r, \sigma \rangle \not\downarrow$, $\langle r, \sigma \rangle \overset{a}{\not\to}$, $\langle r, \sigma \rangle \overset{\Delta}{\mapsto} \langle r', \sigma' \rangle$ and $t = \pi_M(r')$, for some $r' \in P$ and $a \in M$. Since $(r, s) \in S_{pq}$, we have that $\langle s, \sigma \rangle \not\downarrow$ and that there exists an $s' \in P$ such that $\langle s, \sigma \rangle \overset{\Delta}{\mapsto} \langle s', \sigma' \rangle$ and $(r', s') \in S_{pq}$. In addition, $\langle s, \sigma \rangle \overset{a}{\not\to}$. By Rule $\langle \mathsf{mp\text{-}tick}_1 \rangle$, $\langle \pi_M(s), \sigma \rangle \overset{\Delta}{\mapsto} \langle \pi_M(s'), \sigma' \rangle$. According to S, from $(r', s') \in S_{pq}$ we obtain $(\pi_M(r'), \pi_M(s')) \in S$.

We now show that R satisfies the conditions of Definition 5.1.4. Let $(\pi_M(r), \pi_M(s))$ be some pair from R.

Cond. $\langle \mathsf{sl\text{-}term}^{\Delta} \rangle$**:** Suppose $\langle \pi_M(r), \sigma \rangle\downarrow$. Since Rule $\langle \mathsf{mp\text{-}term} \rangle$ is the final rule with $\langle \pi_M(r), \sigma \rangle\downarrow$ as conclusion, we obtain $\langle r, \sigma \rangle\downarrow$. It follows that there exist $s', s'' \in P$ such that $\langle s, \sigma \rangle \twoheadrightarrow \langle s', \sigma \rangle$, $\langle s', \sigma \rangle\downarrow$, $(r, s') \in R_{pq}$ and, if $\langle r, \sigma \rangle \overset{\Delta}{\not\mapsto}$, then also $\langle s', \sigma \rangle \overset{\Delta}{\not\mapsto}$. By Rules $\langle \mathsf{mp\text{-}tran} \rangle$ and $\langle \mathsf{mp\text{-}term} \rangle$, $\langle \pi_M(s), \sigma \rangle \twoheadrightarrow \langle \pi_M(s'), \sigma \rangle$ and $\langle \pi_M(s'), \sigma \rangle\downarrow$. Suppose $\langle \pi_M(r), \sigma \rangle \overset{\Delta}{\not\mapsto}$. Then, because $\langle r, \sigma \rangle\downarrow$, by Rule $\langle \mathsf{mp\text{-}tick}_1 \rangle$ we obtain that $\langle r, \sigma \rangle \overset{\Delta}{\not\mapsto}$. From this we have $\langle s', \sigma \rangle \overset{\Delta}{\not\mapsto}$ and then, by Rules $\langle \mathsf{mp\text{-}tick}_1 \rangle$ and $\langle \mathsf{mp\text{-}tick}_1 \rangle$, that $\langle \pi_M(s'), \sigma \rangle \overset{\Delta}{\not\mapsto}$.

Cond. $\langle \mathsf{sl\text{-}tran} \rangle$**:** Suppose $\langle \pi_M(r), \sigma \rangle \overset{a}{\to} \langle t, \sigma' \rangle$ for some $a \in A$ and $t \in P$. Rule $\langle \mathsf{mp\text{-}tran} \rangle$ is the final rule with this transition as conclusion, so we obtain $\langle r, \sigma \rangle \overset{a}{\to} \langle r', \sigma' \rangle$ and $t = \pi_M(r')$. Since R_{pq} satisfies $\langle \mathsf{sl\text{-}tran} \rangle$, there exist $s', s'' \in P$ such that $\langle s, \sigma \rangle \twoheadrightarrow \langle s'', \sigma \rangle \xrightarrow{(a)} \langle s', \sigma \rangle$, $(\langle r, \sigma \rangle, \langle s'', \sigma \rangle) \in R_{pq}$ and $(\langle r', \sigma' \rangle, \langle s', \sigma' \rangle) \in R_{pq}$. Now, by Rule $\langle \mathsf{mp\text{-}tran} \rangle$, $\langle \pi(s_0), \sigma \rangle \twoheadrightarrow \langle \pi(s''), \sigma \rangle \xrightarrow{(a)} \langle \pi(s'), \sigma' \rangle$. Since $(r, s'') \in R_{pq}$ and $(r', \langle s', \sigma' \rangle) \in R_{pq}$, by the definition of R, we have $(\pi(r), \pi(s'')) \in R$ and $(\pi(r'), \pi(s')) \in R$.

Cond. $\langle \mathsf{sl\text{-}div} \rangle$**:** Suppose that there exist $t_0, t_1, t_2, \ldots \in P$, such that $t_0 = \pi(r)$, $\langle t_0, \sigma \rangle \to \langle t_1, \sigma \rangle \to \langle t_2, \sigma \rangle \to \cdots$ and that $(t_i, \pi(s)) \in R$ for all

$i \geqslant 0$. According to the definition of R, there exist $r_0, r_1, r_2, \ldots \in P$ such that $t_i = \pi(r_i)$ for all $i \geqslant 0$. Since Rule $\langle\mathsf{mp\text{-}tran}\rangle$ is the only rule that has $\langle \pi(r_i), \sigma\rangle \rightarrow \langle \pi(r_{i+1}), \sigma\rangle$ as conclusion, we have $\langle r_0, \sigma\rangle \rightarrow \langle r_1, \sigma\rangle \rightarrow \langle r_2, \sigma\rangle \rightarrow \cdots$ and $(t_i, \pi(s)) \in R_{pq}$. Since R_{pq} satisfies $\langle\mathsf{div}\rangle$, there exist $s' \in P$ and $k \geqslant 0$ such that $\langle s, \sigma\rangle \rightarrow^{+} \langle s', \sigma\rangle$ and $(r_k, s') \in R_{pq}$. By Rule $\langle\mathsf{mp\text{-}tran}\rangle$, $\langle \pi(s), \sigma\rangle \rightarrow^{+} \langle \pi(s'), \sigma\rangle$. Since $(r_k, s') \in R_{pq}$, according to the definition of R, $(\pi(r), \pi(s'')) \in R$.

Cond. $\langle\mathsf{sl\text{-}tick}^{\Delta}\rangle$**:** Suppose now that $\langle \pi_M(r), \sigma\rangle \overset{\Delta}{\mapsto} \langle t, \sigma'\rangle$ for some $t \in P$. The final rule in a derivation with this as conclusion must be either Rule $\langle\mathsf{mp\text{-}tick}_1\rangle$ or Rule $\langle\mathsf{mp\text{-}tick}_2\rangle$.

If Rule $\langle\mathsf{mp\text{-}tick}_1\rangle$ is the final rule applied, we have $\langle r, \sigma\rangle\downarrow$ and $\langle r, \sigma\rangle \overset{\Delta}{\mapsto} \langle r', \sigma'\rangle$ for some $r' \in P$, and $t = \pi_M(r')$. It follows that there exist $s', s'' \in P$ such that $\langle s, \sigma\rangle \twoheadrightarrow \langle s'', \sigma\rangle \overset{\Delta}{\mapsto} \langle s', \sigma'\rangle$, $\langle s'', \sigma\rangle\downarrow$, $(r, s'') \in R_{pq}$ and $(r', s') \in R_{pq}$. By Rules $\langle\mathsf{mp\text{-}tran}\rangle$ and $\langle\mathsf{mp\text{-}tick}_1\rangle$, $\langle \pi_M(s), \sigma\rangle \twoheadrightarrow \langle \pi_M(s''), \sigma\rangle \overset{\Delta}{\mapsto} \langle \pi_M(s'), \sigma'\rangle$. Since $(r, s'') \in R_{pq}$ and $(r', s') \in R_{pq}$, according to the definition of R, $(r, s'') \in R$ and $(r', s') \in R$. Since $\langle r, \sigma\rangle\downarrow$ and $\langle s'', \sigma\rangle\downarrow$, by Rule $\langle\mathsf{mp\text{-}term}\rangle$, we have $\langle \pi_M(r), \sigma\rangle\downarrow$ and $\langle \pi_M(s''), \sigma\rangle\downarrow$.

If Rule $\langle\mathsf{mp\text{-}tick}_2\rangle$ is the final rule applied, then $\langle r, \sigma\rangle\not\downarrow$, $\langle r, \sigma\rangle \overset{a}{\not\rightarrow}$ and $\langle r, \sigma\rangle \overset{\Delta}{\mapsto} \langle r', \sigma'\rangle$ for some $r' \in P$ and $a \in M$, and $t = \pi_M(r')$. It easily follows (see the proof of Lemma 2.6.2 - deadlock preservation; this is the place where the divergence condition is used) that there exists an $s'' \in P$ such that $\langle s, \sigma\rangle \twoheadrightarrow \langle s'', \sigma\rangle$, $\langle s'', \sigma\rangle \overset{a}{\not\rightarrow}$, $\langle s'', \sigma\rangle\not\downarrow$ and $(r, s'') \in R_{pq}$. It follows that there exists an $s' \in P$ such that $\langle s'', \sigma\rangle \overset{\Delta}{\mapsto} \langle s', \sigma'\rangle$ and $(r', s') \in R_{pq}$. By Rule $\langle\mathsf{mp\text{-}tran}\rangle$ and Rule $\langle\mathsf{mp\text{-}tick}_2\rangle$, $\langle \pi_M(s), \sigma\rangle \twoheadrightarrow \langle \pi_M(s''), \sigma\rangle \overset{\Delta}{\mapsto} \langle \pi_M(s'), \sigma'\rangle$. Since $\langle r, \sigma\rangle\not\downarrow$ and $\langle s'', \sigma\rangle\not\downarrow$, by Rule $\langle\mathsf{mp\text{-}term}\rangle$, we have $\langle \pi_M(r), \sigma\rangle\not\downarrow$ and $\langle \pi_M(s''), \sigma\rangle\not\downarrow$. ■

Chapter 6

Application: Translating χ to Promela

The language χ [90] is a modeling language developed to detect design flaws and to optimize performance of industrial systems (machines, manufacturing lines, warehouses, factories, etc.) It allows for the specification of discrete-event, continuous and probabilistic aspects of systems. Its simulator has been successfully applied to a large number of industrial cases, such as a car assembly line (NedCar [54]), a multi-product, multi-process wafer fab (Philips [26]), a brewery (Heineken), a fruit juice blending and packaging plant (Riedel [46]) and process industry plants ([12]). Simulation is a powerful technique for performance analysis, like calculating throughput and cycle time, but it is less suitable for verification.

To facilitate verification, either verification tools have to be developed especially for χ, or existing verification tools and techniques have to be made available for use with χ. Currently, the latter approach is pursued [19, 18, 90, 95, 99]. The idea is to extend χ with facilities for model checking by establishing a connection with other state-of-the-art verification tools and techniques on the level of the specification language. That is, formal verification of a χ model is done by first translating it into the input language of some model checker and then performing the actual verification there. Preferably, the translation closely resembles the original, so that counterexamples produced by the model checker can be related to the original specification. It should also use, as much as possible, the features of the target language to ensure that the full power of the verification tool is used.

In [95] the translation of (a subset of) χ specifications into Promela, the input language of the popular model checker Spin [65], is discussed and presented in detail. The translation process explained there proceeds in two phases. The first phase, which is called the *preprocessing phase*, consists of a transformation of the χ model in an attempt to eliminate all constructs that do not directly map to PROMELA constructs. For instance, χ has an explicit construct for parallel composition which facilitates nested parallelism, whereas PROMELA only allows implicit parallel composition of process definitions; so in the preprocessing phase the nested parallelism in the χ model is eliminated. If the result after the preprocessing phase is a χ model that only has constructions with a direct translation into PROMELA, then it can be translated to a PROMELA model; this phase is called the *translation phase*.

The main difficulty for establishing the correctness of the whole translation is that the two languages do not have a formal semantics in common. An advantage of the two-phase approach sketched above is that the preprocessing phase of the translation, which is usually the most involved part, takes place entirely within the realm of χ. Therefore, a correctness proof for this phase only involves the formal semantics of χ. An additional advantage of the two-phase approach is that the preprocessing phase (and its correctness proof) is potentially reusable, e.g., when defining a translation from χ to some other language.

In this chapter we present how silent congruence can be interpreted in the χ setting and how it can be used to prove the correctness of the preprocessing phase from [95].

6.1 The language χ

The variant of χ that we use is the subset of the language that is used for the modeling of discrete-event aspects of systems only. We also do not consider data types.

The language κ is based on χ so it is not surprising that the two languages have many features in common. There are, however, also some differences. First, the timing model in χ is dense, and not discrete like in κ. Next, χ has the explicit notion of assignments. Also, χ has communication features based on the process algebra CSP, i.e. the communication goes via channels by send and receive statements, while in κ they are based on the process algebra ACP. Finally, χ has no notion of explicit (successful) termination (although the previous version of χ, called χ_σ, on which κ is based, had successful termination). Other differences also appear in how repetition is handled.

Let us now give the syntax and semantics of χ. We start with atomic processes. The process δ is the same as in κ, and the process skip corresponds to $\boldsymbol{\tau}$ (if τ does not change the state). The *delay* process Δe delays the number of ticks that is equal to the value of the expression e; the process $\Delta 0$ is equivalent to skip. The *(multi)assignment* process $x_1, \ldots, x_n := e_1, \ldots, e_n$ assigns the value of the expression e_i to the variable x_i, $1 \leqslant i \leqslant n$. It does not have the possibility to delay. The *send* process $m!!e$ sends the value of the expression e along the channel m and cannot delay. The *delayable send* $m!e$ behaves as $m!!e$ but it can delay arbitrarily long. The *receive* process $m??x$ inputs a value over the channel m and assigns it to x. It cannot delay. The *delayable receive* $m?x$ is the same as $m??x$ but can delay.

From the compound processes, χ has guards ($\mathsf{b} :\rightarrow p$), sequential (;) and alternative ($[\!]$) composition, scopes ($[\![s \mid p]\!]$), encapsulation ($\partial_{\Xi}(p)$), and hiding ($\tau_I(p)$). They all have the same semantics as in κ. There are also other operators. The *repetition* operator $*p$ behaves as p infinitely many times. The *guarded repetition* process $\mathsf{b} \overset{*}{\rightarrow} p$ is interpreted as 'while b do ($\mathsf{skip}; p$)'. Note that this makes $\mathit{true} \overset{*}{\rightarrow} p$ fundamentally different from $*p$; the executability of $\mathit{true} \overset{*}{\rightarrow} p$ does not depend on the executability of p. The *parallel composition* operator $\parallel$ executes p and q concurrently in an interleaved fashion like in κ. In addition, if one of the processes can execute a send action and the other one can execute a receive action on the same channel, then they can also communicate, i.e. $p \parallel q$ can also execute the communication action on this channel. So χ has a special communication function. The *urgent communication* operator $\mathcal{U}_{\mathcal{H}}$ gives communication actions via channels from $\mathcal{H}$ a higher priority over the passage of time.

6.2 Embedding χ into Timed κ

In this section we explain how χ can be interpreted in the Timed κ setting. We presuppose a set M of channel names, a set V of variables, a set D of data values, and a set E of data expressions that includes V and D. Since we are working with the untyped subset of χ, we can, without loss of generality, assume that D is the set of natural numbers.

We first define the set of actions associated with χ, and then we define the functions eff and comm. Next, we define the set of action execution processes and the function act. Finally, we define the set of atomic propositions and the function check.

The set of actions that a χ process can perform contains the internal action, assignment actions, send and receive actions, and communication actions. Formally:

$$\begin{aligned}\mathsf{A} \;=\; & \{\tau\} \\ & \cup \{\mathsf{asgn}[(x_1,d_1),\ldots,(x_n,d_n)] \mid x_i \in V, d_i \in D\} \\ & \cup \{\mathsf{snd}(m,d) \mid m \in M, d \in D\} \\ & \cup \{\mathsf{rcv}(m,x,d) \mid m \in M, d \in D, x \in V\} \\ & \cup \{\mathsf{comm}(m,x,d) \mid m \in M, d \in D, x \in V\}.\end{aligned}$$

The communication mechanism of χ is achieved by defining the communication function comm by

$$\mathsf{comm}(\mathsf{snd}(m,d),\mathsf{rcv}(m,x,d)) = \mathsf{comm}(m,x,d).$$

The set of action execution processes contains the multi assignments, the skip process, and the undelayable send and receive processes. Formally:

$$\begin{aligned}\mathsf{Act} \;=\; & \{x_1,\ldots,x_n := e_1,\ldots,e_n \mid x_i \in V, e_i \in E, 1 \leqslant i \leqslant n\} \\ & \cup \{\mathsf{skip}\} \\ & \cup \{m!!e \mid m \in M, e \in E\} \\ & \cup \{m??x \mid m \in M, x \in V\}.\end{aligned}$$

We define the function act. For the assignment process we let

$$\mathsf{act}(x_1,\ldots,x_n := e_1,\ldots,e_n,\sigma) = \{\mathsf{asgn}[(x_1,d_1),\ldots,(x_n,d_n)]\},$$

when $\sigma(e_i) = d_i$ for all $1 \leqslant i \leqslant n$, and let $\mathsf{act}(x_1,\ldots,x_n := e_1,\ldots,e_n,\sigma)$ be the empty set otherwise. The process skip in χ corresponds to the process $\boldsymbol{\tau}$ in κ, and so $\mathsf{act}(\mathsf{skip},\sigma) = \{\tau\}$ for all $\sigma \in \Sigma$. For the undelayable send process the observable actions are defined by: $\mathsf{act}(m!!e,\sigma) = \{\mathsf{snd}(m,d)\}$ if $\sigma(e) = d$, and $\mathsf{act}(m!!e,\sigma) = \emptyset$ if $\sigma(e)$ is undefined. For the undelayable receive we have $\mathsf{act}(m??x,\sigma) = \{\mathsf{rcv}(m,x,d) \mid d \in D\}$. The receive process in χ has an option to receive any value.

Assume Σ to be the set of valuations, i.e. of partial functions that assign values to variables and expressions. We now define the effects of χ actions on valuations. The τ action and the send action have no effect on the valuation. Formally, for all $\sigma \in \Sigma$,

$$\mathsf{eff}(\tau,\sigma) = \mathsf{eff}(\mathsf{snd}(m,d),\sigma) = \{\sigma\}.$$

For the assignment action $\mathsf{asgn}[(x_1,d_1),\ldots,(x_n,d_n)]$ we let

$$\mathsf{eff}(\mathsf{asgn}[(x_1,d_1),\ldots,(x_n,d_n)],\sigma) = \{\sigma \ll \{x_1 \mapsto d_1,\ldots,x_n \mapsto d_n\}\}.$$

The receive action $\mathsf{rcv}(m, x, d)$ and the communication action $\mathsf{comm}(m, x, d)$ both assign d to x and so, for all $\sigma \in \Sigma$, we have

$$\mathsf{eff}(\mathsf{rcv}(m, x, d), \sigma) = \mathsf{eff}(\mathsf{comm}(m, x, d), \sigma) = \{\sigma \ll \{x \mapsto d\}\}.$$

Note that χ actions can change the valuation only in one way, i.e., the effect of every action is a singleton set.

To simplify the presentation we only consider $\Pi = \{x{=}e \mid x \in V, e \in E\}$ to be the set of atomic propositions (generalization to the full set of χ propositions, including relations other than just equality, is easy). Let B be the set of boolean expressions over the set Π and assume that B includes the set of truth values $\{true, false\}$. We set $\mathsf{check}(x{=}e, \sigma) = true$ iff $\sigma(x) = d = \sigma(e)$ for some $d \in D$. Recall that the function check extends to B.

We now explain how χ process terms are mapped to κ.

The processes $\mathsf{b} :\rightarrow p$, $p \,;\, q$, $p \,[]\, q$, $[\![s \mid p]\!]$, $\partial_\Xi(p)$ and $\tau_I(p)$ map trivially. Recall that infinite repetition $*p$ is expressed in κ as $p^* \cdot \delta$. The guarded repetition $\mathsf{b} \overset{*}{\rightarrow} p$ is also easily interpreted as $(\mathsf{b} :\rightarrow p)^* \cdot (\neg\mathsf{b} :\rightarrow \boldsymbol{\tau})$.

Timing in χ is dense and so cannot directly be embedded into κ. However, the delays in χ range over rational numbers and so there is always a number that all can be multiplied by to obtain natural delays of the same ratios. Therefore, there is no loss of expressivity if in a χ specification the timing is discrete. The only difference between the delay operators in χ and κ is that in χ delaying zero time units is equivalent to $\boldsymbol{\tau}$ and not to ε like in κ. This is resolved by taking the process $\Delta e \cdot \boldsymbol{\tau}$ to be the κ interpretation of the χ process Δe.

The delayable versions of send and receive is obtained by $m!e = (\Delta 1)^* \cdot m!!e$ and $m?x = (\Delta 1)^* \cdot m??x$. To express the urgent communication operator in κ we can use the maximal progress operator with its parameter set including the desired set of communication actions.

We denote the set of χ processes interpreted in κ by P_χ. The following theorem shows an important property of this set. The processes in P_χ cannot terminate immediately. Moreover, after an action the resulting process can terminate only if it cannot do anything else, and after a tick the resulting process cannot terminate. This property is crucial for the correctness of the translation to PROMELA.

Theorem 6.2.1 For all $p \in P_\chi$ and all $\sigma \in \Sigma$, the following holds:

1. $\langle p, \sigma\rangle \not\downarrow$,
2. if $\langle p, \sigma\rangle \xrightarrow{a} \langle p', \sigma'\rangle$ for some $a \in \mathsf{A}$, $p' \in P$, and $\sigma' \in \Sigma$, then either $p' \in P_\chi$ or $p' \approx_s^\Delta \varepsilon$, and

3. if $\langle p, \sigma\rangle \overset{\Delta}{\mapsto} \langle p', \sigma'\rangle$ for some $p' \in P$ and $\sigma' \in \Sigma$, then $p' \in P_\chi$. □

Proof Since ε is not in P_χ, since repetition in χ is either infinite or exited with a τ, and since Δe is always followed by $\boldsymbol{\tau}$, the theorem can now be easily proven by the structural induction on P_χ. ■

Since χ can be fully interpreted in κ, we have timed silent congruence defined for P_χ as well. In the next section we show how it can be used to establish the correctness of some syntactical reductions needed when translating χ to PROMELA.

6.3 Translation to Promela

In [95] it is pointed out that the translation of some χ constructs is straightforward (e.g., for assignments and alternative composition), since they also exist in PROMELA. However, the translation of guards, nested scopes and nested parallelism is less straightforward, since they have no direct equivalents in PROMELA. We recall some results from [95].

Translation of guards A χ process $\mathsf{b}\!:\to p$ cannot be directly translated to PROMELA. The reason is that the guards in PROMELA act as statements that are executable if they evaluate to true. This means that the executability of a guard in PROMELA depends only on the validity of the guard. This is different from χ which looks for both b to be *true* and for p to be executable before taking the step. A typical example is the χ process $true\!:\to\delta \,[]\, true\!:\to$ skip which can only perform the action τ and terminate, while its naive translation to PROMELA could pass the first guard (because it is always true) and deadlock. However, as explained in [95], if p is an atomic χ process, i.e. if $p \in \{\delta, \varepsilon, m??x, m!!e\}$, then $\mathsf{b}:\to p$ can be correctly translated to PROMELA. If all guards are to be translated, a possible solution is to push them down to the level of atomic processes. The preprocessing phase of the translation process presented in [95] provides the rules to achieve that. These rules are presented in Table 6.1. The first rule is meant to be applied only when p is an atomic process. Its purpose is to ensure that *all* atomic processes are guarded which is more convenient for implementation.

It is clear that, if a parallel composition does not appear in the process p, then the rules from Table 6.1 are enough to construct a process q that is equivalent to $\mathsf{b}:\to p$ and in which only atomic (sub)processes are guarded.

p	$true :\rightarrow p$
$\mathsf{b}_1 :\rightarrow \mathsf{b}_2 :\rightarrow p$	$\mathsf{b}_1 \wedge \mathsf{b}_2 :\rightarrow p$
$\mathsf{b} :\rightarrow (p \mathbin{[\!]} q)$	$(\mathsf{b} :\rightarrow p) \mathbin{[\!]} (\mathsf{b} :\rightarrow q)$
$\mathsf{b} :\rightarrow (p \mathbin{;} q)$	$(\mathsf{b} :\rightarrow p) \mathbin{;} q$
$\mathsf{b} :\rightarrow *p$	$(\mathsf{b} :\rightarrow p) \mathbin{;} *p$
$\mathsf{b}_1 :\rightarrow \mathsf{b}_2 \xrightarrow{*} p$	$((\mathsf{b}_1 \wedge \mathsf{b}_2 :\rightarrow \mathsf{skip}) \mathbin{;} \mathsf{b}_2 \xrightarrow{*} p) \mathbin{[\!]} (\mathsf{b}_1 \wedge \neg\mathsf{b}_2 :\rightarrow \mathsf{skip})$

Table 6.1: Simplification of guards

Nested parallelism is a problem here, but also in other places. We will discuss that problem later.

Timed silent congruence can serve as a correctness criterion for the guard simplification rules. It is not hard to see that every process in the left column of Table 6.1 is timed silent congruent to the corresponding process in the right column. We only give a sketch of the proof that $\mathsf{b}{:}\rightarrow *p \approx_s^{\Delta} (\mathsf{b}{:}\rightarrow p) ; *p$. In κ terms this is expressed by the following theorem.

Theorem 6.3.1 For all $p \in P_\chi$ and all $\mathsf{b} \in \mathsf{B}$, $\mathsf{b}{:}\rightarrow(p^* \cdot \delta) \approx_s^{\Delta} (\mathsf{b}{:}\rightarrow p) \cdot p^* \cdot \delta$.□

Proof Let I be the identity relation on P. Let

$$S = I \cup \{(\mathsf{b} :\rightarrow (p^* \cdot \delta), (\mathsf{b} :\rightarrow p) \cdot p^* \cdot \delta), ((\mathsf{b} :\rightarrow p) \cdot p^* \cdot \delta), (\mathsf{b} :\rightarrow (p^* \cdot \delta))\}$$

and $R = S$. It is not hard to show that (S, R) is a stateless timed silent congruence. ■

Note that for the simplification of guards we do not need to use the full power of timed silent congruence. To show that the rules are correct we do not need an equivalence that abstracts away from internal steps; the original and the simplified process agree on every step they take (this is why we could define $R = S$ in the proof of the previous theorem). In the next two paragraphs we show some reduction rules that rely on timed silent congruence.

Elimination of nested scopes A Promela specification consists of global variables and a sequence of process definitions. Every process definition allows for the declaration of local variables. Therefore, there are only two scope levels, process-local, in process definitions, and global, outside of them. It is not possible to introduce blocks inside the process declarations with block-local variables. Since χ features the scope operator, local variables can be introduced anywhere. To translate a χ process to Promela we must make sure that scopes are in proper places. Elimination of nested scopes is in most cases trivial after some variables are properly renamed. When a scope is in the context of a repetition, the elimination is more complicated. Note that the process $*[\![s \mid p]\!]$ has different behavior than $[\![s \mid *p]\!]$. This is because p in $*[\![s \mid p]\!]$, when it has finished executing, starts again in the 'fresh valuation' s while p in $[\![s \mid *p]\!]$ starts from a possibly modified valuation. A solution is to make p restore the old valuation when it is done. The rules for nested scope elimination, also a part of the preprocessing phase from [95], are given in Table 6.2. It is required that the free variables in guards and in processes outside the scope of s are first renamed if needed (i.e. if these variables also appear in s). In the last two cases we assume that $s = \{x_1 \mapsto d_1, \ldots, x_n \mapsto d_n\}$ and that $I = \{\mathsf{asgn}[(x_1, d_1), \ldots, (x_n, d_n)]\}$

$[\![- \mid p]\!]$	p
$\mathsf{b} :\rightarrow [\![s \mid p]\!]$	$[\![s \mid \mathsf{b} :\rightarrow p]\!]$
$[\![s \mid p]\!] \,;\, q$	$[\![s \mid p \,;\, q]\!]$
$p \,;\, [\![s \mid q]\!]$	$[\![s \mid p \,;\, q]\!]$
$[\![s \mid p]\!] \mathbin{[\,]} q$	$[\![s \mid p \mathbin{[\,]} q]\!]$
$[\![s \mid p]\!] \parallel q$	$[\![s \mid p \parallel q]\!]$
$[\![s_1 \mid [\![s_2 \mid p]\!]]\!]$	$[\![s_1 \ll s_2 \mid p]\!]$
$*[\![s \mid p]\!]$	$[\![s \mid *(p \,;\, \tau_I(x_1, \ldots, x_n := d_1, \ldots, d_n))]\!]$
$\mathsf{b} \stackrel{*}{\rightarrow} [\![s \mid p]\!]$	$[\![s \mid \mathsf{b} \stackrel{*}{\rightarrow} (p \,;\, \tau_I(x_1, \ldots, x_n := d_1, \ldots, d_n))]\!]$

Table 6.2: Elimination of nested scopes

Every process in the left column of Table 6.1 is timed silent congruent to its corresponding process in the right column. Note that in the last two rules we needed to introduce an additional action, that is an assignment, to establish the proper reductions. The hiding operator renames these assignments to τ's, and since they only change the local valuation, they give rise to internal steps.

As we did for guards, we only prove one non-trivial rule in Table 6.2. We choose to prove that $*[\![s \mid p]\!] \approx_s^\Delta [\![s \mid *(p \,;\, x_1, \ldots, x_n := d_1, \ldots, d_n)]\!]$ and, for clarity, only consider the special case when $n = 1$. In κ terms we have the following:

Theorem 6.3.2 For all $p \in P_\chi$, all $x \in V$, and all $d \in D$, $[\![\{x \mapsto d\} \mid p]\!]^* \cdot \delta \approx_s^\Delta [\![\{x \mapsto d\} \mid (p \cdot \tau_I(x := d))^* \cdot \delta]\!]$ with $I = \{\mathsf{asgn}[(x, d)]\}$. $\square$

Proof Let P_χ^ε be the set of processes such that $p \in P_\chi^\varepsilon$ iff $p \in P_\chi$ or $p \approx_s^\Delta \varepsilon$.

Define

$$
\begin{aligned}
S \;=\; & \{([\![\{x \mapsto d\} \mid p]\!]^* \cdot \delta, [\![\{x \mapsto d\} \mid (p \cdot \tau_I(x := d))^* \cdot \delta]\!])\} \\
& \cup \{([\![\{x \mapsto c\} \mid r]\!] \cdot [\![\{x \mapsto d\} \mid p]\!]^* \cdot \delta, \\
& \qquad [\![\{x \mapsto c\} \mid r \cdot \tau_I(x := d) \cdot (p \cdot \tau_I(x := d))^* \cdot \delta]\!]) \mid r \in P_\chi, c \in D\} \\
& \cup \ \ldots \text{(symmetric pairs)}.
\end{aligned}
$$

and

$$
\begin{aligned}
R \;=\; & S \\
& \cup \{([\![\{x \mapsto c\} \mid r]\!] \cdot [\![\{x \mapsto d\} \mid p]\!]^* \cdot \delta, \\
& \qquad [\![\{x \mapsto c\} \mid r \cdot \tau_I(x := d) \cdot (p \cdot \tau_I(x := d))^* \cdot \delta]\!]) \mid r \in P_\chi^\varepsilon, c \in D\} \\
& \cup \{([\![\{x \mapsto c\} \mid r]\!] \cdot [\![\{x \mapsto d\} \mid p]\!]^* \cdot \delta, \\
& \qquad [\![\{x \mapsto d\} \mid (p \cdot \tau_I(x := d))^* \cdot \delta]\!] \mid r \approx_s^\Delta \varepsilon, c \in D\} \\
& \cup \ \ldots \text{(symmetric pairs)}.
\end{aligned}
$$

It is easy to show that (S, R) is a timed stateless silent congruence. ■

Nested parallelism As we said before, a Promela specification consists of a list of process definitions. There is no (explicit) operator for parallel composition, and processes are either implicitly executed in parallel or started by a special statement. This statement is always executable and, therefore, can make a choice if executed in the context of alternative composition. Because of this, similar problems as in the naive translation of guards appear. The solution is to eliminate nested parallelism on the χ level, i.e. to move it to the outermost level. This, unfortunately, is rarely possible.

Remark 6.3.3 In some cases it is possible to linearize χ specifications, that is to eliminate all parallelism. This however is not a desired solution because it would drastically move us away from the original specification, and we would not be fully exploiting the Spin's powerful verification mechanism. Therefore, linearization should be performed only if it is the last option. □

Nested parallelism can be eliminated in the context of sequential composition and repetition. This is because a sequential composition can be simulated by a parallel composition at the expense of introducing an extra synchronization variable, and repetition can be distributed over a parallel composition with a proper synchronization mechanism to restrict the possible additional behavior. This technique is introduced in [95] and shown in Table 6.3. To apply these rules we must ensure that w is an unused variable, i.e. that p, q and r cannot change its value, and similarly for the channel s. The variable x is assumed not to be free in the whole specification; we could alternatively declare it locally, i.e., replace $s?x$ by e.g. $[\![x \mapsto 0 \mid s?x]\!]$. For the first rule we assume that $I = \{\mathsf{asgn}[(w,1)\}$. For the second and the third rule we assume that $I = \{\mathsf{comm}[(s,x,0)]\}$ and $\Xi = \{\mathsf{snd}(s,0)\} \cup \{\mathsf{rcv}(s,x,d) \mid d \in D\}$. Note that the technique can be easily be extended from two to an arbitrary number of parallel components.

$p\,;(q \parallel r)$	$[\![\{w \mapsto 0\} \mid p\,;\tau_I(w := 1) \parallel w = 1 :\to q \parallel w = 1 :\to r]\!]$
$(p \parallel q)\,; r$	$\tau_I(\partial_\Xi(\ p\,; s!0 \parallel q\,; s!0 \parallel s?x\,; s?x\,; r\ ))$
$*(p \parallel q)$	$\tau_I(\partial_\Xi(\ *(p\,; s!0) \parallel *(q\,; s?x)\ ))$.

Table 6.3: Elimination of nested parallelism

As before, it can be proven that both rules in Table 6.3 are correct modulo timed silent congruence. We only do this for the first rule.

Theorem 6.3.4 For all $p, q, r \in P_\chi$, all $x \in V$ and all $d \in D$, $p \cdot (q \parallel r) \approx_s^\Delta [\![w \mapsto 0 \mid p \cdot \tau_I(w := 1) \parallel w = 1 :\to q \parallel w = 1 :\to r]\!]$ with $I = \{\mathsf{asgn}[(w,1)\}$. $\square$

Proof Let P_χ^ε be the set of processes such that $p \in P_\chi^\varepsilon$ iff $p \in P_\chi$ or $p \approx_s^\Delta \varepsilon$. Define

$$
\begin{aligned}
S \quad = \quad & \{(x \cdot (q \parallel r), \\
& \quad [\![\{w \mapsto 0\} \mid x \cdot \tau_I(w := 1) \parallel w = 1 :\to q \parallel w = 1 :\to r]\!]) \mid x \in P_\chi\} \\
& \cup \ \ldots \text{(symmetric pairs).}
\end{aligned}
$$

and

$$
\begin{aligned}
R \quad = \quad & S \\
& \cup \{(x \cdot (q \parallel r), \\
& \quad [\![\{w \mapsto 0\} \mid x \cdot \tau_I(w := 1) \parallel w = 1 :\to q \parallel w = 1 :\to r]\!]) \mid x \in P_\chi^\varepsilon\} \\
& \cup \{(x \cdot (q \parallel r), [\![\{w \mapsto 1\} \mid \varepsilon \parallel w = 1 :\to q \parallel w = 1 :\to r]\!]) \mid x \approx_s^\Delta \varepsilon\} \\
& \cup \{(x \parallel y, [\![\{w \mapsto 1\} \mid \varepsilon \parallel x \parallel w = 1 :\to y]\!]) \mid x, y \in P_\chi^\varepsilon\} \\
& \cup \{(x \parallel y, [\![\{w \mapsto 1\} \mid \varepsilon \parallel w = 1 :\to x \parallel y]\!]) \mid x, y \in P_\chi^\varepsilon\} \\
& \cup \{(x \parallel y, [\![\{w \mapsto 1\} \mid \varepsilon \parallel x \parallel y]\!]) \mid x, y \in P_\chi^\varepsilon\} \\
& \cup \{(x \parallel y, [\![\{w \mapsto 1\} \mid x \parallel y]\!]) \mid x, y \in P_\chi^\varepsilon\} \\
& \cup \{(x, [\![\{w \mapsto 1\} \mid x]\!]) \mid x \in P_\chi^\varepsilon\} \\
& \cup \ \ldots \text{(symmetric pairs).}
\end{aligned}
$$

It is tedious but straightforward to verify that (S, R) is a timed stateless silent congruence. ■

Note that it is not so convenient to, instead of the second rule in Table 6.3, have the rule that transform $(p \parallel q)\,;\, r$ to, for example, the process $[\![\{w \mapsto 0\} \mid p\,;\, w := w + 1 \parallel q\,;\, w := w + 1 \parallel w = 2 :\to r]\!]$. The reason is that with this rule the set of rewriting rules is not terminating. We give an example. The process $*((p \parallel q)\,;\, r)$ is first transformed to the process $*[\![\{w \mapsto 0\} \mid p\,;\, w := w + 1 \parallel q\,;\, w := w + 1 \parallel w = 2 :\to r]\!]$, and then further to $[\![\{w \mapsto 0\} \mid *((p\,;\, w := w + 1 \parallel q\,;\, w := w + 1 \parallel w = 2 :\to r)\,;\, w := 0)]\!]$. We are back where we started. There is, of course, a way out, which is to push the assignment $w := 0$ directly behind r. This however means that the situation $*((p \parallel q)\,;\, r)$ would then require its own rule which would make the syntactic definition of the translatable subset even more complicated.

Remark 6.3.5 One simple extension to χ (and κ) would allow us to eliminate nested parallelism in the context of guards. This is to let declaration of local variables to be of the form $x_1 \mapsto e_1, \ldots, x_n \mapsto e_n$ where e_i are arbitrary expressions and not only data values as now. Then we would simply have that $x{=}d :\to (p \parallel q)$ is the same as $[\![x' \mapsto x, x \mapsto d \mid x' :\to p \parallel x' :\to q]\!]$. However, by allowing expression in local scopes we lose the option to translate those scopes when they are inside a repetition. □

Conclusion to Part I

Our goal was to find an equivalence relation on process specifications that preserves all relevant properties of the system being modeled, and that is a congruence. The main correctness requirements was set to be the preservation of deadlock and the validity of formulas of CTL^*_{-X} temporal logic. We also wanted the equivalence not to be too restrictive giving us sufficient flexibility in establishing the correctness of symbolic transformations of the specification.

It is usually convenient to have a behavioral equivalence pertaining to the operational semantics, i.e. to define it as a bisimulation relation. In a simple setting, the notion of branching bisimulation directly corresponds to our criterion. In Part I we performed a sequence of extensions to adapt branching bisimulation to the more complicated setting with data, termination, explicit divergence, and timing. The adaptation was a conservative extension, assuring that the relevant properties are preserved. We proved that the obtained relation was compatible for all the constructs of a typical modeling language.

A proper treatment of divergence is crucial for certain interpretations of the notion of CTL^*_{-X} logic, and for timing. The addition of a divergence condition to the bisimulation brought in several complications when proving standard properties, such as transitivity and the stuttering property. We successfully solved this problem by introducing alternative (but equivalent) conditions, each more applicable in some situations than in others. Thereby we also solved the open problem of showing that the notions of branching bisimulation with explicit divergence of [53], and the stuttering equivalences of [85, 50, 81], are equivalence relations.

The addition of data introduced only the expected and known congruence problems; the standard solution could be easily adapted to our setting. The addition of termination alone introduced no problems. However, its combination with timing had the result that the bisimulation was no longer compatible with the sequential composition operator. To keep the congruence property we refined the timing and termination conditions of timed branching bisimulation. This solved the open problem of defining timed branching bisimulation in the setting with successful termination.

To illustrate the power of the equivalence we demonstrated how (a part of) the translation from the industrial modeling language χ to the popular model checker Spin could be proved correct. Several syntactic simplifications were given, e.g. to eliminate nested parallelism, and each was shown to be correct modulo our equivalence.

In the future we want to axiomatize our equivalence and to provide minimization algorithms.

Part II

Transition Systems and Bisimulations in Matrix Theory

Chapter 7

Introduction

As we already mentioned in the introduction to Part I, labeled transition systems (with termination) are a well established formalism for modeling qualitative aspects of systems. We also said that there existed a full spectrum of different equivalences for labeled transition systems [52, 51], each with a well-specified set of properties (usually represented as a set of temporal logic formulas) that it preserved. For example, if two systems agree on every step they take, then they are equivalent modulo *strong bisimulation* equivalence [84, 79]. If there is an action in the model, called the *internal action*, which is unobservable, then systems are usually related by *weak* [79] or by *branching* [53] bisimulation equivalence.

A popular method to obtain transition systems is by means of some expression in a process algebraic language. This method enables the generation of large models from smaller components. When fully built, almost all models of realistic systems suffer from the state explosion problem, i.e. their analysis is hardly ever possible due to their size. One solution to this problem is to reduce the model while keeping all relevant properties of the system. That leads to a vast number of methods used to reduce a system modulo the equivalences mentioned above. All methods are based on dividing states into equivalence classes to obtain a quotient system.

The state explosion problem is also present in formalisms for quantitative analysis of systems, such as in continuous time Markov chains. There the notion of *ordinary lumpability* [67, 82, 23], which corresponds to the notion of strong bisimulation, is used. The unobservable behavior in this setting can be seen as performing an immediate, i.e. timeless, step and in Part III we deal with this weaker form of lumping. The theory of Markov chains, and therefore of lumpability too, is almost always presented in terms of matrix theory. The well developed matrix apparatus has shown to be a powerful method for reasoning about Markov chains. It increases clarity and compactness, simplifies proofs, makes known results from linear algebra applicable which leads to new insights, etc.

Recently, as a consequence of the appearance of many stochastic process algebras and of their extensions of the Markov chain model, there has been some work on establishing some of the notions from [52] in the setting of Markov chains (see e.g. [10, 100]). We work here in the opposite direction and approach the theory of labeled transition systems and bisimulations from the setting of Markov chains, i.e. from matrix theory. We list some points that speak in favor of the matrix approach.

- The approach sets the theory in a new algebraic setting, i.e. on a boolean and relation algebra ground. It can be used as an alternative to or in combination with the standard process algebraic approach.

- The notion of (bi)simulation has been, in some forms, extensively studied in graph and modal logic theory by the methods of boolean matrices and relation algebras [91, 47]. Since relations on finite systems can also be represented as matrices with elements in the set $\{0, 1\}$, the extension of the known definitions to the setting of labeled transition systems is natural and sometimes even trivial.

- We expect the new proofs of old and known results to be shorter and, once one becomes used to the machinery, more readable and easier to check. This is expected since it is the case for the theory of Markov chains.

- The approach also has a didactical advantage in our case. We hope that it enables a reader not familiar with Markov chains, but familiar with the standard theory of labeled transition systems, to understand Markov chains faster. And vice versa, we also hope that a reader experienced with matrix techniques of Markov chain theory will have no problems understanding the labeled transition systems when presented in a similar way. That is why Part II has an important place in this thesis.

- The more-or-less unified setting points to many similarities between the theory of labeled transition systems and Markov chains, like e.g. it directly indicates the known fact that strong bisimulation reduction is the same as ordinary lumping. However, it also provides an automatic way to obtain some unknown but useful notions. For example, we will see that the notion of τ-lumping in Part III is just a weak bisimulation interpreted in the Markov chain context.

- The new interface to labeled transition systems is a big step towards a unified presentation of dynamical systems and to a unified theory of bisimulation as a major behavioral equivalence on systems. The new theory is initially rich since it combines the well-developed results from different environments and communities.

The matrix approach also has some disadvantages.

- Although the proofs are shorter and easier to check for correctness, they are obtained completely by algebraic reasoning and the usual intuition that exists in standard proofs is lost. In many cases, as we will see later, this is not a real problem and can also be seen as an advantage. It is almost always clear how to proceed with the proof, i.e. which algebraic formula to apply, and at the end the proof comes out quickly and in a completely mechanical way. However, in some cases a complicated formula must be applied and it is not always easy to recognize these situations. This is where it would be helpful to have some intuition. The conclusion is that the matrix approach works well but sometimes it needs to be used in combination with the standard one.
- The second big disadvantage of the approach is that it requests that the set of states is ordered. This is an unnecessary restriction and it forces us to, for each result, prove that it is independent of the ordering of states, i.e. insensitive to permutation. Moving from matrices to linear operators would make this problem disappear.
- Not every standard notion can be directly represented in terms of matrix theory. An example of this case, as we will see later, is the definition of branching bisimulation. The way out is to give a matrix definition of a similar notion and then show that it corresponds, in some sense, to the standard one.

Note that to use (special) matrices to represent different models of dynamic systems is not a novel approach. Matrices over a Kleene algebra have been successfully applied in automata theory [34, 72]. In [47], matrices over a boolean algebra were used to represent Kripke structures. There the notion of (strong) bisimulation was defined in matrix terms (as a relation between two structures). In [4], timed and stochastic event graphs were given in terms of matrices over a max-plus algebra. In [33], a general approach to aggregation of systems was given in terms of matrix theory over an idempotent semiring. Our notion of strong lumping is a special case of the lumping in [33]. Petri-nets are also modeled as relation algebras [49, 14].

7.1 Outline

In this section we present our approach in more detail, explaining every decision, and we show how the algebraic apparatus is put to work.

We work with finite state labeled transition systems with one starting state and with (successful) termination. The reason we consider only the case of finitely many states is just to simplify the presentation; there should be no problems when extending the theory to (at least) the countable case (due to the completeness property of boolean algebras). The reason for incorporating successful termination is to point out the direct parallel with the reward mechanism for Markov chains (they are dual).

We define a labeled transition system as a system of matrices. More precisely, as a triple of an initial vector, which indicates which of the states is the starting state, a transition matrix, which contains the actions that the system performs when transiting from one state to another, and a termination vector, which indicates which states are the successfully terminating ones. The starting state and the set of terminating states are modeled as vectors, and not, for example, as sets of indices, to enable them to interact with matrix T and to fully use the matrix algebraic approach.

We define standard operations on transition systems such as alternative, sequential and parallel composition in matrix terms. This is only to justify the approach more, that is to show its compactness, both in these definitions themselves, and later in the proof that bisimulations are compositional with respect to these operators. The representation of operators in terms of matrices is a direct application of the powerful block-matrix representation method common in every matrix setting.

In relation and boolean matrix algebras a relation is represented simply as a 0–1 matrix that indicates which pairs are related. A strong bisimulation is then a system of matrix inequalities involving a symmetric relation and the matrices representing a system. We define these inequalities in particular for our representation of labeled transition systems and show that the matrix definition corresponds to the standard notion of strong bisimulation. We treat strong bisimulation in this text because it is the most common relation between transition systems, and because the notion already exists, in some forms, in the relation algebra and graph theory.

There are many different aspects connected with strong bisimulation. We decided to deal only with the following few.

1. Bisimilarity is the most commonly used equivalence on transition systems and is therefore often used as a correctness criterion for their reduction. It is interesting to define this reduction in matrix terms because it corresponds to the notion of ordinary lumping [67, 23] from Markov chain theory. This is a known fact but is now directly seen from the matrix representation.

2. Backward bisimulation was introduced in [75] as a dual to the standard, i.e. forward bisimulation. It requires that every backward step in a system is simulated. The idea was introduced to Markov chains first as a form of lumping in [23], and then as a bisimulation relation in [93]. To show that the ideas correspond, we decided to treat backward bisimulation with our matrix techniques as well. Another reason is that it turns out that its matrix definition is obtained just by transposing the standard bisimulation conditions. This means that when working with backward bisimulation we can reuse all the results from strong bisimulation.

3. Bisimulation up-to techniques [79, 89] are often used to ease the definition of a witnessing relation when proving two systems bisimilar. Since the technique is very useful, we present it in our matrix setting as well. The method is unknown in the Markov chain world and by treating it in matrix terms we hope that in the future it will directly lead to some application there.

From strong bisimulation we move to bisimulations that to a certain degree ignore transitions labeled with the initial action τ, i.e. to those that abstract away from internal steps. These equivalences fit well into the matrix setting and their treatment there is shown to rely on many known complex results from matrix theory.

We first deal with weak bisimulation [79] since it ignores silent transitions in the most general way. It has a simple matrix characterization that uses the standard matrix definition of reflexive-transitive closure [91, 68] from relation algebra and boolean matrix theory. We also define the corresponding notion of lumping by linking weak bisimulation with strong bisimulation.

Although the matrix definition of a weak bisimulation should be enough to convince the reader that the matrix approach works well for equivalences weaker than strong bisimulation, we decided to also incorporate branching bisimulation [53] into our theory. The reason for this is not the known fact that branching bisimulation preserves the branching structure of a system more than weak bisimulation does, but the fact that it is not possible to express its definition directly in matrix terms as it was the case for the weak bisimulation (unless with a very strong requirement that bisimulation is transitive). We give a similar definition and show that it is equivalent to the standard one if the bisimulation relation is transitive. We also show that our branching bisimulation satisfies the so-called stuttering property, a property nicely expressible in matrix terms. We reestablish these results not only to illustrate compactness again, but more because they feature several direct applications of some of the important results from relational algebra, such as the Dedekind formula.

Note that in Part II we mostly (re)prove old and known results. However, there are some things that are new. For example, we provide an alternative version of branching bisimulation, and we show that every bisimulation we use can also be expressed as a strong bisimulation on a somehow transformed system. This is an important property because it allows for the direct reuse of all the results from the theory of strong bisimulation. This result is known for weak bisimulation but the method has not been adapted before for e.g. branching bisimulation. Moreover, for every bisimulation we provide a corresponding notion of lumping and prove its soundness with respect to the transformation of the system.

The structure of Part II is as follows.

In Chapter 8 we give an introduction to matrix theory over a boolean set algebra and we state some notions and important results from relation algebra. Next we define labeled transition systems and the standard operations on them in matrix terms.

Chapter 9 is about strong bisimulation. We define it in matrix terminology and prove that it coincides with the standard notion. We reprove some standard results and give the notion of lumping. We also extend strong bisimulation to a relation between two systems and use that to prove the compatibility with the operators. In the last two sections we introduce backward bisimulation and the bisimulation up-to technique.

In Chapter 10 we deal with systems with internal steps. We define weak and branching bisimulation relations. As we did for strong, we show that the new definitions match the standard ones, we reprove some standard results, and for each bisimulation we introduce a corresponding notion of lumping with its soundness proof. At the end we define the stuttering property and show that branching bisimulation satisfies it.

Chapter 8

Transition Systems as Matrices

A transition system is a directed graph in which each node represents a state of a system, and each arrow is labeled by an action denoting that the system can perform a transition from a state to another while executing that action. One state of the system is the initial state, and some states are considered successfully terminating. In this chapter we define finite state transition systems with termination in terms of matrices over a boolean algebra that is built from the set of actions.

First we give some preliminaries, mostly taken from [91] and [68].

8.1 Preliminaries

Let A be a set and let $\mathcal{P}(A)$ be the set of all subsets of A. Then $\mathbb{P}(\mathsf{A}) = (\mathcal{P}(A), +, \cdot, ^-, 0, 1)$ is a boolean algebra with $+$, $\cdot$, $^-$, 0 and 1 representing union, intersection, complement, the empty set and the full set A respectively. We use $+$, $\cdot$, 0 and 1 instead of $\cup$, $\cap$, $\emptyset$ and A to emphasize the connections with standard matrix theory and the theory of lumping in Markov chains.

$\mathbb{P}(\mathsf{A})^{n\times m}$ denotes the set of all $n \times m$ matrices with elements in $\mathbb{P}(\mathsf{A})$. Elements of $\mathbb{P}(\mathsf{A})^{1\times n}$ and $\mathbb{P}(\mathsf{A})^{n\times 1}$ are called vectors. $\mathbf{1}^n$ denotes the vector in $\mathbb{P}(\mathsf{A})^{n\times 1}$ that consists of n 1's. $\mathbf{0}^{n\times m}$ denotes the $n \times m$ matrix consisting entirely of zeroes. I^n denotes the $n \times n$ *identity matrix*. We omit the n and m when they are clear from the context. A matrix A of which every element is either 0 or 1, i.e. an element of $\{0,1\}^{n\times m}$, is called a 0–1 matrix. Sometimes we also call a 0–1 matrix R a *relation*. This is to emphasize the intuitive fact that $R[i, j] = 1$ iff the i-th and the j-th element are in a relation.

We now introduce some operations on the set $\mathbb{P}(\mathsf{A})^{n\times m}$.

Sum For $A, B \in \mathbb{P}(\mathsf{A})^{n\times m}$, the sum $A + B \in \mathbb{P}(\mathsf{A})^{n\times m}$ is defined by:

$$(A + B)[i, j] = A[i, j] + B[i, j] \text{ for } i = 1, \ldots, n \text{ and } j = 1, \ldots, m.$$

Scalar product For $A \in \mathbb{P}(\mathsf{A})^{n\times m}$ and $\alpha \in \mathbb{P}(\mathsf{A})$, the element product $\alpha \cdot A \in \mathbb{P}(\mathsf{A})^{n\times m}$ is defined by:

$$(\alpha \cdot A)[i, j] = \alpha \cdot A[i, j] \text{ for } i = 1, \ldots, n \text{ and } j = 1, \ldots, m.$$

Product For $A \in \mathbb{P}(\mathsf{A})^{n\times p}$ and $B \in \mathbb{P}(\mathsf{A})^{p\times m}$ the product $A \cdot B \in \mathbb{P}(\mathsf{A})^{n\times m}$ is defined by:

$$(A \cdot B)[i, j] = \sum_{k=1}^{p} A[i, k] + B[k, j] \text{ for } i = 1, \ldots, n \text{ and } j = 1, \ldots, m.$$

Intersection For $A, B \in \mathbb{P}(\mathsf{A})^{n\times m}$, the intersection $A \sqcap B \in \mathbb{P}(\mathsf{A})^{n\times m}$ is defined by:

$$(A \sqcap B)[i, j] = A[i, j] \cdot B[i, j] \text{ for } i = 1, \ldots, n \text{ and } j = 1, \ldots, m.$$

Complement For $A \in \mathbb{P}(\mathsf{A})^{n\times m}$, the complement $\bar{A} \in \mathbb{P}(\mathsf{A})^{n\times m}$ is defined by:

$$\bar{A}[i, j] = \overline{A[i, j]} \text{ for } i = 1, \ldots, n \text{ and } j = 1, \ldots, m.$$

Transpose For $A \in \mathbb{P}(\mathsf{A})^{n\times m}$, the transpose $A^{\mathsf{T}} \in \mathbb{P}(\mathsf{A})^{m\times n}$ is defined by:

$$A^{\mathsf{T}}[i, j] = A[j, i] \text{ for } i = 1, \ldots, n \text{ and } j = 1, \ldots, m.$$

Kronecker product and sum The notion of Kronecker product comes from the standard matrix theory where it has many applications (see e.g. [55]). For example, it has been successfully used for the decomposition of Markov chains [24] into smaller parallel components. The notion directly maps to our setting and it is the core of our definition of parallel composition with synchronization for transition systems.

The Kronecker product of two matrices is a block matrix in which every block is the scalar product of an element from the first matrix and the entire second matrix. We give the formal definition. For $A \in \mathbb{P}(\mathsf{A})^{n_1 \times n_2}$ and $B \in \mathbb{P}(\mathsf{A})^{m_1 \times m_2}$, the *Kronecker product* of A and B is the matrix $A \otimes B \in \mathbb{P}(\mathsf{A})^{n_1 m_1 \times n_2 m_2}$ defined by

$$(A \otimes B)[(i-1)m_1 + k, (j-1)m_2 + \ell] = A[i,j] \cdot B[k,\ell]$$

for $i = 1, \dots, n_1$, $j = 1, \dots, n_2$, $k = 1, \dots, m_1$ and $\ell = 1, \dots, m_2$.

The *Kronecker sum* of two (square) matrices $A \in \mathbb{P}(\mathsf{A})^{n \times n}$ and $B \in \mathbb{P}(\mathsf{A})^{m \times m}$ is the matrix $(A \oplus B) \in \mathbb{P}(\mathsf{A})^{mn \times mn}$ defined by

$$A \oplus B = A \otimes I^m + I^n \otimes B.$$

We write AB instead of $A \cdot B$. We assume that $\cdot$ and $\otimes$ bind stronger than the intersection and sum, and that the intersection is stronger than the sum. We also write

$$A \leqslant B \quad \text{for} \quad A + B = B.$$

We now list some properties of these operations assuming that the matrices A, B, C and D are such that all the operations are well defined.

Properties of transpose:

$$\begin{aligned}
&\bar{A}^\mathsf{T} = \overline{A^\mathsf{T}},\\
&(A^\mathsf{T})^\mathsf{T} = A,\\
&A \leqslant B \quad \text{iff} \quad A^\mathsf{T} \leqslant B^\mathsf{T},\\
&(A+B)^\mathsf{T} = A^\mathsf{T} + B^\mathsf{T},\\
&(AB)^\mathsf{T} = B^\mathsf{T} A^\mathsf{T},\\
&(A \sqcap B)^\mathsf{T} = A^\mathsf{T} \sqcap B^\mathsf{T},
\end{aligned}$$

Properties of sum:

$$\begin{aligned}
&A + A = A,\\
&A + B = B + A,\\
&(A + B) + C = A + (B + C).
\end{aligned}$$

Properties of intersection:

$$\begin{aligned}
&A \sqcap B = B \sqcap A,\\
&(A \sqcap B) \sqcap C = A \sqcap (B \sqcap C),\\
&A \sqcap B \leqslant A,\\
&\text{if } A \leqslant B, \text{ then } A \sqcap B = A.
\end{aligned}$$

Properties of scalar and matrix product:

$$\begin{array}{l}\alpha(AB) = (\alpha A)B = A(\alpha B),\\ (AB)C = A(BC),\\ AI = IA = A,\\ A\mathbf{0} = \mathbf{0}A = \mathbf{0},\\ \text{if } A \leqslant B \text{ and } C \leqslant D, \text{ then } AB \leqslant CD,\\ \alpha(A+B) = \alpha A + \alpha B,\\ A(B+C) = AB + AC, \quad (A+B)C = AC + BC,\\ A(B \sqcap C) \leqslant AB \sqcap AC, \quad (A \sqcap B)C \leqslant AC \sqcap BC.\end{array}$$

Schröder equivalences

$$AB \leqslant C \quad \text{iff} \quad A^{\mathsf{T}}\bar{C} \leqslant \bar{B} \quad \text{iff} \quad \bar{C}B^{\mathsf{T}} \leqslant \bar{A}.$$

Dedekind formula

$$AB \sqcap C \leqslant (A \sqcap CB^{\mathsf{T}})(B \sqcap A^{\mathsf{T}}C).$$

Properties of Kronecker product and sum

$$\begin{array}{l}(A \otimes B)(C \otimes D) = AC \otimes BD\\ (A+B) \otimes (C+D) = A \otimes C + A \otimes D + B \otimes C + B \otimes D\\ \alpha(A \otimes B) = (\alpha A \otimes B) = (A \otimes \alpha B)\\ \alpha(A \oplus B) = (\alpha A \oplus \alpha B)\\ (A+B) \oplus (C+D) = A \oplus C + B \oplus D\\ (A \otimes B)^{\mathsf{T}} = A^{\mathsf{T}} \otimes B^{\mathsf{T}}\end{array}$$

A relation $R \in \{0,1\}^{n \times n}$ is called *reflexive* if $I \leqslant R$, *symmetric* if $R^{\mathsf{T}} = R$ and *transitive* if $R^2 \leqslant R$. It is an *equivalence* relation if it is reflexive, symmetric and transitive.

Given a relation $R \in \{0,1\}^{n \times n}$, we call

$$R^+ = \sum_{n=1}^{\infty} R^n \quad \text{and} \quad R^* = \sum_{n=0}^{\infty} R^n$$

the *transitive* and the *reflexive-transitive closure* respectively of R. Note that $R^+ = RR^*$ and $R^* = I + R^+$. For the two closures we have the following properties:

$$(R^+)^\mathsf{T} = (R^\mathsf{T})^+, \quad (R^*)^\mathsf{T} = (R^\mathsf{T})^*,$$
$$(R+S)^+ = R^+ + (R^*S)^+R^*, \quad (R+S)^* = (R^*S)^*R^*.$$

For an equivalence relation R we also have

$$(AR \sqcap B)R = AR \sqcap BR = (A \sqcap BR)R.$$

A relation $R \in \{0,1\}^{n\times m}$ is called *total* if $RR^\mathsf{T} \geqslant I$. Note that a reflexive relation is always total.

A relation $P \in \{0,1\}^{n\times m}$ is called an *isomorphism* if $PP^\mathsf{T} = I^n$ and $P^\mathsf{T}P = I^m$. It can be shown that P is an isomorphism iff its every row and every column contains exactly one nonzero entry. Note that this condition implies that $n = m$, i.e. that P must be a square matrix. Clearly, if P is an isomorphism, then so is P^T. We say that two matrices $A, B \in \mathbb{P}(\mathsf{A})^{n\times n}$ are equal modulo isomorphism if there exists an isomorphism $P \in \{0,1\}^{n\times n}$ such that $A = PBP^\mathsf{T}$. Isomorphism corresponds to the notion of permutation matrix in classical matrix theory.

8.2 Transition systems

A transition system (with the set of actions A and the set of states S) is standardly defined as a quadruple $(S, \rightarrow, S_0, \downarrow)$ where $\rightarrow \subseteq S \times \mathsf{A} \times S$ is called the *transition relation*, $s_0 \in S$ is the *initial state* and $\downarrow \subseteq S$ is the set of *(successfully) terminating states*.

In matrix terms we define a transition system as a triple of a 0–1 row vector that indicates which of the states is initial, a matrix over a subset of actions that contains the actions that the system performs when transiting from one state to another, and a 0–1 vector that indicates which states are terminating.

Definition 8.2.1 (Transition system) A *transition system* (with the set of actions A and of the dimension n) is a triple $\langle \sigma, A, \rho \rangle$ where:

- $\sigma \in \{0,1\}^{1\times n}$ is the *initial vector*; its exactly one entry is 1,
- $A \in \mathbb{P}(\mathsf{A})^{n\times n}$ is the *transition matrix*, and
- $\rho \in \{0,1\}^{n\times 1}$ is the *termination vector*.

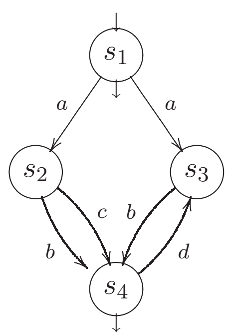
The set of all transition systems with the set of actions A and of the dimension n is denoted $\mathcal{TS}^n_{\mathsf{A}}$. □

If $S = \{s_1, \ldots, s_n\}$, our definition is obtained from the standard one by putting:

$$A[i,j] = \{a \mid s_i \xrightarrow{a} s_j\}, \quad \sigma[i] = \begin{cases} 1, & \text{if } s_i = s_0 \\ 0, & \text{if } s_i \neq s_0 \end{cases} \quad \text{and} \quad \rho[i] = \begin{cases} 1, & \text{if } s_i\downarrow \\ 0, & \text{if } s_i\not\downarrow. \end{cases}$$

That is, for each two states s_i and s_j, $A[i,j]$ contains the set of actions that the system can perform by going from s_i to s_j. The i-th element of σ is 1 if the state s_i is initial. The i-th element of ρ is either 0 or 1 depending if the state s_i is terminating or not. It is clear that, given an ordered S, we can obtain the standard definition from our definition easily.

Example 8.2.2 Figure 8.1 depicts a transition system and gives its matrix representation. The set of states is $S = \{s_1, s_2, s_3, s_4\}$, the set of actions is $\mathsf{A} = \{a, b, c\}$. State s_1 is the initial state; states s_1 and s_4 are terminating.□



$$\sigma = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}$$

$$A = \begin{pmatrix} 0 & \{a\} & \{a\} & 0 \\ 0 & 0 & 0 & \{b,c\} \\ 0 & 0 & 0 & \{b\} \\ 0 & 0 & \{d\} & 0 \end{pmatrix} \qquad \rho = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

Figure 8.1: Transition system and its matrix representation – Example 8.2.2

8.3 Operations on transition systems

In this section we define some special transition systems and some standard operations on transition systems, namely alternative and sequential composition, repetition and parallel composition.

Terminated system (ε) The transition system $\langle \sigma, A, \rho \rangle \in \mathcal{TS}^1_{\mathsf{A}}$ defined by

$$\sigma = \rho = \begin{pmatrix} 1 \end{pmatrix} \quad \text{and} \quad A = \begin{pmatrix} 0 \end{pmatrix}$$

we call the *terminated system* and we denote it by ε. This system has only one state in which it starts, terminates and cannot do an action.

Deadlocked system The transition system $\langle \sigma, A, \rho \rangle \in \mathcal{TS}^1_{\mathsf{A}}$ defined as

$$\sigma = \begin{pmatrix} 1 \end{pmatrix} \quad \text{and} \quad \rho = A = \begin{pmatrix} 0 \end{pmatrix}$$

we call the *deadlocked system* and denote it by δ. This system has only one state in which it starts and cannot do an action nor terminate.

Action execution Let $a \in \mathsf{A}$. The transition system $\langle \sigma, A, \rho \rangle \in \mathcal{TS}^2_{\mathsf{A}}$ defined as

$$\sigma = \begin{pmatrix} 1 & 0 \end{pmatrix}, \quad A = \begin{pmatrix} 0 & \{a\} \\ 0 & 0 \end{pmatrix} \quad \text{and} \quad \rho = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

is the *action execution* of a, and is denoted by $\boldsymbol{a}$. This system starts in the first state, performs the action a and goes to the second state in which it terminates.

Before we define the operations on transition systems we explain some of the matrix products that we will use. Let $\langle \sigma, A, \rho \rangle \in \mathcal{TS}^n_{\mathsf{A}}$. The product $\sigma A \in \mathbb{P}(\mathsf{A})^{n \times n}$ is the row vector that contains for each state the actions that can be performed initially while transiting to that state. The product $\sigma\rho \in \{0, 1\}$ is 1 iff the initial state is also terminating.

Sequential composition ($\cdot$) Let $\langle \sigma_A, A, \rho_A \rangle \in \mathcal{TS}^n_{\mathsf{A}}$ and $\langle \sigma_B, B, \rho_B \rangle \in \mathcal{TS}^m_{\mathsf{A}}$. Then $\langle \sigma_A, A, \rho_A \rangle \cdot \langle \sigma_B, B, \rho_B \rangle$ is the transition system $\langle \sigma, T, \rho \rangle \in \mathcal{TS}^{n+m}_{\mathsf{A}}$ defined by

$$\sigma = \begin{pmatrix} \sigma_A & \mathbf{0} \end{pmatrix}, \quad T = \begin{pmatrix} A & \rho_A \sigma_B B \\ \mathbf{0} & B \end{pmatrix}, \quad \rho = \begin{pmatrix} \rho_A \sigma_B \rho_B \\ \rho_B \end{pmatrix}.$$

Sequential composition of two systems can perform everything that the first system can perform and, in the case the first system is terminating, also everything that the second systems can perform. In the latter case the composition proceeds as the second system. This intuition is captured by putting A and B on the diagonal and by the vector $\rho_A \sigma_B B$ which lets every terminating state of the system $\langle \sigma_A, A, \rho_A \rangle$

perform an action that $\langle \sigma_B, B, \rho_B \rangle$ initially can. Sequential composition is terminated when both systems are terminated; the vector $\rho_A \sigma_B \rho_B$ is either $\mathbf{0}$ or ρ_A depending if the initial state in $\langle \sigma_B, B, \rho_B \rangle$ is also terminating or not.

Alternative composition (+) Let $\langle \sigma_A, A, \rho_A \rangle \in \mathcal{TS}^n_{\mathsf{A}}$ and $\langle \sigma_B, B, \rho_B \rangle \in \mathcal{TS}^m_{\mathsf{A}}$. Then $\langle \sigma_A, A, \rho_A \rangle + \langle \sigma_B, B, \rho_B \rangle$ is the transition system $\langle \sigma, T, \rho \rangle \in \mathcal{TS}^{n+m+1}_{\mathsf{A}}$ defined by

$$\sigma = \begin{pmatrix} 1 & \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad T = \begin{pmatrix} 0 & \sigma_A A & \sigma_B B \\ \mathbf{0} & A & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & B \end{pmatrix}, \quad \rho = \begin{pmatrix} \sigma_A \rho_A + \sigma_B \rho_B \\ \rho_A \\ \rho_B \end{pmatrix}.$$

Alternative composition should describe the non-deterministic choice between two systems. We achieve this by adding a new state, setting it to be the initial one, and by letting it perform everything that the two systems initially can and then transit to a state in one of these systems. Note that we need this extra state to unwind a possible recursive behavior in either system. Alternative composition terminates only if one of the systems terminates. This is captured by having the $\sigma_A \rho_A + \sigma_B \rho_B \in \{0, 1\}$ as the first element of the termination vector.

Repetition (∗) Let $\langle \sigma, A, \rho \rangle \in \mathcal{TS}^n_{\mathsf{A}}$. Then $*\langle \sigma, A, \rho \rangle$ is the transition system $\langle \sigma', A', \rho' \rangle \in \mathcal{TS}^{n+1}_{\mathsf{A}}$ defined by

$$\sigma' = \begin{pmatrix} 1 & \mathbf{0} \end{pmatrix}, \quad A' = \begin{pmatrix} 0 & \sigma A \\ \mathbf{0} & A + \rho \sigma A \end{pmatrix}, \quad \rho' = \begin{pmatrix} 1 \\ \rho \end{pmatrix}.$$

The repetition operator repeats the process zero or more times. The vector $\rho \sigma A$ captures the idea that terminating states can also perform actions of the initial state. We introduce an extra state again for the possible unwinding. The system can repeat itself zero times, which is considered successful termination, and so the new state is also terminating state.

Parallel composition with synchronization ($\|_\Omega$) Let $\langle \sigma_A, A, \rho_A \rangle \in \mathcal{TS}^n_{\mathsf{A}}$ and $\langle \sigma_B, B, \rho_B \rangle \in \mathcal{TS}^m_{\mathsf{A}}$. Let $\Omega \subseteq \mathsf{A}$. Then $\langle \sigma_A, A, \rho_A \rangle \|_\Omega \langle \sigma_B, B, \rho_B \rangle$ is the transition system $\langle \sigma, T, \rho \rangle \in \mathcal{TS}^{nm}_{\mathsf{A}}$ defined by

$$\sigma = \sigma_A \otimes \sigma_B, \quad T = \bar{\Omega} \cdot (A \oplus B) + \Omega \cdot (A \otimes B), \quad \rho = \rho_A \otimes \rho_B.$$

For $\alpha \subseteq \mathbb{P}(\mathsf{A})$, the scalar product $\alpha \cdot A$ restricts A to the actions in α. The Kronecker product corresponds to the idea of synchronization and the Kronecker sum captures the interleaving part. Note that we relied on $\Omega \cdot A \otimes \Omega \cdot B = \Omega \cdot (A \otimes B)$ and $\bar{\Omega} \cdot A \oplus \bar{\Omega} \cdot B = \bar{\Omega} \cdot (A \oplus B)$. The idea that parallel composition terminates only if both systems terminate is captured in the vector $\rho_A \otimes \rho_B$.

For clarity, we choose to give only the CSP [64] style of parallel composition. Note that the more general ACP style [9] of parallel composition, can also be easily defined in matrix terms.

We give some examples now.

Example 8.3.1 a. We compute the process $\boldsymbol{a}^* \cdot \delta$. First

$$\boldsymbol{a}^* = \left\langle \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & \{a\} \\ 0 & 0 & \{a\} \\ 0 & 0 & \{a\} \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \right\rangle,$$

and so

$$\boldsymbol{a}^* \cdot \delta = \left\langle \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & \{a\} & 0 \\ 0 & 0 & \{a\} & 0 \\ 0 & 0 & \{a\} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} \right\rangle.$$

b. It easily follows that the process $\boldsymbol{a} + \boldsymbol{b}$ is defined by

$$\boldsymbol{a} + \boldsymbol{b} = \left\langle \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & \{a\} & 0 & \{b\} \\ 0 & 0 & \{a\} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \{b\} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} \right\rangle.$$

c. We now compute $\boldsymbol{a} \cdot \boldsymbol{b} \parallel_{\{b\}} \boldsymbol{b}$. First we obtain

$$\boldsymbol{a} \cdot \boldsymbol{b} = \left\langle \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & \{a\} & 0 & 0 \\ 0 & 0 & 0 & \{b\} \\ 0 & 0 & 0 & \{b\} \\ 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \right\rangle.$$

Then we have

$$\begin{pmatrix} 0 & \{a\} & 0 & 0 \\ 0 & 0 & 0 & \{b\} \\ 0 & 0 & 0 & \{b\} \\ 0 & 0 & 0 & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & \{b\} \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \{b\} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \{b\} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

and

$$\begin{pmatrix} 0 & \{a\} & 0 & 0 \\ 0 & \{a\} & 0 & \{b\} \\ 0 & 0 & 0 & \{b\} \\ 0 & 0 & 0 & 0 \end{pmatrix} \oplus \begin{pmatrix} 0 & \{b\} \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & \{b\} & \{a\} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \{a\} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \{b\} & 0 & 0 & \{b\} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \{b\} \\ 0 & 0 & 0 & 0 & 0 & \{b\} & \{b\} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \{b\} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \{b\} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

From this, we obtain

$$\boldsymbol{a} \cdot \boldsymbol{b} \parallel_{\{b\}} \boldsymbol{b} = \left\langle \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & \{a\} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \{a\} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \{b\} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \{b\} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \right\rangle .$$

□

Chapter 9

Strong bisimulation

Strong bisimulation is the most common relation on transition systems. In this chapter we define strong bisimulation in terms of matrix theory. First we define it as a relation on one system and, for the case when it is an equivalence relation, we give the corresponding aggregation method called lumping. We reprove some standard results, such that the biggest bisimulation is an equivalence relation, in order to convince the reader of the notational benefits of the approach. Then we extend the notion of strong bisimulation to a relation between two systems. We use this to show that strong bisimulation is compatible with the operators introduced in the previous chapter. In the end, we introduce the concepts of backward bisimulation and bisimulation up-to. We link the two notions with the standard bisimulation and give corresponding lumping methods.

9.1 Strong bisimulation on a system

Strong bisimulation relation on a transition system relates states that behave exactly in the same way. In other words, it relates states that can perform the same actions and have equal termination behavior. In matrix terms, the definition of strong bisimulation is just an extension of the notion of simulation from relation algebra to transition systems [47, 91].

Definition 9.1.1 (Strong bisimulation) A symmetric relation $R \in \{0,1\}^{n\times n}$ is called a *strong bisimulation on* the transition system $\langle \sigma, A, \rho \rangle \in \mathcal{TS}^n_{\mathsf{A}}$ if

$$\sigma \leqslant \sigma R, \quad RA \leqslant AR \quad \text{and} \quad R\rho \leqslant \rho. \qquad \square$$

Let us show that the new definition of strong bisimulation agrees with the standard one. First note that $\sigma \leqslant \sigma R$ means that the initial state is related to itself. Next, note that $a \in (RA)[i,j]$ iff there is a k such that $R[i,k] = 1$ and $a \in A[k,j]$. Similarly, $a \in (AR)[i,j]$ iff there is an ℓ such that that $a \in A[i,\ell]$ and $R[\ell,j] = 1$. The condition $RA \leqslant AR$ then says that

$$\begin{array}{ccc} s_i \overset{R}{-\!-\!-} s_k & & s_i \\ \quad\quad\; {\scriptstyle a}\big\downarrow & \text{implies} & \big\downarrow{\scriptstyle a} \\ \quad\quad\;\; s_j & & s_\ell \underset{R}{-\!-\!-} s_j . \end{array}$$

This clearly corresponds to the standard definition of strong bisimulation. Finally, note that $(R\rho)[i] = 1$ iff there is a j such that $R[i,j] = 1$ and $\rho[j] = 1$. Thus, the condition $R\rho \leqslant \rho$ says that:

$$s_i \overset{R}{-\!-\!-} s_j{\downarrow} \qquad \text{implies} \qquad s_i{\downarrow}.$$

This again matches with the standard definition.

We now give an example.

Example 9.1.2 Let $\langle \sigma, A, \rho \rangle$ with

$$\sigma = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \; A = \begin{pmatrix} 0 & \{a\} & \{a\} & 0 & 0 \\ 0 & 0 & 0 & \{b,c\} & 0 \\ 0 & 0 & 0 & \{b\} & \{c\} \\ 0 & 0 & \{d\} & 0 & 0 \\ 0 & 0 & \{d\} & 0 & 0 \end{pmatrix} \text{ and } \rho = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

be a transition system (see Figure 9.1). The relation

$$R = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

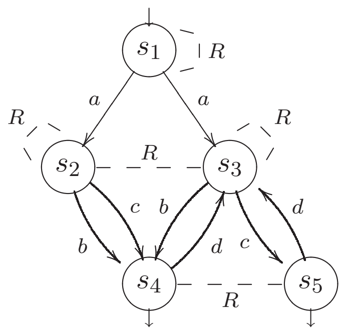
depicted in Figure 9.1 is a strong bisimulation on it because

$$RA = \begin{pmatrix} 0 & \{a\} & \{a\} & 0 & 0 \\ 0 & 0 & 0 & \{b,c\} & \{c\} \\ 0 & 0 & 0 & \{b,c\} & \{c\} \\ 0 & 0 & \{d\} & 0 & 0 \\ 0 & 0 & \{d\} & 0 & 0 \end{pmatrix} \leqslant \begin{pmatrix} 0 & \{a\} & \{a\} & 0 & 0 \\ 0 & 0 & 0 & \{b,c\} & \{b,c\} \\ 0 & 0 & 0 & \{b,c\} & \{b,c\} \\ 0 & \{d\} & \{d\} & 0 & 0 \\ 0 & \{d\} & \{d\} & 0 & 0 \end{pmatrix} = AR$$

and

$$R\rho = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} \leqslant \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \rho.$$

□



$$R = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Figure 9.1: Transition system and a strong bisimulation on it – Example 9.1.2

We now prove some standard properties of strong bisimulation using the new apparatus to get the reader accustomed with the notation and the way of proving.

A union of bisimulations is a bisimulation.

Theorem 9.1.3 Let $\{R_i\}_{i\in I}$ be strong bisimulations on $\langle \sigma, A, \rho \rangle \in \mathcal{TS}^n_{\mathsf{A}}$. Then $R = \sum_{i\in I} R_i$ is also a strong bisimulation on $\langle \sigma, A, \rho \rangle$. □

Proof Since $\sigma \leqslant \sigma R_i$, for all $i \in I$, by summing over all $i \in I$, we have $\sigma \leqslant \sigma R$. Next,

$$RA = (\sum_{i\in I} R_i)A = \sum_{i\in I} R_i A \leqslant \sum_{i\in I} AR_i = A(\sum_{i\in I} R_i) = AR$$

and

$$R\rho = (\sum_{i\in I} R_i)\rho = \sum_{i\in I} R_i\rho = \sum_{i\in I} \rho = \rho.$$

■

A state can always be related to itself in a bisimulation.

Theorem 9.1.4 If R is a strong bisimulation on $\langle\sigma, A, \rho\rangle \in \mathcal{TS}^n_{\mathsf{A}}$, then so is $R' = I + R$. □

Proof We have $\sigma \leqslant \sigma R \leqslant \sigma(I+R) = \sigma R'$. Also,

$$R'A = A + RA \leqslant A + AR = A(I+R) = AR'$$

and $R'\rho = R\rho + \rho \leqslant \rho + \rho = \rho$. ■

Transitive closure of a strong bisimulation is a strong bisimulation.

Theorem 9.1.5 If R be a strong bisimulation on $\langle\sigma, A, \rho\rangle \in \mathcal{TS}^n_{\mathsf{A}}$, then so is R^+. □

Proof Note that $R \leqslant R^+$ and so $\sigma \leqslant \sigma R \leqslant \sigma R^+$. We now prove, by induction on n, that $R^n A \leqslant AR^n$ and $R^n\rho \leqslant \rho$ for all $n \geqslant 1$. For $n = 1$, the statement holds by the definition of strong bisimulation. Suppose the statement holds for $n \geqslant 1$. Then

$$R^{n+1}A = RR^nA \leqslant RAR^n \leqslant ARR^n = AR^{n+1}$$

and $R^{n+1}\rho = RR^n\rho \leqslant R\rho \leqslant \rho$. Now,

$$R^+A = (\sum_{n=1}^{\infty} R^n)A = \sum_{n=1}^{\infty} R^nA \leqslant \sum_{n=1}^{\infty} AR^n = A\sum_{n=1}^{\infty} R^n = AR^+,$$

and similarly $R^+\rho \leqslant \rho$. ■

Strong bisimulation is preserved under isomorphism. Note that this property is vital; it says that the numbering we picked to represent the system in matrix terms is irrelevant.

Theorem 9.1.6 Let R be a strong bisimulation on $\langle\sigma, A, \rho\rangle \in \mathcal{TS}^n_{\mathsf{A}}$. Then, for any isomorphism P, the relation $R' = PRP^{\mathsf{T}}$ is a strong bisimulation on $\langle\sigma', A', \rho'\rangle \in \mathcal{TS}^n_{\mathsf{A}}$ where $\sigma' = \sigma P^{\mathsf{T}}$, $A' = PAP^{\mathsf{T}}$ and $\rho' = P\rho$. □

Proof First, we have

$$\sigma' = \sigma P^{\mathsf{T}} \leqslant \sigma RP^{\mathsf{T}} = \sigma P^{\mathsf{T}}PRP^{\mathsf{T}} = \sigma' R'.$$

Then,

$$R'A' = PRP^{\mathsf{T}}PAP^{\mathsf{T}} = PRAP^{\mathsf{T}} \leqslant PARP^{\mathsf{T}} = PAP^{\mathsf{T}}PRP^{\mathsf{T}} = A'R'$$

and finally $R'\rho' = PRP^{\mathsf{T}}P\rho = PR\rho \leqslant P\rho = \rho'$. ■

Note that if R is an equivalence relation, then R' is also an equivalence relation; a strong bisimulation equivalence is therefore also preserved under isomorphism.

9.2 Strong lumping

Lumping is a process of obtaining a smaller system from a bigger one by joining states that are bisimulation equivalent. In this section we define lumping using matrices. First we need to introduce the concept of a collector matrix.

A relation $V \in \mathbb{P}(\mathsf{A})^{n\times N}$, $n \geqslant N$ in which every row contains exactly one 1 is called a *collector*. Note that $V \cdot \mathbf{1} = \mathbf{1}$. A matrix $U \in \mathbb{P}(\mathsf{A})^{N\times n}$ such that $U \cdot \mathbf{1} = \mathbf{1}$ and $UV = I^N$ is a *distributor* for V. The matrix $W = V^\mathsf{T}$ is an example of a distributor for V; it is called the *maximal* distributor for V (no entry can be changed to 1 in W if it is to stay a distributor for V).

Example 9.2.1 The matrix $V = \begin{pmatrix} 1&0&0\\0&1&0\\0&1&0\\0&0&1\\0&0&1\end{pmatrix}$ is a collector. The matrix $U = \begin{pmatrix} 1&0&0&0&0\\0&\{a,b\}&1&0&0\\0&0&0&1&0\end{pmatrix}$, with $a, b \in \mathsf{A}$, is an example of a distributor for V, and $V^\mathsf{T} = \begin{pmatrix} 1&0&0&0&0\\0&1&1&0&0\\0&0&0&1&1\end{pmatrix}$ is the maximal one. □

The following theorem shows that any equivalence relation can be decomposed into a product of a collector and its transpose.

Theorem 9.2.2 Let $R \in \{0,1\}^{n\times n}$ be an equivalence relation. Then there exists a unique (modulo isomorphism) matrix $V \in \mathbb{P}(\mathsf{A})^{n\times N}$ such that $R = VV^\mathsf{T}$. Moreover, this V is a collector. □

Proof See [91]. ■

We give an example of an equivalence relation and the collector associated to it.

Example 9.2.3 Let $R = \begin{pmatrix} 1&0&0\\0&1&1\\0&1&1\end{pmatrix}$. It is easy to show that R is an equivalence relation. We obtain $V = \begin{pmatrix} 1&0\\0&1\\0&1\end{pmatrix}$.

Every block diagonal matrix with blocks consisting entirely of 1's is an equivalence relation. This example shows how the collector is obtained when a relation is in this form. This automatically gives the general method for obtaining the collector because it is not hard to show that every equivalence relation can be permuted into the mentioned block form. □

We can think of a collector matrix as a matrix in which the rows represent states, the columns represent the equivalence classes, and the entries indicate which states belong to which classes.

Any system can be reduced modulo an equivalence relation as follows. The states of the reduced system are the equivalence classes of the original system. The initial state is the class that contains the initial state of the original system. The set of terminating states consists of the classes that contain at least one terminating state. The reduced system performs a transition from one class to another if there is a state in the first class that performs the same transition to some state in the other class. In matrix terms, the reduction by some equivalence relation is formally given in the following definition.

Definition 9.2.4 (Reduction by an equivalence relation) Let $\langle \sigma, A, \rho \rangle \in \mathcal{TS}^n_{\mathsf{A}}$ be a transition system and let $R \in \{0,1\}^{n\times n}$ be some equivalence relation. If $R = VV^\mathsf{T}$ for a collector $V \in \{0,1\}^{n\times N}$, then $\langle \sigma, A, \rho \rangle$ reduces by R to the transition system $\langle \hat{\sigma}, \hat{A}, \hat{\rho} \rangle \in \mathcal{TS}^N_{\mathsf{A}}$ defined by:

$$\hat{\sigma} = \sigma V, \quad \hat{A} = V^\mathsf{T} A V \quad \text{and} \quad \hat{\rho} = V^\mathsf{T}\rho. \qquad \square$$

We are particulary interested in reduction modulo a strong bisimulation equivalence.

Definition 9.2.5 (Strong lumping) If $\langle \sigma, A, \rho \rangle \in \mathcal{TS}^n_{\mathsf{A}}$ reduces by an equivalence relation R to $\langle \hat{\sigma}, \hat{A}, \hat{\rho} \rangle \in \mathcal{TS}^N_{\mathsf{A}}$, and R is a strong bisimulation, we say that it *lumps (by R) to* $\langle \hat{\sigma}, \hat{A}, \hat{\rho} \rangle$. $\square$

We sometimes call $\langle \hat{\sigma}, \hat{A}, \hat{\rho} \rangle$ the *lumped* system assuming that R can be obtained from the context.

Example 9.2.6 Consider the transition system and the strong bisimulation from Example 9.1.2. By adding the identity matrix to R we get a strong bisimulation $R' = I + R$ that is an equivalence relation. This situation is depicted in Figure 9.2a. Since

$$R' = \begin{pmatrix} 1&0&0&0&0\\ 0&1&1&0&0\\ 0&1&1&0&0\\ 0&0&0&1&1\\ 0&0&0&1&1 \end{pmatrix} = \begin{pmatrix} 1&0&0\\ 0&1&0\\ 0&1&0\\ 0&0&1\\ 0&0&1 \end{pmatrix} \begin{pmatrix} 1&0&0&0&0\\ 0&1&1&0&0\\ 0&0&0&1&1 \end{pmatrix} = VV^\mathsf{T},$$

the system $\langle \sigma, A, \rho \rangle$ from Example 9.1.2 lumps by R' to

$$\hat{A} = V^\mathsf{T} A V = \begin{pmatrix} 0 & \{a\} & 0\\ 0 & 0 & \{b,c\}\\ 0 & \{d\} & 0 \end{pmatrix}, \quad \hat{\sigma} = \sigma V = (1,0,0), \quad \hat{\rho} = V^\mathsf{T}\rho = \begin{pmatrix} 0\\0\\1 \end{pmatrix}.$$

$\square$

The lumped system is depicted in Figure 9.2b.

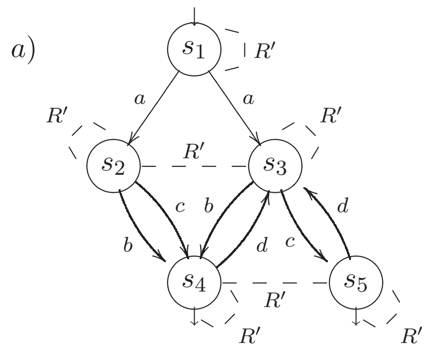


Figure 9.2: Transition system, strong lumping and the lumped system – Example 9.2.6

We now show that strong lumping coincides with the notion of ordinary lumping [67, 82] from Markov chain theory where it is more usual to define lumping conditions in terms of a collector and an arbitrary distributor associated to it. Given a bisimulation equivalence relation $R = VV^\mathsf{T}$, we have that $VV^\mathsf{T}A \leqslant AVV^\mathsf{T}$ and $VV^\mathsf{T}\rho \leqslant \rho$. Note that $\sigma \leqslant \sigma VV^\mathsf{T}$ holds trivially because $VV^\mathsf{T} \geqslant I$. We show that the first inequality is equivalent to $VV^\mathsf{T}AV \leqslant AV$. Note that $VV^\mathsf{T}AV \leqslant AV$ is implied by $VV^\mathsf{T}A \leqslant AVV^\mathsf{T}$ by multiplying both sides of the equality on the right by V. To prove the other implication, we have $VV^\mathsf{T}A \leqslant VV^\mathsf{T}AVV^\mathsf{T} \leqslant AVV^\mathsf{T}$. Because $VV^\mathsf{T} \geqslant I$ we also have $AV \leqslant VV^\mathsf{T}AV$ and $\rho \leqslant VV^\mathsf{T}\rho$, and so $VV^\mathsf{T}AV = AV$ and $VV^\mathsf{T}\rho = \rho$. These conditions do not depend on the particular choice of a distributor; in particular, it does not have to be the maximal one. Suppose that there is some U such that $UV = I$. Then $VUAV = VUVV^\mathsf{T}AV = VV^\mathsf{T}AV \leqslant AV$ and similarly $VU\rho = \rho$. This is exactly the same as the conditions for ordinary lumping of Markov reward chains proposed in [82]. Note that the definition of the lumped process $\langle\hat{\sigma}, \hat{A}, \hat{\rho}\rangle$ is also independent on the particular distributor for U. This is because $\hat{A} = V^\mathsf{T}AV = V^\mathsf{T}VUAV = UAV$ and $\hat{\rho} = V^\mathsf{T}\rho = V^\mathsf{T}VU\rho = U\rho$ for any distributor U.

9.3 Strong bisimulation between systems

The notion of bisimulation naturally extends to a relation between two systems.

Definition 9.3.1 (Strong bisimulation between systems) A relation $R \in \{0,1\}^{m\times n}$ is a *strong bisimulation between* the transition systems $\langle \sigma_A, A, \rho_A\rangle \in \mathcal{TS}^n_{\mathsf{A}}$ and $\langle \sigma_B, B, \rho_B\rangle \in \mathcal{TS}^m_{\mathsf{A}}$ if

$$RB \leqslant AR, \quad R\rho_B \leqslant \rho_A, \quad \text{and} \quad \sigma_B \leqslant \sigma_A R,$$

and symmetrically

$$R^{\mathsf{T}}A \leqslant BR^{\mathsf{T}}, \quad R^{\mathsf{T}}\rho_A \leqslant \rho_B, \quad \text{and} \quad \sigma_A \leqslant \sigma_B R^{\mathsf{T}}. \qquad \square$$

The following theorem shows that strong bisimulation between a system and itself induces a strong bisimulation on that system. We could have alternatively defined strong bisimulation on a system this way.

Theorem 9.3.2 Let $\langle \sigma, A, \rho\rangle \in \mathcal{TS}^n_{\mathsf{A}}$ be a matrix transition system. Let $R \in \{0,1\}^{n\times n}$ be a strong bisimulation between $\langle \sigma, A, \rho\rangle$ and $\langle \sigma, A, \rho\rangle$. Then $R' = R + R^{\mathsf{T}}$ is a strong bisimulation on $\langle \sigma, A, \rho\rangle$. □

Proof The relation R' is clearly symmetric. We only prove that it satisfies the first condition of Definition 9.1.1. We have $R'A = (R + R^{\mathsf{T}})A = RA + R^{\mathsf{T}}A \leqslant AR + AR^{\mathsf{T}} = A(R + R^{\mathsf{T}}) = AR'$ and $R'\rho = (R + R^{\mathsf{T}})\rho = R\rho + R^{\mathsf{T}}\rho \leqslant \rho + \rho = \rho$. ■

Strong bisimulation between two systems can also be defined via a strong bisimulation on a combined system.

Theorem 9.3.3 Let $\langle \sigma_A, A, \rho_A\rangle \in \mathcal{TS}^n_{\mathsf{A}}$ and $\langle \sigma_B, B, \rho_B\rangle \in \mathcal{TS}^m_{\mathsf{A}}$ be two matrix transition systems. A relation R is a strong bisimulation on between $\langle \sigma_A, A, \rho_A\rangle$ and $\langle \sigma_B, B, \rho_B\rangle$ if the symmetric relation

$$R' = \begin{pmatrix} 1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & R \\ \mathbf{0} & R^{\mathsf{T}} & \mathbf{0} \end{pmatrix}$$

is a strong bisimulation on the transition system $\langle \sigma, T, \rho\rangle$ defined by:

$$\sigma = \begin{pmatrix} 1 & \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad T = \begin{pmatrix} 1 & \sigma_A & \sigma_B \\ \mathbf{0} & A & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & B \end{pmatrix} \quad \text{and} \quad \rho = \begin{pmatrix} 1 \\ \rho_A \\ \rho_B \end{pmatrix}. \qquad \square$$

Proof We have

$$R'T = \begin{pmatrix} 1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & R \\ \mathbf{0} & R^{\mathsf{T}} & \mathbf{0} \end{pmatrix} \begin{pmatrix} 1 & \sigma_A & \sigma_B \\ \mathbf{0} & A & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & B \end{pmatrix} = \begin{pmatrix} 1 & \sigma_A & \sigma_B \\ \mathbf{0} & \mathbf{0} & RB \\ \mathbf{0} & R^{\mathsf{T}}A & \mathbf{0} \end{pmatrix},$$

$$TR' = \begin{pmatrix} 1 & \sigma_A & \sigma_B \\ \mathbf{0} & A & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & B \end{pmatrix} \begin{pmatrix} 1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & R \\ \mathbf{0} & R^{\mathsf{T}} & \mathbf{0} \end{pmatrix} = \begin{pmatrix} 1 & \sigma_B R^{\mathsf{T}} & \sigma_A R \\ \mathbf{0} & \mathbf{0} & AR \\ \mathbf{0} & BR^{\mathsf{T}} & \mathbf{0} \end{pmatrix}$$

and

$$R'\rho = \begin{pmatrix} 1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & R \\ \mathbf{0} & R^{\mathsf{T}} & \mathbf{0} \end{pmatrix} \begin{pmatrix} 1 \\ \rho_A \\ \rho_B \end{pmatrix} = \begin{pmatrix} 1 \\ R\rho_B \\ R^{\mathsf{T}}\rho_A \end{pmatrix}.$$

From $R'T \leqslant TR'$ we get $\sigma_A \leqslant \sigma_B R^{\mathsf{T}}$, $\sigma_B \leqslant \sigma_A R$, $RB \leqslant AR$ and $R^{\mathsf{T}}A \leqslant BR^{\mathsf{T}}$. From $R'\rho \leqslant \rho$ we get $\rho_B \leqslant \rho_A$ and $R^{\mathsf{T}}\rho_A \leqslant \rho_B$. These are the conditions of Definition 9.3.1. ■

The definition of a strong bisimulation between two systems allows us to establish another relation between the original and the lumped process.

Theorem 9.3.4 Suppose $\langle \sigma, A, \rho \rangle$ lumps by $R = VV^{\mathsf{T}}$ to $\langle \hat{\sigma}, \hat{A}, \hat{\rho} \rangle$. Then V is a strong bisimulation between A and $\hat{A}$. □

Proof Since $R = VV^{\mathsf{T}}$ is a bisimulation relation, we have $VV^{\mathsf{T}}A \leqslant AVV^{\mathsf{T}}$ and $VV^{\mathsf{T}}\rho \leqslant \rho$. Multiplying the first condition by V from the right we get $VV^{\mathsf{T}}AV \leqslant AV$ and thus $V\hat{A} \leqslant AV$. Moreover, since $\hat{\rho} = V^{\mathsf{T}}\rho$, it follows from $VV^{\mathsf{T}}\rho \leqslant \rho$ that $V\hat{\rho} \leqslant \rho$. We trivially have $\sigma V \leqslant \sigma V$ and so, V satisfies the first set of conditions in Definition 9.3.1. That it also satisfies the second set of conditions follows directly from $VV^{\mathsf{T}} \geqslant I$. ■

9.3.1 Compatibility with the operations

The following series of theorems show that strong bisimulation is compatible with all the operations on transition systems that we defined in Section 8.3. Note how the block matrix definitions of the operators give very compact and readable proofs.

Theorem 9.3.5 (Alternative composition) Let R_A be a strong bisimulation between $\langle \sigma_A, A, \rho_A \rangle \in \mathcal{TS}_{\mathsf{A}}^{n_A \times m_A}$ and $\langle \sigma'_A, A', \rho'_A \rangle \in \mathcal{TS}_{\mathsf{A}}^{n'_A \times m'_A}$. Let

R_B be a strong bisimulation between the transition systems $\langle \sigma_B, B, \rho_B \rangle \in \mathcal{TS}_{\mathsf{A}}^{n_B \times m_B}$ and $\langle \sigma'_B, B', \rho'_B \rangle \in \mathcal{TS}_{\mathsf{A}}^{n'_B \times m'_B}$. Then

$$R = \begin{pmatrix} 1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & R_A & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & R_B \end{pmatrix}$$

is a strong bisimulation between $\langle \sigma_A, A, \rho_A \rangle + \langle \sigma_B, B, \rho_B \rangle$ and $\langle \sigma'_A, A', \rho'_A \rangle + \langle \sigma'_B, B', \rho'_B \rangle$. □

Proof The relation R is symmetric because R_A and R_B are. We have

$$R \begin{pmatrix} 1 & \sigma'_A A' & \sigma'_B B' \\ \mathbf{0} & A' & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & B' \end{pmatrix} = \begin{pmatrix} 1 & \sigma'_A A' & \sigma'_B B' \\ \mathbf{0} & R_A A' & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & R_B B' \end{pmatrix} \leqslant \begin{pmatrix} 1 & \sigma_A R_A A' & \sigma_B R_B B' \\ \mathbf{0} & R_A A' & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & R_B B' \end{pmatrix}$$
$$\leqslant \begin{pmatrix} 1 & \sigma_A A R_A & \sigma_B B R_B \\ \mathbf{0} & A R_A & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & B R_B \end{pmatrix} = \begin{pmatrix} 1 & \sigma_A A & \sigma_B B \\ 0 & A & \mathbf{0} \\ 0 & \mathbf{0} & B \end{pmatrix} R$$

and

$$R \begin{pmatrix} \sigma'_A \rho'_A + \sigma'_B \rho'_B \\ \rho'_A \\ \rho'_B \end{pmatrix} = \begin{pmatrix} \sigma'_A \rho'_A + \sigma'_B \rho'_B \\ R_A \rho'_A \\ R_B \rho'_B \end{pmatrix} \leqslant$$
$$\leqslant \begin{pmatrix} \sigma_A R_A \rho'_A + \sigma_B R_B \rho'_B \\ R_A \rho'_A \\ R_B \rho'_B \end{pmatrix} \leqslant \begin{pmatrix} \sigma_A \rho_A + \sigma_B \rho_B \\ \rho_A \\ \rho_B \end{pmatrix}. \quad \blacksquare$$

Note also that $\begin{pmatrix} 1 & \mathbf{0} & \mathbf{0} \end{pmatrix} R = \begin{pmatrix} 1 & \mathbf{0} & \mathbf{0} \end{pmatrix}$.

Similarly for the symmetric case.

Theorem 9.3.6 (Sequential composition) Let R_A be a strong bisimulation between $\langle \sigma_A, A, \rho_A \rangle \in \mathcal{TS}_{\mathsf{A}}^{n_A \times m_A}$ and $\langle \sigma'_A, A', \rho'_A \rangle \in \mathcal{TS}_{\mathsf{A}}^{n'_A \times m'_A}$. Let R_B be a strong bisimulation between $\langle \sigma_B, B, \rho_B \rangle \in \mathcal{TS}_{\mathsf{A}}^{n_B \times m_B}$ and $\langle \sigma'_B, B', \rho'_B \rangle \in \mathcal{TS}_{\mathsf{A}}^{n'_B \times m'_B}$. Then

$$R = \begin{pmatrix} R_A & \mathbf{0} \\ \mathbf{0} & R_B \end{pmatrix}$$

is a strong bisimulation between $\langle \sigma_A, A, \rho_A \rangle \cdot \langle \sigma_B, B, \rho_B \rangle$ and $\langle \sigma'_A, A', \rho'_A \rangle \cdot \langle \sigma'_B, B', \rho'_B \rangle$. □

Proof The relation R is symmetric because R_A and R_B are. We have

$$R\begin{pmatrix} A' & \rho'_A\sigma'_B B' \\ \mathbf{0} & B' \end{pmatrix} = \begin{pmatrix} R_A A' & R_A\rho'_A\sigma'_B B' \\ \mathbf{0} & R_B B' \end{pmatrix} \leqslant \begin{pmatrix} R_A A' & R_A\rho'_A\sigma_B R_B B' \\ \mathbf{0} & R_B B' \end{pmatrix}$$
$$\leqslant \begin{pmatrix} AR_A & \rho_A\sigma_B B R_B \\ \mathbf{0} & BR_B \end{pmatrix} = \begin{pmatrix} A & \rho_A\sigma_B B \\ \mathbf{0} & B \end{pmatrix} R$$

and

$$R\begin{pmatrix} \rho'_A\sigma'_B\rho'_B \\ \rho'_B \end{pmatrix} = \begin{pmatrix} R_A\rho'_A\sigma'_B\rho'_B \\ R_B\rho'_B \end{pmatrix} \leqslant \begin{pmatrix} R_A\rho'_A\sigma_B R_B\rho'_B \\ R_B\rho'_B \end{pmatrix} \leqslant \begin{pmatrix} \rho_A\sigma_B\rho_B \\ \rho_B \end{pmatrix}. \qquad \blacksquare$$

Also $\begin{pmatrix} 1 & \mathbf{0} \end{pmatrix} R = \begin{pmatrix} 1 & \mathbf{0} \end{pmatrix}$.

Similarly for the symmetric case.

Theorem 9.3.7 (Repetition) Let R be a strong bisimulation between $\langle \sigma, A, \rho \rangle \in \mathcal{TS}_{\mathsf{A}}^{n\times m}$ and $\langle \sigma', A', \rho' \rangle \in \mathcal{TS}_{\mathsf{A}}^{n'\times m'}$. Then

$$R' = \begin{pmatrix} 1 & \mathbf{0} \\ \mathbf{0} & R \end{pmatrix}$$

is a strong bisimulation between $*\langle \sigma, A, \rho \rangle$ and $*\langle \sigma', A', \rho' \rangle$. $\square$

Proof The relation R' is symmetric because R is. We have

$$R'\begin{pmatrix} 0 & \sigma' A' \\ \mathbf{0} & A' + \rho'\sigma' A' \end{pmatrix} = \begin{pmatrix} 0 & \sigma' A' \\ \mathbf{0} & RA' + R\rho'\sigma' A' \end{pmatrix} \leqslant \begin{pmatrix} 0 & \sigma R A' \\ \mathbf{0} & AR + \rho\sigma R A' \end{pmatrix} \leqslant$$
$$\leqslant \begin{pmatrix} 0 & \sigma A R \\ \mathbf{0} & AR + \rho\sigma A R \end{pmatrix} = \begin{pmatrix} 0 & \sigma A \\ \mathbf{0} & A + \rho\sigma A \end{pmatrix} R'$$

and

$$R'\begin{pmatrix} 1 \\ \rho' \end{pmatrix} = \begin{pmatrix} 1 \\ R\rho' \end{pmatrix} \leqslant \begin{pmatrix} 1 \\ \rho \end{pmatrix}.$$

Also, $\begin{pmatrix} 1 & \mathbf{0} \end{pmatrix} R' = \begin{pmatrix} 1 & \mathbf{0} \end{pmatrix}$. Similarly for the symmetric case. $\blacksquare$

Parallel composition is the only operation that is not defined using block matrices. The proof, however, goes smoothly by applying the equalities for Kronecker product and Kronecker sum established in the preliminaries.

Theorem 9.3.8 (Parallel composition) Let R_A be a strong bisimulation between $\langle \sigma_A, A, \rho_A \rangle \in \mathcal{TS}_{\mathsf{A}}^{n_A \times m_A}$ and $\langle \sigma'_A, A', \rho'_A \rangle \in \mathcal{TS}_{\mathsf{A}}^{n'_A \times m'_A}$. Let R_B be a strong bisimulation between $\langle \sigma_B, B, \rho_B \rangle \in \mathcal{TS}_{\mathsf{A}}^{n_B \times m_B}$ and $\langle \sigma'_B, B', \rho'_B \rangle \in \mathcal{TS}_{\mathsf{A}}^{n'_B \times m'_B}$. Then $R = R_A \otimes R_B$ is a strong bisimulation between $\langle \sigma_A, A, \rho_A \rangle \parallel \langle \sigma_B, B, \rho_B \rangle$ and $\langle \sigma'_A, A', \rho'_A \rangle \parallel \langle \sigma'_B, B', \rho'_B \rangle$. □

Proof First, we have $(R_A \otimes R_B)^{\mathsf{T}} = R_A{}^{\mathsf{T}} \otimes R_B{}^{\mathsf{T}} = R_A \otimes R_B$ and hence $R_A \otimes R_B$ is symmetric.

Next,

$$
\begin{array}{cl}
 & (R_A \otimes R_B)\left(\bar{\Omega} \cdot (A' \oplus B') + \Omega \cdot (A' \otimes B')\right) \\
= & \bar{\Omega} \cdot (R_A \otimes R_B)(A' \oplus B') + \Omega \cdot (R_A \otimes R_B)(A' \otimes B') \\
= & \bar{\Omega} \cdot (R_A \otimes R_B)(A' \otimes I + I \otimes B') + \Omega \cdot (R_A \otimes R_B)(A' \otimes B') \\
= & \bar{\Omega} \cdot (R_A A' \otimes R_B) + \bar{\Omega} \cdot (R_A \otimes R_B B') + \Omega \cdot (R_A A' \otimes R_B B') \\
\leqslant & \bar{\Omega} \cdot (A R_A \otimes R_B) + \bar{\Omega} \cdot (R_A \otimes B R_B) + \Omega \cdot (A R_A \otimes B R_B) \\
= & \bar{\Omega} \cdot (A \otimes I)(R_A \otimes R_B) + \bar{\Omega} \cdot (I \otimes B)(R_A \otimes R_B) + \Omega \cdot (A \otimes B)(R_A \otimes R_B) \\
= & \bar{\Omega} \cdot (A \oplus B)(R_A \otimes R_B) + \Omega \cdot (A \otimes B)(R_A \otimes R_B) \\
= & \left(\bar{\Omega} \cdot (A \oplus B) + \Omega \cdot (A \otimes B)\right)(R_A \otimes R_B).
\end{array}
$$

Also,

$$(R_A \otimes R_B)(\rho'_A \otimes \rho'_B) = R_A \rho'_A \otimes R_B \rho'_B \leqslant \rho_A \otimes \rho_B$$

and

$$\sigma'_A \otimes \sigma'_B \leqslant \sigma_A R_A \otimes \sigma_B R_B = (\sigma_A \otimes \sigma_B)(R_A \otimes R_B).$$

Similarly for the symmetric case. ■

9.4 Backward bisimulation

Backward bisimulation as introduced in [75] is a dual to standard strong bisimulation. It requests that two related states have the same set of actions leading to them. In the setting with successful termination, it is natural to also require that they must have the same termination behavior too. The relation must be total to avoid the case when there is only one pair in the relation, namely the one consisting of the initial state and itself.

Definition 9.4.1 (Backward strong bisimulation) A symmetric and total relation $R \in \{0,1\}^{n \times n}$ is called a *backward strong bisimulation on* the matrix transition system $\langle \sigma, A, \rho \rangle \in \mathcal{TS}_{\mathsf{A}}^{n}$ if

$$\sigma R = \sigma, \quad R A^{\mathsf{T}} \leqslant A^{\mathsf{T}} R \quad \text{and} \quad R\rho \leqslant \rho. \qquad \square$$

This definition corresponds to the original definition of backward strong bisimulation because $\sigma R = \sigma$ says that the initial state is only related to itself, and $RA^{\mathsf{T}} \leqslant A^{\mathsf{T}} R$ says that:

$$\begin{array}{ccc} & & s_j \\ & & \big\downarrow{\scriptstyle a} \\ s_i & \overset{R}{- - -} & s_k \end{array} \qquad \text{implies} \qquad \begin{array}{ccc} s_\ell & \underset{R}{- - -} & s_j. \\ \big\downarrow{\scriptstyle a} & & \\ s_i & & \end{array}$$

The condition $R\rho \leqslant \rho$ is the same as in the standard strong bisimulation.

Note that an alternative definition is to say that R is a backward strong bisimulation on $\langle \sigma, A, \rho \rangle$ iff it is a total strong bisimulation on $\langle \sigma, A^{\mathsf{T}}, \rho \rangle$ that also satisfies $\sigma R \leqslant \sigma$. From this it follows that an arbitrary sum of backward strong bisimulations and the equivalence closure of a backward strong bisimulation is again a backward strong bisimulation.

Example 9.4.2 Consider the transition system and the relation R from Figure 9.3. We have

$$\sigma = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}, \quad A = \begin{pmatrix} 0 & \{a\} & \{a\} \\ \{b,c\} & 0 & 0 \\ \{b\} & 0 & 0 \end{pmatrix} \quad \text{and} \quad \rho = \mathbf{0}.$$

The relation

$$R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

is a backward strong bisimulation on $\langle \sigma, A, \rho \rangle$ because

$$\sigma R = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} = \sigma, \quad R\rho = \mathbf{0} = \rho \quad \text{and}$$

$$RA^{\mathsf{T}} = \begin{pmatrix} 0 & \{b,c\} & \{b\} \\ \{a\} & 0 & 0 \\ \{a\} & 0 & 0 \end{pmatrix} \leqslant \begin{pmatrix} 0 & \{b,c\} & \{b,c\} \\ \{a\} & 0 & 0 \\ \{a\} & 0 & 0 \end{pmatrix} = A^{\mathsf{T}} R.$$

Note that R is total because $RR^{\mathsf{T}} = R^2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \geqslant I$. □

To give more correspondence with the result of [75] we give a definition of backward strong bisimulation between systems. We can apply the technique

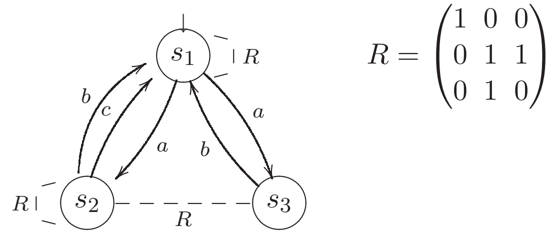


Figure 9.3: Transition system and a backward strong bisimulation on it – Example 9.4.2

of Theorem 9.3.3 and say that relation R is a backward strong bisimulation between $\langle \sigma_A, A, \rho_A \rangle$ and $\langle \sigma_B, B, \rho_B \rangle$ if

$$R' = \begin{pmatrix} 1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & R \\ \mathbf{0} & R^{\mathsf{T}} & \mathbf{0} \end{pmatrix}$$

is a backward strong bisimulation on the transition system

$$\langle \begin{pmatrix} 1 & \mathbf{0} & \mathbf{0} \end{pmatrix}, \begin{pmatrix} 1 & \sigma_A & \sigma_B \\ \mathbf{0} & A & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & B \end{pmatrix}, \begin{pmatrix} 1 \\ \rho_A \\ \rho_B \end{pmatrix} \rangle.$$

The requirement that R' is total implies that both R and R^{T} are total. Also, because R' satisfies the conditions of Definition 9.4.1, we obtain the following conditions for the backward strong bisimulation between systems:

$$RB^{\mathsf{T}} \leqslant A^{\mathsf{T}} R, \quad R\rho_B \leqslant \rho_A, \quad R{\sigma_B}^{\mathsf{T}} \leqslant {\sigma_A}^{\mathsf{T}}$$

and

$$R^{\mathsf{T}} A^{\mathsf{T}} \leqslant B^{\mathsf{T}} R^{\mathsf{T}}, \quad R^{\mathsf{T}} \rho_A \leqslant \rho_B, \quad R^{\mathsf{T}} {\sigma_A}^{\mathsf{T}} \leqslant {\sigma_B}^{\mathsf{T}}.$$

Note that the condition on the initial vector is different than in the case of strong bisimulation. It says that only the initial state of one system can be related to the initial state of the other system.

We now define backward strong lumping.

Definition 9.4.3 (Backward Strong lumping) If $\langle \sigma, A, \rho \rangle \in \mathcal{TS}^n_{\mathsf{A}}$ reduces to $\langle \hat{\sigma}, \hat{A}, \hat{\rho} \rangle \in \mathcal{TS}^N_{\mathsf{A}}$ by a backward strong bisimulation equivalence $R = VV^{\mathsf{T}}$ we say that it *backward lumps (by R) to $\langle \hat{\sigma}, \hat{A}, \hat{\rho} \rangle$.* □

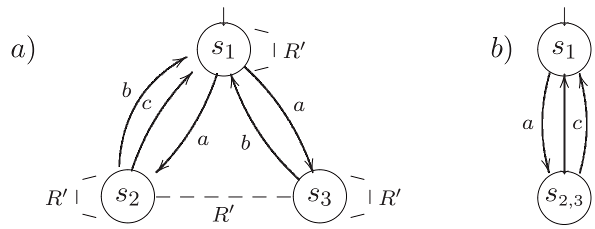


Figure 9.4: Backward strong lumping and the lumped system – Example 9.4.4

Example 9.4.4 Consider again the transition system and the backward strong bisimulation R from Example 9.4. By adding the identity matrix to R we obtain a backward strong bisimulation $R' = I+R$ that is an equivalence relation. This situation is depicted in Figure 9.4a. Since

$$R' = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix} = VV^\mathsf{T},$$

the system backward lumps by R' to

$$\hat{A} = V^\mathsf{T} A V = \begin{pmatrix} 0 & \{a\} \\ \{b, c\} & 0 \end{pmatrix}, \quad \hat{\sigma} = \sigma V = \begin{pmatrix} 1 & 0 \end{pmatrix}, \quad \hat{\rho} = V^\mathsf{T} \rho = \mathbf{0}.$$

The lumped system is depicted in Figure 9.4b. □

Definitions 9.4.3 and 9.4.1 give the following diagram:

$$\begin{array}{ccc} \langle \sigma, A, \rho \rangle & \xrightarrow{\textit{transpose}} & \langle \sigma, A^\mathsf{T}, \rho \rangle \\ \Big\downarrow {\scriptstyle \textit{backward strong lumping}} & & \Big\downarrow {\scriptstyle \textit{induced strong lumping}} \\ \langle \sigma V, V^\mathsf{T} A V, V^\mathsf{T} \rho \rangle & & \langle \sigma V, V^\mathsf{T} A^\mathsf{T} V, V^\mathsf{T} \rho \rangle. \end{array}$$

For the definition of backward strong lumping to be considered sound, we have to show that the diagram can be closed, i.e. that

$$\langle \sigma V, V^\mathsf{T} A V, V^\mathsf{T} \rho \rangle \xrightarrow{\textit{transpose}} \langle \sigma V, V^\mathsf{T} A^\mathsf{T} V, V^\mathsf{T} \rho \rangle\ .$$

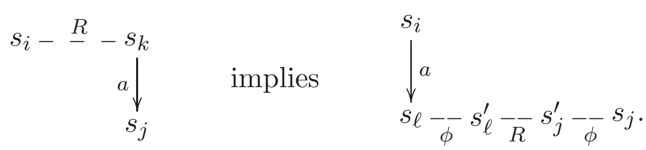
This, however, is trivial because $(V^\mathsf{T} A^\mathsf{T} V)^\mathsf{T} = V^\mathsf{T} (A^\mathsf{T})^\mathsf{T} V^{\mathsf{T}\mathsf{T}} = V^\mathsf{T} A V$.

9.5 Strong bisimulation up-to

Definition 9.5.1 (Strong bisimulation up-to) A symmetric relation $R \in \{0,1\}^{n\times n}$ is called a *strong bisimulation up-to* $\phi \in \{0,1\}^{n\times n}$ on the matrix transition system $\langle \sigma, A, \rho \rangle \in \mathcal{TS}^n_{\mathsf{A}}$ if

$$\sigma \leqslant \phi R \phi, \quad RA \leqslant A\phi R\phi, \quad \text{and} \quad R\rho \leqslant \rho. \qquad \square$$

Let us explain the definition using the standard terms. First, $\sigma \leqslant \phi R \phi$ means that the initial state does not need to be related to itself but rather that it is in relation ϕ with some k, ℓ that are related by R. The main idea of the second condition is that every action must be simulated but the resulting states are allowed to transform themselves by ϕ and then be related in R. Formally, the condition $RA \leqslant A\phi R$ means that:



The condition on the termination vector is the same as in the standard strong bisimulation.

Note that R is a strong bisimulation up-to the identity relation I iff it is a strong bisimulation. Also, if R is a transitive bisimulation up-to ϕ and $\phi \leqslant R$, then $RA \leqslant A\phi R\phi \leqslant ARRR \leqslant AR$ and so, R is a strong bisimulation on $\langle \sigma, A, \rho \rangle$.

We give an example.

Example 9.5.2 Consider the transition system and the relations R and ϕ from Figure 9.5. We have

$$\sigma = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}, \quad A = \begin{pmatrix} 0 & \{a\} & \{a\} & 0 \\ 0 & 0 & 0 & \{b\} \\ 0 & 0 & 0 & \{c\} \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \rho = \mathbf{0}.$$

We obtain

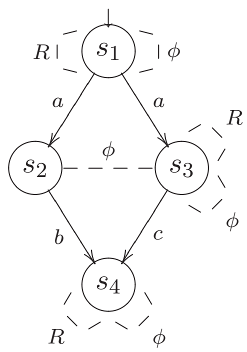
$$\phi R \phi = \begin{pmatrix} 1&0&0&0 \\ 0&0&1&0 \\ 0&1&1&0 \\ 0&0&0&1 \end{pmatrix} \begin{pmatrix} 1&0&0&0 \\ 0&0&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{pmatrix} \begin{pmatrix} 1&0&0&0 \\ 0&0&1&0 \\ 0&1&1&0 \\ 0&0&0&1 \end{pmatrix} = \begin{pmatrix} 1&0&0&0 \\ 0&1&1&0 \\ 0&1&1&0 \\ 0&0&0&1 \end{pmatrix}.$$

Now we show that R is a bisimulation up-to ϕ on $\langle \sigma, a, \rho \rangle$. We have

$$\sigma = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix} = \sigma\phi R\phi, \quad R\rho = \mathbf{0} = \rho \quad \text{and}$$

$$RA = \begin{pmatrix} 0 & \{a\} & \{a\} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \{b,c\} \\ 0 & 0 & 0 & 0 \end{pmatrix} \leqslant \begin{pmatrix} 0 & \{a\} & \{a\} & 0 \\ 0 & 0 & 0 & \{b\} \\ 0 & 0 & 0 & \{b,c\} \\ 0 & 0 & 0 & 0 \end{pmatrix} = A\phi R\phi.$$

□



$$R = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\phi = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Figure 9.5: Transition system and a bisimulation up-to on it – Example 9.5.2

The most common use of the up-to technique is when ϕ is also a bisimulation relation (in most cases the maximal one). The following theorem shows that in this case a strong bisimulation up-to ϕ can be transformed into a standard strong bisimulation on the same system.

Theorem 9.5.3 Let ϕ be a strong bisimulation equivalence on $\langle \sigma, A, \rho \rangle \in \mathcal{TS}^n_{\mathsf{A}}$. Then a symmetric relation R is a strong bisimulation up-to ϕ on $\langle \sigma, A, \rho \rangle$ iff $R' = \phi R \phi$ is a strong bisimulation on $\langle \sigma, A, \rho \rangle$. □

Proof Suppose first that R is a strong bisimulation up-to ϕ on $\langle \sigma, A, \rho \rangle$. Then $R'^{\mathsf{T}} = (\phi R \phi)^{\mathsf{T}} = \phi^{\mathsf{T}} R^{\mathsf{T}} \phi^{\mathsf{T}} = \phi R \phi = R'$, and so R' is symmetric. Also, we have

$$\begin{aligned} R'A = \phi R \phi A \leqslant \phi R A \phi \leqslant \phi A \phi R \phi \phi = \\ = \phi A \phi R \phi \leqslant A \phi \phi R \phi = A \phi R \phi = AR', \end{aligned}$$

and $R'\rho = \phi R \phi \rho \leqslant \phi R \rho \leqslant \phi \rho \leqslant \rho$.

Suppose now that $\phi R \phi$ is a strong bisimulation on $\langle \sigma, A, \rho \rangle$. Using that $I \leqslant \phi$ we have $RA \leqslant \phi R \phi A \leqslant A \phi R \phi$ and $R\rho \leqslant \phi R \phi \rho \leqslant \rho$. ■

Using this theorem we can define strong lumping up-to.

Definition 9.5.4 (Strong lumping up-to) Let R be a strong bisimulation up-to ϕ on $\langle \sigma, A, \rho \rangle \in \mathcal{TS}^n_{\mathsf{A}}$. If $\phi R \phi$ is an equivalence relation, and $\langle \sigma, A, \rho \rangle$ reduces by $\phi R \phi$ to $\langle \hat{\sigma}, \hat{A}, \hat{\rho} \rangle \in \mathcal{TS}^N_A$, we say that $\langle \sigma, A, \rho \rangle$ lumps by R up-to ϕ to $\langle \hat{\sigma}, \hat{A}, \hat{\rho} \rangle$. □

We give an example of a strong lumping up-to.

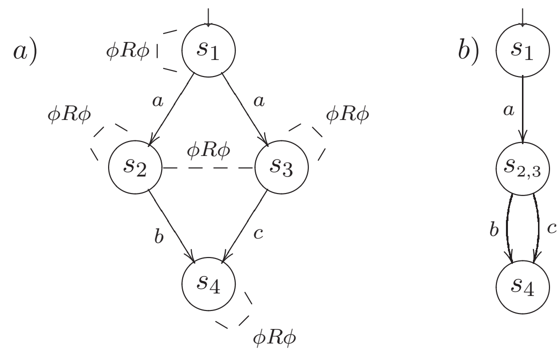


Figure 9.6: Lumping up-to and the lumped system – Example 9.5.5

Example 9.5.5 Let $\langle \sigma, A, \rho \rangle$ and the relations R and ϕ be as in Example 9.5.2. We have that

$$\phi R \phi = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = V V^{\mathsf{T}}$$

is an equivalence relation. The relation $\phi R \phi$ on $\langle \sigma, A, \rho \rangle$ is depicted in Figure 9.6a. Now, the system $\langle \sigma, A, \rho \rangle$ lumps by R up-to ϕ to

$$\hat{A} = V^{\mathsf{T}} A V = \begin{pmatrix} 0 & 0 & \{a\} \\ 0 & 0 & \{b, c\} \\ 0 & 0 & 0 \end{pmatrix}, \quad \hat{\sigma} = \sigma V = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}, \quad \hat{\rho} = V^{\mathsf{T}} \rho = \mathbf{0}.$$

The lumped system is depicted in Figure 9.6b. □

Chapter 10

Bisimulations on systems with silent steps

A silent step in a transition system is a step that is labeled by the internal action τ. In this chapter we give matrix definitions of two most popular bisimulation relations that abstract away from silent steps, that is of weak [79] and branching bisimulation [53].

Note that every matrix $T \in \mathbb{P}(\mathsf{A})^{n\times n}$ can be uniquely represented as $T = A + \{\tau\} \cdot S$ where $\tau \in \mathsf{A}$, and $A, S \in \mathbb{P}(\mathsf{A})^{n\times n}$ are such that $\{\tau\} \cdot A = \mathbf{0}$ and S is a 0–1 matrix. To make this form of T more explicit we write $\langle \sigma, A, S, \rho \rangle$ instead of $\langle \sigma, T, \rho \rangle$. Note that the conditions imposed on T in all bisimulation definitions from the previous chapter can be decomposed into separate conditions on A and S. For example, for the strong bisimulation, the condition $RT \leqslant TR$ is valid if and only if the inequalities $RA \leqslant AR$ and $RS \leqslant SR$ both hold.

10.1 Weak bisimulation

Weak bisimulation [79] ignores silent transitions in a very general way. It requests that a transition labeled with an action is simulated by a transition labeled with the same action but preceded and followed by a sequence of transitions labeled by τ. This allows for a simple matrix characterization using the known matrix definition of reflexive-transitive closure. It is known that weak bisimulation can be interpreted as a strong bisimulation on a system obtained by an operation we call τ-closure. We define this transformation in terms of matrices. We also introduce the notion of weak lumping, that is the reduction modulo weak bisimulation, and prove its soundness with respect to τ-closure. With soundness we mean the property that the τ-closure followed by a strong lumping is the same as the induced weak lumping followed by the τ-closure.

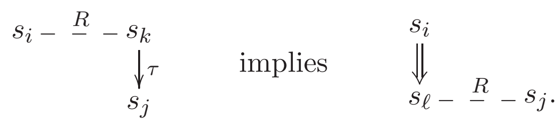
We now give a matrix definition of a weak bisimulation.

Definition 10.1.1 (Weak bisimulation on a system) A symmetric relation $R \in \mathbb{P}(\mathsf{A})^{n\times n}$ is a *weak bisimulation on* $\langle \sigma, A, S, \rho\rangle \in \mathcal{TS}^n_{\mathsf{A}}$ if

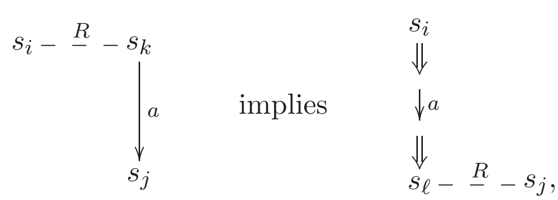
$$\sigma \leqslant \sigma R, \quad RS \leqslant S^* R, \quad RA \leqslant S^* A S^* R, \quad \text{and} \quad R\rho \leqslant S^*\rho. \qquad \square$$

Clearly, if there are no silent steps, then $S = \mathbf{0}$, and so $S^* = I$ and R is a strong bisimulation.

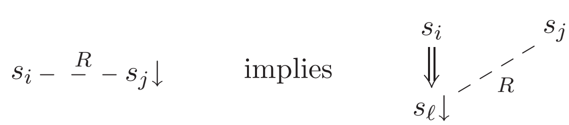
Our definition of weak bisimulation corresponds to the standard one. As for the strong bisimulation we require that the initial state must be related to itself. Note that $S^*[i,j] = 1$ iff there is an $n \geqslant 0$ such that $S^n[i,j] = 1$. Furthermore, this is equivalent to saying that there exist $i_0, \ldots, i_n$ such that $i_0 = i$, $i_n = j$ and $S[i_k, i_{k+1}] = 1$ for all $k = 0, \ldots, n-1$. Recall that $S[i,j] = 1$ means, in the standard theory, that $s_i \xrightarrow{\tau} s_j$. Thus, $S^*[i,j] = 1$ means that we have $s_{i_0} \xrightarrow{\tau} \ldots \xrightarrow{\tau} s_{i_n}$ or that, in the standard notation, $s_i \Rightarrow s_j$. Therefore, $RS \leqslant S^* R$ means that



As before, $a \in (RA)[i,j]$ iff there is a k such that $R[i,k] = 1$ and $a \in A[k,j]$. Now, $a \in (S^*AS^*R)[i,j]$ iff there exist $1 \leqslant \ell, \ell', \ell'' \leqslant n$ such that $S^*[i,\ell'] = 1$, $a \in A[\ell', \ell'']$, $S^*[\ell'', \ell] = 1$ and $R[\ell, j] = 1$. Therefore, $RA \leqslant S^*AS^*R$ means that



for $a \neq \tau$. Finally, $R\rho \leqslant S^*\rho$ means that



This is the standard definition of weak bisimulation.

Example 10.1.2 Consider the transition system and the relation R depicted in Figure 10.1. The transition system is defined by

$$\sigma = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad T = \begin{pmatrix} 0 & \{\tau\} & \{\tau\} & 0 & 0 \\ 0 & \{c\} & 0 & \{\tau\} & \{a,b\} \\ 0 & 0 & \{c\} & \{\tau\} & \{a\} \\ 0 & 0 & 0 & 0 & \{b\} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \rho = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}.$$

We have

$$S = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \text{ and } A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \{c\} & 0 & 0 & \{a,b\} \\ 0 & 0 & \{c\} & 0 & \{a\} \\ 0 & 0 & 0 & 0 & \{b\} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

From this

$$S^* = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \; S^*AS^* = \begin{pmatrix} 0 & \{c\} & \{c\} & \{c\} & \{a,b\} \\ 0 & \{c\} & 0 & \{c\} & \{a,b\} \\ 0 & 0 & \{c\} & \{c\} & \{a,b\} \\ 0 & 0 & 0 & 0 & \{b\} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \text{ and } S^*\rho = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}.$$

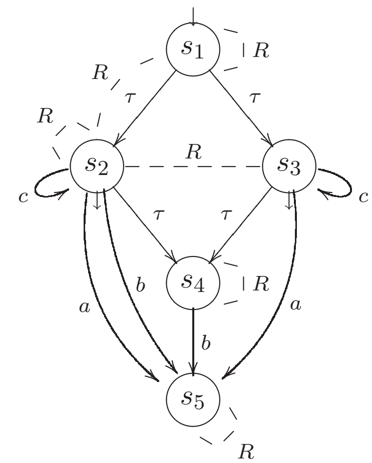
Now, since $R = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$, we have $\sigma R = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \end{pmatrix} \geqslant \sigma$,

$$R\rho = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = S^*\rho, \quad RS = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \leqslant \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} = S^*R$$

and

$$RA = \begin{pmatrix} 0 & \{c\} & 0 & 0 & \{a,b\} \\ 0 & \{c\} & \{c\} & 0 & \{a,b\} \\ 0 & \{c\} & 0 & 0 & \{a,b\} \\ 0 & 0 & 0 & 0 & \{b\} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \leqslant \begin{pmatrix} \{c\} & \{c\} & \{c\} & \{c\} & \{a,b\} \\ \{c\} & \{c\} & \{c\} & \{c\} & \{a,b\} \\ 0 & \{c\} & 0 & \{c\} & \{a,b\} \\ 0 & 0 & 0 & 0 & \{b\} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} = S^*AS^*R.$$

So, R is a weak bisimulation. □



$$R = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Figure 10.1: Transition system and a weak bisimulation on it – Example 10.1.2

10.1.1 Weak bisimulation as a strong bisimulation

We now prove that weak bisimulation can also be defined as a strong bisimulation on a transformed system. First we introduce the notion of τ-*closure*.

Definition 10.1.3 (τ-closure) Let $\langle \sigma, A, S, \rho \rangle \in \mathcal{TS}^n_{\mathsf{A}}$ be a transition system. The τ-*closure* of $\langle \sigma, A, S, \rho \rangle$ is the transition system $\langle \sigma, S^* A S^*, S^*, \rho \rangle \in \mathcal{TS}^n_{\mathsf{A}}$. □

We give an example.

Example 10.1.4 Consider again the transition system from Example 10.1.2, i.e. the one depicted in Figure 10.1. After τ-closing it becomes the transition system depicted in Figure 10.2. □

We prove that a relation is a weak bisimulation on a transition system iff it is a strong bisimulation on its τ-closure.

Theorem 10.1.5 A relation $R \in \mathbb{P}(\mathsf{A})^{n \times n}$ is a weak bisimulation on the transition system $\langle \sigma, A, S, \rho \rangle \in \mathcal{TS}^n_{\mathsf{A}}$ iff it is a strong bisimulation on the system $\langle \sigma, S^* A S^*, S^*, \rho \rangle \in \mathcal{TS}^n_{\mathsf{A}}$. □

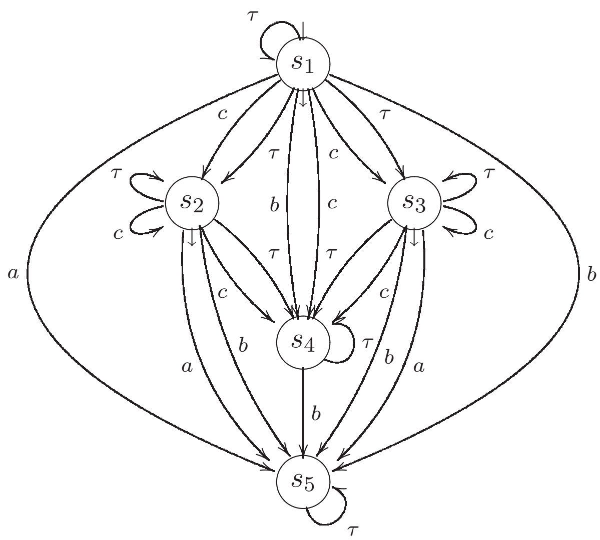


Figure 10.2: Transition system from Figure 10.1 after τ-closure

Proof Suppose R is a weak bisimulation on $\langle \sigma, A, S, \rho \rangle$. First we prove, by induction on n, that $RS^n \leqslant S^*R$. The base case follows from $I \leqslant S^*$. Suppose the statement holds for $n \geqslant 0$. Since $RS \leqslant S^*R$, we have

$$RS^{n+1} = RS^nS \leqslant S^*RS \leqslant S^*S^*R = S^*R.$$

Then,

$$RS^* = R\sum_{n=0}^{\infty} S^n = \sum_{n=0}^{\infty} RS^n \leqslant \sum_{n=0}^{\infty} S^*R = S^*R.$$

Using this and that $RA \leqslant S^*AS^*R$, we have

$$RS^*AS^* \leqslant S^*RAS^* \leqslant S^*S^*AS^*RS^* = \\ = S^*AS^*RS^* \leqslant S^*AS^*S^*R = S^*AS^*R.$$

Also, $RS^*\rho \leqslant S^*R\rho \leqslant S^*\rho$.

Suppose now that R is a strong bisimulation on $\langle \sigma, S^*AS^*, S^*, \rho \rangle$. Using that $I \leqslant S^*$, we have $RS \leqslant RS^* \leqslant S^*R$, $RA \leqslant RS^*AS^* \leqslant S^*AS^*R$ and $R\rho \leqslant RS^*\rho \leqslant S^*\rho$. ■

10.1.2 Weak lumping

We now introduce a notion of lumping that corresponds to weak bisimulation.

Definition 10.1.6 (Weak lumping) If $\langle \sigma, A, S, \rho \rangle \in \mathcal{TS}^n_{\mathsf{A}}$ reduces by a weak bisimulation equivalence $R = VV^{\mathsf{T}}$ to $\langle \hat{\sigma}, \hat{A}, \hat{S}, \hat{\rho} \rangle \in \mathcal{TS}^N_{\mathsf{A}}$, then we say that it *weakly lumps (by R) to $\langle \hat{\sigma}, \hat{A}, \hat{S}, \hat{\rho} \rangle$*. □

An example follows.

Example 10.1.7 Consider the transition system and the weak bisimulation from Example 10.1.2. By adding the identity matrix to R we get a weak bisimulation $R' = I + R$ that is also an equivalence relation. This situation is depicted in Figure 10.3a. Since

$$R' = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} = VV^{\mathsf{T}},$$

the system from Example 10.1.2 weakly lumps by R' to $\langle \hat{\sigma}, \hat{A}, \hat{\rho} \rangle$ defined by

$$\hat{\sigma} = \sigma V = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}, \quad \hat{A} = V^{\mathsf{T}} A V = \begin{pmatrix} 0 & 0 & \{a, b\} \\ 0 & 0 & \{b\} \\ 0 & 0 & 0 \end{pmatrix}, \quad \hat{\rho} = V^{\mathsf{T}} \rho = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

The weakly lumped system is depicted in Figure 10.3b. □

Definition 10.1.6 and Theorem 10.1.5 induce the following diagram:

$$\begin{array}{ccc} \text{Transition System} & \xrightarrow{\ \tau\text{-closure}\ } & \begin{array}{c}\tau\text{-closed} \\ \text{Transition System}\end{array} \\ \Big\downarrow \begin{array}{c}\textit{weak} \\ \textit{lumping}\end{array} & & \Big\downarrow \begin{array}{c}\textit{induced} \\ \textit{strong lumping}\end{array} \\ \begin{array}{c}\text{Weakly Lumped} \\ \text{Transition System}\end{array} & & \begin{array}{c}\text{Strongly Lumped} \\ \tau\text{-closed Transition System.}\end{array} \end{array}$$

We can claim that our theory is sound only if we can show that the above diagram can be closed, i.e. that also

$$\begin{array}{c}\text{Weakly Lumped} \\ \text{Transition System}\end{array} \xrightarrow{\ \tau\text{-closure}\ } \begin{array}{c}\text{Strongly Lumped} \\ \tau\text{-closed Transition System.}\end{array}$$

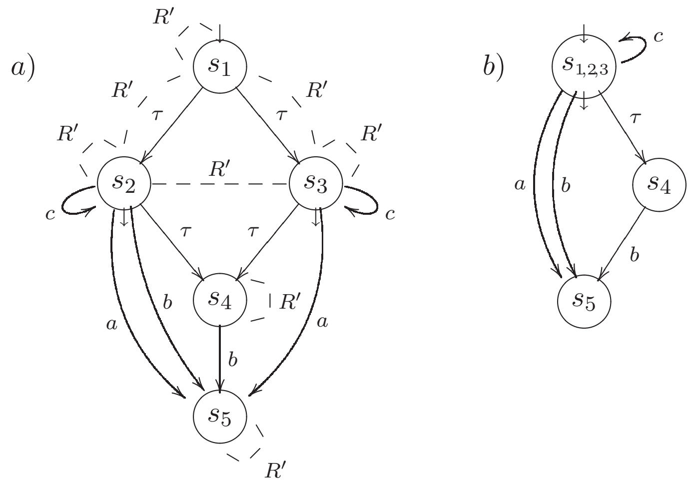


Figure 10.3: Transition system, weak lumping and the weakly lumped system – Example 10.1.7

Closing the diagram means that the order of application of τ-closure and lumping is irrelevant.

We prove a couple of lemmas first.

Lemma 10.1.8 For all $n \geqslant 0$, $V^{\mathsf{T}} S^n V \leqslant (V^{\mathsf{T}} S V)^n \leqslant V^{\mathsf{T}} S^* V$. □

Proof The proof is by induction on n. For $n = 0$, we have $V^{\mathsf{T}} S^0 V = V^{\mathsf{T}} I V = V^{\mathsf{T}} V = I = (V^{\mathsf{T}} S V)^0$ and $(V^{\mathsf{T}} S V)^0 = I = V^{\mathsf{T}} V \leqslant V^{\mathsf{T}} S^* V$. Suppose the lemma holds for $n \geqslant 0$. Then

$$V^{\mathsf{T}} S^{n+1} V = V^{\mathsf{T}} S S^n V \leqslant V^{\mathsf{T}} S V V^{\mathsf{T}} S^n V \leqslant V^{\mathsf{T}} S V (V^{\mathsf{T}} S V)^n \leqslant (V^{\mathsf{T}} S V)^{n+1}.$$

By Theorem 10.1.5, we have $V V^{\mathsf{T}} S^* \leqslant S^* V V^{\mathsf{T}}$ implying $V V^{\mathsf{T}} S^* V \leqslant S^* V$. Therefore,

$$(V^{\mathsf{T}} S V)^{n+1} = V^{\mathsf{T}} S V (V^{\mathsf{T}} S V)^n \leqslant V^{\mathsf{T}} S V V^{\mathsf{T}} S^* V = V^{\mathsf{T}} S S^* V \leqslant V^{\mathsf{T}} S^* V.$$

■

Lemma 10.1.9 $S^* V V^{\mathsf{T}} = S^* V V^{\mathsf{T}} S^*$. □

Proof From $I \leqslant S^*$, we have $S^* V V^{\mathsf{T}} = S^* V V^{\mathsf{T}} I \leqslant S^* V V^{\mathsf{T}} S^*$. Since $V V^{\mathsf{T}}$ is a weak bisimulation relation, by Theorem 10.1.5, $V V^{\mathsf{T}} S^* \leqslant S^* V V^{\mathsf{T}}$. Multiplying by S^* from the left we get $S^* V V^{\mathsf{T}} S^* \leqslant S^* S^* V V^{\mathsf{T}} = S^* V V^{\mathsf{T}}$.■

We now prove the soundness theorem.

Theorem 10.1.10 If $R = VV^\top$ is a weak bisimulation equivalence on $\langle\sigma, A, \rho\rangle$, then $(V^\top SV)^* = V^\top S^* V$, $(V^\top SV)^* V^\top AV(V^\top SV)^* = V^\top S^* AS^* V$ and $V^\top S^* \rho = (V^\top SV)^* V^\top \rho$. □

Proof By Lemma 10.1.8 we have $V^\top S^n V \leqslant (V^\top SV)^n \leqslant V^\top S^* V$ all $n \geqslant 0$. Summing over all n we get $V^\top S^* V \leqslant (V^\top SV)^* \leqslant V^\top S^* V$ and hence $(V^\top SV)^* = V^\top S^* V$.

By Theorem 10.1.5 we have that $VV^\top S^* \leqslant S^* VV^\top$, $VV^\top S^* AS^* \leqslant S^* AS^* VV^\top$ and $VV^\top S^* \rho \leqslant S^* \rho$. These conditions imply (multiplying by V from the right and using that $I \leqslant VV^\top$) that $VV^\top S^* V = S^* V$, $VV^\top S^* AS^* V = S^* AS^* V$ and $VV^\top S^* \rho = S^* \rho$. Now, using Lemma 10.1.9 and the equality $(V^\top SV)^* = V^\top S^* V$ proven above, we have

$$\begin{aligned}
&(V^\top SV)^* V^\top AV(V^\top SV)^* = V^\top S^* VV^\top AVV^\top S^* V = \\
&= V^\top S^* VV^\top AS^* V = V^\top S^* VV^\top S^* AS^* V = \\
&\qquad\qquad\qquad = V^\top S^* S^* AS^* V = V^\top S^* AS^* V
\end{aligned}$$

and

$$V^\top S^* \rho = V^\top S^* S^* \rho = V^\top S^* VV^\top S^* \rho = V^\top S^* VV^\top \rho = (V^\top SV)^* V^\top \rho. \quad \blacksquare$$

10.2 Branching bisimulation

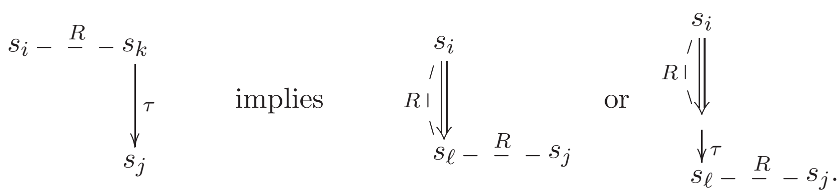
Branching bisimulation [53] preserves the branching structure of a system more than weak bisimulation by requiring that after the initial sequence of τ steps the resulting state must again be bisimilar to the same state that the starting state is bisimilar to. As we will see later the matrix approach fails here because it cannot express this property directly unless we require transitivity. The way out is to use a similar relation (not so uncommon, see [35]) and prove that it is equivalent to the standard definition of branching bisimulation when it is transitive.

We first give our definition of branching bisimulation. Note that the operation $\sqcap$ becomes central. Then, we express branching bisimulation as a strong bisimulation on a system closed under the sequence of τ-transitions that connect related states (note that the closure now depends on the bisimulation). Just as we did for weak, we introduce a notion of branching lumping and prove its soundness by showing that the corresponding diagram commutes. At the end of this chapter we show that the stuttering property has a nice matrix definition and we prove that branching bisimulation satisfies it.

Definition 10.2.1 (Branching bisimulation) A symmetric relation $R \in \mathbb{P}(\mathsf{A})^{n\times n}$ is called a *branching bisimulation on* $\langle \sigma, A, S, \rho\rangle \in \mathcal{TS}^n_{\mathsf{A}}$ iff:

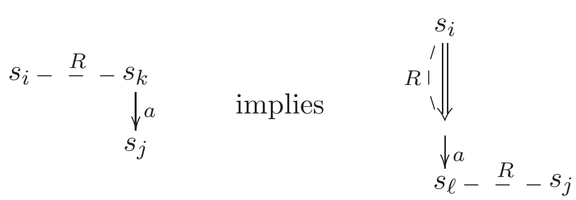
$$\sigma \leqslant \sigma R,\ RS \leqslant (S^* \sqcap R)(I{+}S)R,\ RA \leqslant (S^* \sqcap R)AR \text{ and } R\rho \leqslant (S^* \sqcap R)\rho. \ \square$$

We now explain these conditions. As before, the initial state must be related to itself. Note that $((S^* \sqcap R)(I+S)R)[i,j] = 1$ iff there exist k, ℓ such that $(S^* \sqcap R)[i,k] = 1$, $(I+S)[k,\ell] = 1$ and $R[\ell,j] = 1$. The first equality is equivalent to $S^*[i,k] = R[i,k] = 1$; the second means that either $k = l$ or $S[k,\ell] = 1$. So, $RS \leqslant (S^* \sqcap R)(I+S)R$ says that

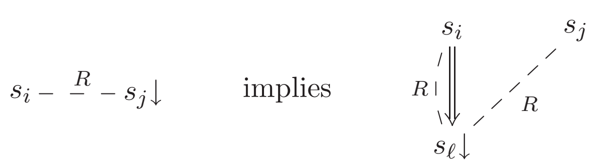


Here is where our definition does not match the standard one. The standard definition requires that the end state of the transition $\Rightarrow$ is related to s_k and not to state s_i. Of course, this is equivalent if R is transitive but, in general, it is not. The matrix approach we followed so far fails here, and we cannot obtain the desired definition directly. The reason is that in a matrix equation of the form $RX \leqslant YR$ the index k "appears" only on the left side and cannot be referred to from the right side.

We now explain the other two conditions. The condition $RA \leqslant (S^* \sqcap R)AR$ expresses the following:



Finally, the last condition, $R\rho \leqslant (S^* \sqcap R)\rho$, means that



Note that, because $(S^* \sqcap R) \leqslant S^* = S^*(I+S)$, every branching bisimulation equivalence is also a weak bisimulation.

Example 10.2.2 a. Consider the labeled transition system and the relation R from Example 10.1.2. We have

$$S^* \sqcap R = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

and then

$$RS = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \nleqslant \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} = (S^* \sqcap R)(I+S)R.$$

We conclude that R is not a branching bisimulation.

b. The problem in the previous example can be solved if we take

$$R^+ = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

instead of R. We have

$$S^* \sqcap R^+ = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

and

$$R^+S = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \leqslant \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} = (S^* \sqcap R^+)(I+S)R^+.$$

However,

$$R^+A = \begin{pmatrix} 0 & \{c\} & \{c\} & 0 & \{a,b\} \\ 0 & \{c\} & \{c\} & 0 & \{a,b\} \\ 0 & \{c\} & \{c\} & 0 & \{a,b\} \\ 0 & 0 & 0 & 0 & \{b\} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \not\leqslant$$

$$\not\leqslant \begin{pmatrix} \{c\} & \{c\} & \{c\} & 0 & \{a,b\} \\ \{c\} & \{c\} & \{c\} & 0 & \{a,b\} \\ \{c\} & \{c\} & \{c\} & 0 & \{a\} \\ 0 & 0 & 0 & 0 & \{b\} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} = (S^* \sqcap R^+)AR^+.$$

So, R^+ is also not a branching bisimulation.

c. Consider now the transition system depicted in Figure 10.4. This system is the same as the one from Example 10.1.2, i.e. the one from Figure 10.1.2, but it can additionally do the action a when going from state s_3 to s_5. We show that for this system the relation R^+ is a branching bisimulation.

The transition matrix is now

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \{c\} & 0 & 0 & \{a,b\} \\ 0 & 0 & \{c\} & 0 & \{a,b\} \\ 0 & 0 & 0 & 0 & \{b\} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

and σ and ρ are as before. Now,

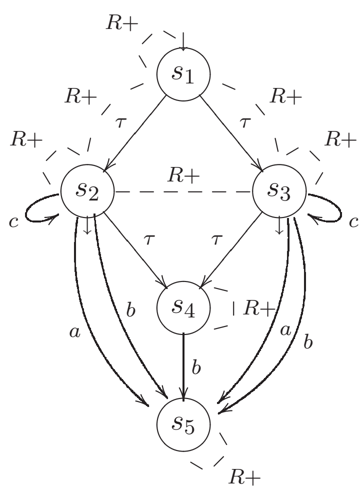
$$R^+A = \begin{pmatrix} 0 & \{c\} & \{c\} & 0 & \{a,b\} \\ 0 & \{c\} & \{c\} & 0 & \{a,b\} \\ 0 & \{c\} & \{c\} & 0 & \{a,b\} \\ 0 & 0 & 0 & 0 & \{b\} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \leqslant$$

$$\leqslant \begin{pmatrix} \{c\} & \{c\} & \{c\} & 0 & \{a,b\} \\ \{c\} & \{c\} & \{c\} & 0 & \{a,b\} \\ \{c\} & \{c\} & \{c\} & 0 & \{a,b\} \\ 0 & 0 & 0 & 0 & \{b\} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} = (S^* \sqcap R^+)AR^+.$$

The conditions on the initial and the terminating vector also hold:

$$\sigma \leqslant \sigma R \leqslant \sigma R^+ \quad \text{and} \quad R^+\rho = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = (S^* \sqcap R)\rho.$$

We conclude that R^+ is a branching bisimulation. □



$$R = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R^+ = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Figure 10.4: Transition system and a branching bisimulation on it – Example 10.2.2b

The following two theorems show that branching bisimulation is closed under arbitrary union and under reflexive-transitive closure. This directly implies that our definition of branching bisimulation corresponds to the standard one.

First we prove that a union of branching bisimulations is a branching bisimulation.

Theorem 10.2.3 Let $\{R_i\}_{i\in\mathcal{I}}$ be branching bisimulations on $\langle\sigma, A, S, \rho\rangle \in \mathcal{TS}^n_{\mathsf{A}}$. Then $R = \sum_{i\in\mathcal{I}} R_i$ is also a branching bisimulation on $\langle\sigma, A, S, \rho\rangle$. □

Proof Clearly, for some $i \in \mathcal{I}$, $\sigma \leqslant \sigma R_i \leqslant \sigma R$. We have

$$RS = (\sum_{i\in\mathcal{I}} R_i)S = \sum_{i\in\mathcal{I}} R_i S \leqslant \sum_{i\in\mathcal{I}} (S^* \sqcap R_i)(I+S)R_i \leqslant$$
$$\leqslant \sum_{i\in\mathcal{I}} (S^* \sqcap R_i)(I+S)R = (S^* \sqcap \sum_{i\in\mathcal{I}} R_i)(I+S)R = (S^* \sqcap R)(I+S)R$$

and

$$RA = \sum_{i\in\mathcal{I}} R_i A \leqslant \sum_{i\in\mathcal{I}} (S^* \sqcap R_i)AR_i \leqslant \sum_{i\in\mathcal{I}} (S^* \sqcap R_i)AR = (S^* \sqcap R)AR.$$

Also,

$$R\rho = (\sum_{i\in\mathcal{I}} R_i)\rho = \sum_{i\in\mathcal{I}} (S^* \sqcap R_i)\rho = (S^* \sqcap R)\rho.$$

■

We now show that the reflexive-transitive closure of a branching bisimulation is a branching bisimulation. The proof illustrates the usefulness of the Dedekind formula when working with the operator $\sqcap$.

Theorem 10.2.4 If R is a branching bisimulation on $\langle \sigma, A, S, \rho \rangle \in \mathcal{TS}^n_{\mathsf{A}}$, then so is R^*. □

Proof We have proven before that for any weak (and therefore also every branching) bisimulation R we have $RS^* \leqslant S^*R$. Using this we have

$$R(S^* \sqcap R^*) \leqslant RS^* \sqcap RR^* = RS^* \sqcap R^+ \leqslant S^*R \sqcap R^*.$$

Applying the Dedekind formula and using that $R^T = R$ we then have

$$S^*R \sqcap R^* \leqslant (S^* \sqcap R^*R)(R \sqcap S^{*\mathsf{T}}R^*) = (S^* \sqcap R^+)(R \sqcap S^{*\mathsf{T}}R^*) \leqslant (S^* \sqcap R^*)R.$$

We prove, by induction on n, that $R^n S \leqslant (S^* \sqcap R^*)(I+S)R^*$. The case when $n = 0$ follows trivially because $S^* \geqslant I$ and $R^* \geqslant I$. Suppose that the statement holds for $n \geqslant 0$. Then

$$R^{n+1}S = RR^nS \leqslant R(S^* \sqcap R^*)(I+S)R^* \leqslant$$
$$\leqslant (S^* \sqcap R^*)R(I+S)R^* = (S^* \sqcap R^*)R^* + (S^* \sqcap R^*)RSR^* \leqslant$$
$$\leqslant (S^* \sqcap R^*)R^* + (S^* \sqcap R^*)(S^* \sqcap R)(I+S)R^* \leqslant$$
$$\leqslant (S^* \sqcap R^*)R^* + (S^* \sqcap R^+)(I+S)R^* \leqslant$$
$$\leqslant (S^* \sqcap R^*)R^* + (S^* \sqcap R^*)(I+S)R^* = (S^* \sqcap R^*)(I+S)R^*.$$

Similarly, we prove that $R^n A \leqslant (S^* \sqcap R^*)AR^*$ and $R^n\rho \leqslant (S^* \sqcap R^*)\rho$. Now, by summing over all $n \geqslant 0$, we obtain that R^* is also a branching bisimulation. ■

10.2.1 Branching bisimulation as a strong bisimulation

As we did for weak, we prove that branching bisimulation can also be defined as a strong bisimulation on a transformed system. In the case of weak bisimulation the transformed system was obtained by τ-closure. Note that τ-closure is independent of the weak bisimulation. For the branching bisimulation we need to define a similar transformation but that also depends on the bisimulation relation itself. We call it τ, R-closure.

Definition 10.2.5 (τ, R-closure) Let $\langle \sigma, A, S, \rho \rangle \in \mathcal{TS}^n_{\mathsf{A}}$ be a transition system and let $R \in \{0,1\}$ be a relation. The τ, R-*closure* of $\langle \sigma, A, S, \rho \rangle$ is the transition system $\langle \sigma, (S^* \sqcap R)A, S^* \sqcap R, (S^* \sqcap R)\rho \rangle \in \mathcal{TS}^n_{\mathsf{A}}$. □

We prove that R is a branching bisimulation iff it is a strong bisimulation on the system obtained by τ, R-closure. Note that we require that R is an equivalence.

Theorem 10.2.6 A relation $R \in \mathbb{P}(\mathsf{A})^{n \times n}$ is a branching bisimulation equivalence on $\langle \sigma, A, S, \rho \rangle \in \mathcal{TS}^n_{\mathsf{A}}$ iff it is a strong bisimulation equivalence on $\langle \sigma, (S^* \sqcap R)A, S^* \sqcap R, (S^* \sqcap R)\rho \rangle \in \mathcal{TS}^n_{\mathsf{A}}$. □

Proof Suppose $R \in \mathbb{P}(\mathsf{A})^{n \times n}$ is a branching bisimulation equivalence on $\langle \sigma, A, S, \rho \rangle \in \mathcal{TS}^n_{\mathsf{A}}$. Then

$$\begin{aligned} R(S^* \sqcap R)(I+S) \leqslant RR(I+S) = R + RS \leqslant \\ \leqslant R + (S^* \sqcap R)(I+S)R = (S^* \sqcap R)(I+S)R. \end{aligned}$$

We also have

$$R(S^* \sqcap R)A \leqslant RRA = RA \leqslant (S^* \sqcap R)AR$$

and $R(S^* \sqcap R)\rho \leqslant R\rho \leqslant (S^* \sqcap R)\rho$.

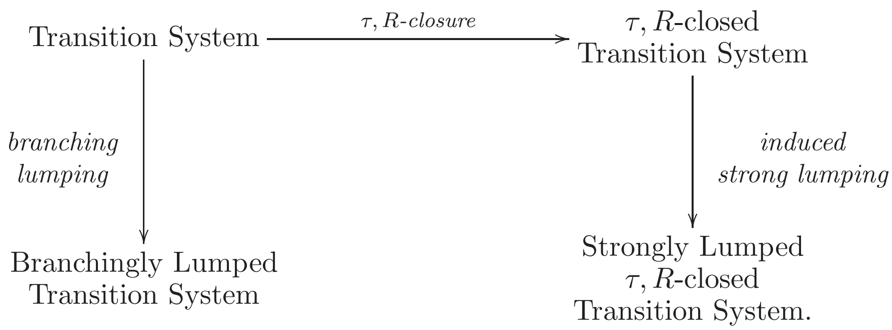
Suppose now that R is a strong bisimulation on the transition system $\langle \sigma, (S^* \sqcap R)A, S^* \sqcap R, (S^* \sqcap R)\rho \rangle \in \mathcal{TS}^n_{\mathsf{A}}$. Then the statement follows directly from $(S^* \sqcap R) \geqslant I$. ■

10.2.2 Branching lumping

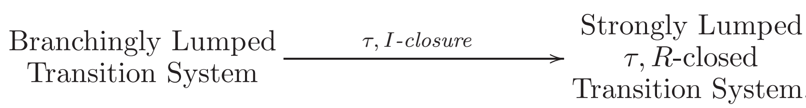
The idea of lumping extends to branching bisimulation naturally.

Definition 10.2.7 (Branching lumping) If $\langle \sigma, A, S, \rho \rangle \in \mathcal{TS}^n_{\mathsf{A}}$ reduces to $\langle \hat{\sigma}, \hat{A}, \hat{S}, \hat{\rho} \rangle \in \mathcal{TS}^N_{\mathsf{A}}$ by a branching bisimulation equivalence $R = VV^{\mathsf{T}}$ we say that it *branching lumps (by R) to* $\langle \hat{\sigma}, \hat{A}, \hat{S}, \hat{\rho} \rangle$. □

Definition 10.2.7 and Theorem 10.2.6 induce the following diagram:



Similarly as we had for weak bisimulation, for the theory to be considered sound we need to close the diagram. Note that $V^\top RV = V^\top VV^\top V = I$ and so we need to show that



Since $(V^\top SV)^* \sqcap I = I$, we only need to show the following:

Theorem 10.2.8 Let V, R, S, A, and ρ be as in Definition 10.2.7. Then $I + V^\top SV = V^\top(S^* \sqcap R)(I + S)V$, $V^\top AV = V^\top(S^* \sqcap R)AV$ and $V^\top\rho = V^\top(S^* \sqcap R)\rho$. □

Proof We have

$$I + V^\top SV = V^\top(I + S)V \leqslant V^\top(S^* \sqcap R)(I + S)V$$

and

$$\begin{aligned} V^\top(S^* \sqcap R)(I + S)V &= V^\top(S^* \sqcap VV^\top)V + V^\top(S^* \sqcap VV^\top)SV \leqslant \\ &\leqslant V^\top S^*V \sqcap I + V^\top S^*V \sqcap V^\top SV = I + V^\top SV. \end{aligned}$$

Clearly, $V^\top AV \leqslant V^\top(S^* \sqcap R)AV$. Also,

$$V^\top(S^* \sqcap R)AV \leqslant V^\top S^*AV \sqcap V^\top RAV = V^\top S^*AV \sqcap V^\top AV = V^\top AV.$$

Finally, we obtain $V^\top\rho \leqslant V^\top(S^* \sqcap R)\rho$ and

$$V^\top(S^* \sqcap R)\rho \leqslant V^\top S^*\rho \sqcap V^\top R\rho = V^\top S^*\rho \sqcap V^\top\rho = V^\top\rho.$$ ■

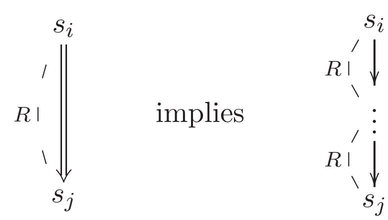
10.2.3 Stuttering property

Stuttering property is a very important property of branching bisimulation. For example, it allows it to be identified with stuttering equivalence which then induces its modal characterization. The property has a nice matrix definition.

Definition 10.2.9 (Stuttering property) A relation $R \in \{0,1\}^{n\times n}$ satisfies the stuttering property in the transition system $\langle \sigma, A, S, \rho\rangle \in \mathcal{TS}^n_{\mathsf{A}}$ iff the following holds:

$$S^* \sqcap R \leqslant (S \sqcap R)^*. \qquad \square$$

In standard terms the stuttering property denotes that:



Note that the inverse of the stuttering property always holds because $(S\sqcap R)^* = \sum_{n=0}^{\infty}(S\sqcap R)^n \leqslant \sum_{n=0}^{\infty}(S^n\sqcap R^n) = \sum_{n=0}^{\infty} S^n \sqcap \sum_{n=0}^{\infty} R^n = S^*\sqcap R$.

Example 10.2.10 Let $\langle \sigma, A, S, \rho\rangle \in \mathcal{TS}^n_{\mathsf{A}}$ be a transition system with

$$S = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}.$$

We obtain

$$S^* = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

Define

$$R_1 = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \text{ and } R_2 = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

The relation R_1 does not satisfy the stuttering property because

$$S^* \sqcap R_1 = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \not\leqslant \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = (S \sqcap R_1)^*.$$

The relation R_2 however does satisfy the property. We have

$$S^* \sqcap R_2 = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \leqslant \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} = (S \sqcap R_2)^*.$$

□

Similarly as we did in Part I we define the notion of stuttering closure as an extended relation that satisfies the stuttering property.

Definition 10.2.11 (Stuttering closure) The stuttering closure of a relation R by S is a relation $\mathrm{stt}_S(R)$ defied by

$$\mathrm{stt}_S(R) = S^{*\mathsf{T}} R S^{*\mathsf{T}}.$$

□

In standard terms the stuttering closure is illustrated by:

$$\text{if } \; s_i \overset{}{\underset{R}{- - -}} s_j, \qquad \text{then} \qquad \begin{array}{ccc} s_i & & s_j \\ \downarrow & & \uparrow \\ s_k & \underset{\mathrm{stt}_S(R)}{- - \to} & s_\ell. \end{array}$$

We give an example.

Example 10.2.12 Let S and R_1 be as in Example 10.2.10. The stuttering closure of R_1 is calculated as follows:

$$R = \mathrm{stt}_S(R_1) = S^{*\mathsf{T}} R_1 S^{*\mathsf{T}} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

It is clear that R satisfies the stuttering property. □

The following theorem shows that the stuttering closure of a relation always satisfies the stuttering property.

Theorem 10.2.13 For any relation $R \in \{0,1\}^{n \times n}$, $\mathrm{stt}_S(R)$ satisfies the stuttering property in any transition system $\langle \sigma, A, S, \rho \rangle \in \mathcal{TS}^n_{\mathsf{A}}$. □

Proof We first prove that $S^n \sqcap S^{*\mathsf{T}} R S^{*\mathsf{T}} \leqslant (S \sqcap S^{*\mathsf{T}} R S^{*\mathsf{T}})^n$ for all $n \geqslant 0$. The base case is trivial so suppose the statement holds for $n \geqslant 0$. By the Dedekind formula, we have

$$\begin{aligned} S^{n+1} \sqcap S^{*\mathsf{T}} R S^{*\mathsf{T}} &= S^n S \sqcap S^{*\mathsf{T}} R S^{*\mathsf{T}} \leqslant \\ &\leqslant (S^n \sqcap S^{*\mathsf{T}} R S^{*\mathsf{T}} S^{\mathsf{T}})(S \sqcap S^{n\mathsf{T}} S^{*\mathsf{T}} R S^{*\mathsf{T}}). \end{aligned}$$

Since $S^{*\mathsf{T}}S^{\mathsf{T}} = (SS^*)^{\mathsf{T}} \leqslant S^{*\mathsf{T}}$ and $S^{n\mathsf{T}}S^{*\mathsf{T}} = S^*S^{n\mathsf{T}} \leqslant S^{*\mathsf{T}}$, we have

$$(S^n \sqcap S^{*\mathsf{T}}RS^{*\mathsf{T}}S^{\mathsf{T}})(S \sqcap S^{n\mathsf{T}}S^{*\mathsf{T}}RS^{*\mathsf{T}}) \leqslant (S^n \sqcap S^{*\mathsf{T}}RS^{*\mathsf{T}})(S \sqcap S^{*\mathsf{T}}RS^{*\mathsf{T}}) \leqslant \\ \leqslant (S \sqcap S^{*\mathsf{T}}RS^{*\mathsf{T}})^n(S \sqcap S^{*\mathsf{T}}RS^{*\mathsf{T}}) = (S \sqcap S^{*\mathsf{T}}RS^{*\mathsf{T}})^{n+1}.$$

The theorem now follows after summing over all $n \geqslant 0$. ■

Note that if R is reflexive, then the relation $R' = S^*RS^* \sqcap S^{*\mathsf{T}}RS^{*\mathsf{T}}$ also satisfies the stuttering property. This easily follows from Theorem 10.2.13 and the fact that $S^n \sqcap S^*RS^* = S^n$ for all $n \geqslant 0$. Note also that R' is symmetric if R is symmetric (contrary to $\mathrm{stt}_S(R)$ that might not be symmetric). This is important because we mostly work with symmetric relations.

We now show that branching bisimulation can also be defined with stuttering steps.

Theorem 10.2.14 If R is a reflexive branching bisimulation on the transition system $\langle \sigma, A, S, \rho \rangle \in \mathcal{TS}^n_{\mathsf{A}}$, then $R' = S^*RS^* \sqcap S^{*\mathsf{T}}RS^{*\mathsf{T}}$ is also branching bisimulation on $\langle \sigma, A, S, \rho \rangle$. □

Proof Note that R' is symmetric because R is.

We first prove that $R' \leqslant (S^* \sqcap R')R$. Using that $R\Pi \leqslant \Pi R$ and the Dedekind formula we have

$$R' = S^*RS^* \sqcap S^{*\mathsf{T}}RS^{*\mathsf{T}} \leqslant S^*R \sqcap RS^{*\mathsf{T}} \leqslant \\ \leqslant (S^* \sqcap RS^{*\mathsf{T}}R)(R \sqcap S^{*\mathsf{T}}RS^{*\mathsf{T}}) \leqslant (S^* \sqcap RS^{*\mathsf{T}})R \leqslant \\ \leqslant (S^* \sqcap S^{*\mathsf{T}}RS^{*\mathsf{T}})R = (S^* \sqcap S^*RS^* \sqcap S^{*\mathsf{T}}RS^{*\mathsf{T}})R \leqslant (S^* \sqcap R')R.$$

Note that since $S^* \sqcap R' = (S \sqcap R')^*$, we have $(S^* \sqcap R')^2 = S^* \sqcap R'$. Using this, we have

$$R'S \leqslant (S^* \sqcap R')RS \leqslant (S^* \sqcap R')(S^* \sqcap R)(I+S)R \leqslant \\ \leqslant (S^* \sqcap R')(S^* \sqcap R')(I+S)R' = (S \sqcap R')^*(I+S)R'$$

and

$$R'A \leqslant (S^* \sqcap R')RA \leqslant (S^* \sqcap R')(S^* \sqcap R)AR \leqslant \\ \leqslant (S^* \sqcap R')(S^* \sqcap R')AR' = (S \sqcap R')^*AR'.$$

Also, $R'\rho \leqslant (S^* \sqcap R')R\rho \leqslant (S^* \sqcap R')\rho$. ■

Conclusion to Part II

We have presented the theory of labeled transition systems (with successful termination) in terms of boolean matrices. We have covered the notion of forward and backward strong bisimulation, of bisimulation up-to a relation, and of weak and branching bisimulation. The powerful block structure matrix representation method has provided a nice mechanism for expressing the most common operations on transition systems as well. Matrix techniques have allowed us to (re)establish the standard results in a clearer, more concise, and uniform way.

By presenting results and notions from transition system theory in terms of matrices we are also able to establish some connections with Markov chain theory. First, the notion of minimization modulo strong bisimulation equivalence, i.e. of strong lumping from Definition 9.2.5, is shown to coincide with the notion of ordinary lumping [67] for Markov chains. As we will see in Part III, weak bisimulation also has an important interpretation in the Markov chain world (see Definition 13.2.1 of τ-lumping).

In the future, we plan to investigate how other equivalences from [52] can be represented in matrix terms. In particular, we want to see how they fit in the general aggregation scheme of [33]. We also want to investigate whether matrix theory can be used as a unifying framework for the reasoning about dynamic systems. We think that our result is a big step in that direction.

Part III

Aggregation of Markov Reward Chains with Fast and Silent Transitions

Chapter 11

Introduction

11.1 Motivation

Homogeneous continuous-time Markov chains (we will refer to them as *Markov chains* for short) have established themselves as very powerful, yet fairly simple models for performance evaluation. A Markov chain (see e.g. [42, 28, 66]) is a finite-state continuous-time stochastic process of which the (stochastic) behavior in every state is completely independent of the prior states visited (i.e. the process satisfies the Markov property) and of the time already spent in the state (i.e. the process is homogeneous in time). It is known that, if some continuity requirement is met, a Markov chain can be represented as a directed graph in which nodes represent states and labels on the outgoing arrows determine the stochastic behavior in the state. Some states are marked as starting and have initial probabilities associated with them. For example, the behavior of the Markov chain depicted in Figure 11.1a is as follows. The process starts from state 1 with probability π and from state 2 with probability $1-\pi$ (we do not depict the initial probability if it is zero). In state 1 it waits the amount of time determined by the minimum of two exponentially distributed delays, one parameterized with rate λ, the other with rate μ (note that this means that the process spends in state 1 exponentially distributed time with rate $\lambda+\mu$). After delaying the process jumps to state 2 or state 3 depending on which of the two delays was shorter. In these two states the process just stays forever, i.e. it is absorbed there.

To increase modeling capability and obtain some very useful performance measures, such as throughput and utilization of a system, Markov chains are often equipped with *rewards* [66]. There are many types of rewards but we

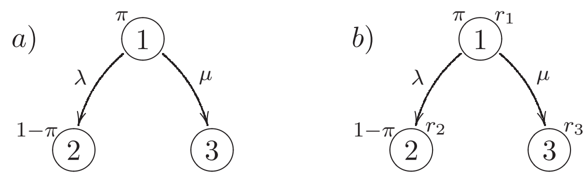


Figure 11.1: a) A simple Markov chain and b) a Markov reward chain

consider only those that are associated to states. A (state) reward represents the gain of a Markov chain while residing in some state. A Markov chain with rewards is called a Markov reward chain (see Figure 11.1b).

A vast mathematical theory has been developed to support Markov chains (as well as Markov reward chains). Efficient methods are available to deal with Markov chains with millions of states making them very applicable in practice. One of the main issues when using Markov chains is to find a Markov chain that correctly represents the system being analyzed.

Over the past few years several performance modeling techniques have been developed to enable high-level and compositional generation of Markov chains (and more recently also Markov reward chains), i.e. to provide ways of constructing big Markov chains from smaller components while staying on the designer level. Some of the best known techniques are stochastic process algebras [61, 63], (generalized) stochastic Petri nets [77, 76, 56], probabilistic I/O automata [101], stochastic automata networks [86], etc. Most of the formalisms first generate some intermediate models that are later used to derive pure Markov chains for performance measuring. Typically, these models are extensions of Markov chains with features to enable interaction between components. These features are special transitions that sometimes have undelayable behavior, i.e. they are instantaneous. In the literature instantaneous transitions are referred to as internal or silent steps (in process algebra) or as immediate transitions (in Petri nets). They are present in the intermediate model but are eliminated in a derivation of a Markov chain. We illustrate this approach in the fields of stochastic process algebra and Petri nets.

Stochastic process algebras are process algebras that include features for the modeling of exponentially distributed delays (e.g. [61, 63]). Stochastic information is generally introduced in one of two ways: by adding a delay parameter to actions, like e.g. in PEPA [63], or by adding delays as separate constructs, like e.g. in Interactive Markov Chains [61]. In the later case silent transitions play a prominent role. For Interactive Markov Chains the underlying Markov chain is obtained as follows. Under the assumption that system does not interact with the environment any longer, all action information can be discarded and the action labeled transitions are transformed into internal τ-transitions. These transitions are considered instantaneous and choices between them are made non-deterministically. To obtain a pure Markov chain τ-transitions are eliminated (if possible) by using a relation on transition systems called weak bisimulation, which is a combination of the standard weak bisimulation for transition systems [79] and of the aggregation method for Markov chains called ordinary lumping [67, 82, 23]. This weak bisimulation always gives priority to τ-transitions over exponential delays based on the intuitive fact that these transitions happen instantly. If there are closed loops of τ-transitions, then the model is considered ill-defined (here 'closed' means that there is no exit from the loop with a τ-transition). We give an example of a reduction modulo this weak bisimulation.

Example 11.1.1 Consider the Interactive Markov chain depicted in Figure 11.2a. If we assume that the system is closed, i.e. that it does not interact with the environment, then the actions a and b can be renamed into the instantaneous transition τ and an equivalent (with respect to performance) model is obtained. The intermediate model, consisting entirely of internal transitions and rates, is depicted in Figure 11.2b. Now, assume that the process in Figure 11.2b starts from state 1. There it exhibits a classical non-determinism, i.e. the probability of taking the τ-transitions is undetermined. However, if we observe the behaviors in states 2 and 3, we notice that they are the same. No matter which transition is taken from state 1, after performing a τ-transition and delaying exponentially with rate λ, the process enters state 4. As τ-transitions are timeless, the process in b) is equivalent to the Markov chain in c) according to weak bisimulation equivalence. □

Generalized stochastic Petri nets are introduced in [77] to enable performance modeling using Petri nets. A Petri net [29] is a bipartite graph with two sets of nodes: places and transitions. Input arcs connect places with transitions and output arcs connect transitions with places. Each place can contain several tokens. A so-called marking represents the configuration of the tokens in the places. A transition is enabled if there are tokens in all places that have an input arc to the transition. Each transition in a generalized stochastic Petri net has a so-called firing time, which can be zero (for immediate transitions) or exponentially distributed (for timed transitions). If a marking enables some immediate transition, then the marking is

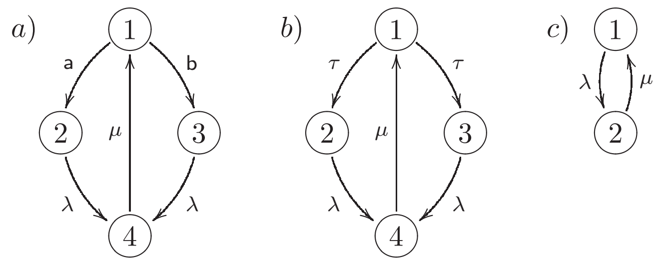


Figure 11.2: a) An Interactive Markov chain, b) the intermediate model with τ-transitions, and c) the induced Markov chain – Example 11.1.1.

called vanishing. The process described by a generalized stochastic Petri net is captured by a so-called extended reachability graph that represents the particular intermediate model and that can be further reduced to a Markov chain [29, 77, 76]. Of interest are the vanishing markings which exist in the extended reachability graph, but are eliminated to give the resulting Markov chain. It is common to assume that immediate transitions cannot form closed loops, i.e. these loops are considered illegal. Also, usually it is required to know the firing probabilities of multiple enabled immediate transitions [76]. A typical elimination of vanishing markings is given in Example 11.1.2.

Example 11.1.2 Figure 11.3 depicts a generalized stochastic Petri net with its corresponding reachability graph and the underlying Markov chain. The graph contains the markings of the only token placed initially in p_1. The vanishing place is p_2 (thus, the vanishing marking is 0100) because of the enabled immediate transitions t_2 and t_3 with probabilities p and $1 - p$. In the derived Markov chain the probabilities of the vanishing place split the normal rate λ into two rates $p\lambda$ and $(1 - p)\lambda$ that reach the final places p_3 and p_4, respectively. □

To prove that the original model and the underlying Markov chain have the same performance, the intermediate performance models from Figure 11.2b and Figure 11.3b must be defined as stochastic processes. The reduction technique of stochastic Petri nets has been (stochastically) formalized in [3] by treating the reachability graphs as discontinuous Markov chains [41] and eliminating the vanishing places by the aggregation approach of [39, 32]. However, this method is only possible when immediate transi-

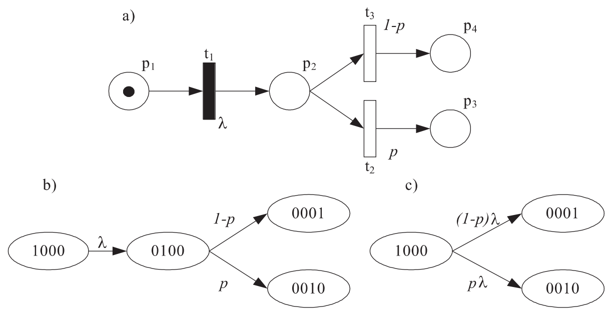


Figure 11.3: a) A generalized stochastic Petri net, b) the corresponding extended reachability graph, and c) the derived Markov chain – Example 11.1.2.

tions are probabilistic, and the same method cannot be directly applied in the case when they are non-deterministic (such as those in Figure 11.2b).

In this part we give a mathematical underpinning of the elimination of both, probabilistic and non-deterministic, types of instantaneous transitions in the above extensions to Markov (reward) chains. We define two methods of aggregation that abstract away from these transitions while preserving performance measures. The first method is based on lumping, i.e. joining states with equivalent behavior into classes. The second method is an extension of [39] (and therefore also of [3]). It is based on the elimination of stochastic discontinuity that arises from having instantaneous probabilistic transitions. The method is very common, often applied in perturbation theory, and this motivated us to extend it and adapt it to the setting with non-determinism. By discussing both methods in a common framework, we are able to compare them.

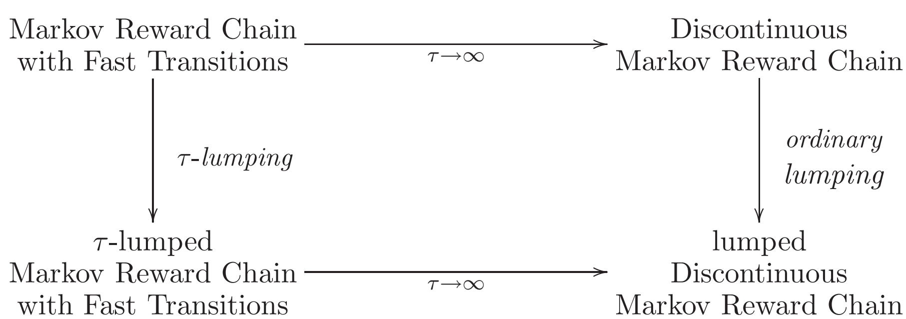
11.2 Our approach

We give an overview of the approach taken in this part.

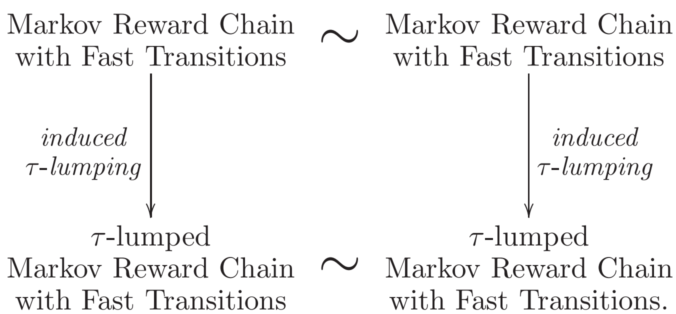
Extensions of the Markov reward chain model To stochastically formalize the phenomenon of instantaneous transitions we turn to a generalization of standard Markov chain model that can perform infinitely many transitions in a finite amount of time. This model is called a *discontinuous Markov reward chain* and it was initially studied (without rewards) in [41, 32]. It is often considered pathological in the literature as it exhibits stochastic discontinuity. However, as shown in [32, 3], it proves very useful for explanation of results. In order to model probabilistic instantaneous transitions we extend the standard Markov reward chain model with transitions that are linearly parameterized with a real variable τ. This extension is referred to as *Markov reward chains with fast transitions*. The intuition comes from the dynamics of Markov chains. If there are fast transitions $a\tau$ and $b\tau$ leading from a state, then the probability of taking $a\tau$ (resp. $b\tau$) is $\frac{a}{a+b}$ (resp. $\frac{b}{a+b}$). Therefore, the numbers a and b, called *speeds*, completely determine the probabilities of target states. We mathematically formalize the idea that fast transitions take zero time by considering the limit process as τ goes to infinity. Indeed, if there is a fast transition $a\tau$ leads from a state, then the sojourn time in this state is of the form $\frac{1}{a\tau+\ldots}$ and it goes to 0 when τ goes to infinity. The limit process is a discontinuous Markov reward chain. Subsequently, we introduce *Markov reward chains with silent transitions* as classes of Markov reward chains with fast transitions that have the same structure, but different speeds assigned to the fast transitions. Thus, a silent transition is a fast transition with unspecified speed, i.e., with unspecified probability of choosing it. This is our way of modeling non-determinism.

For each extension, we introduce two aggregation methods.

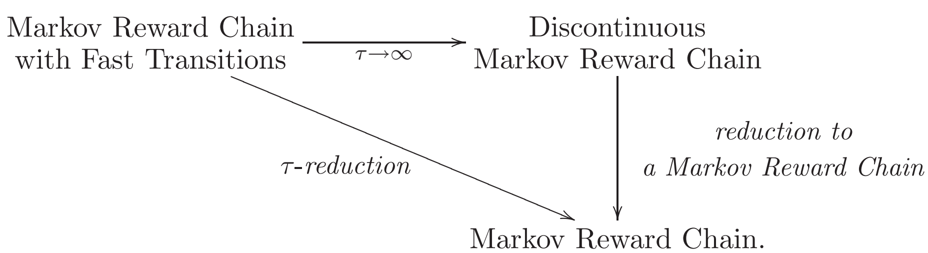
Aggregation by Lumping The first aggregation method is based on lumping, i.e. on joining all states that exhibit the same behavior into classes. We decided to consider the lumping method not only because it is the most common method of aggregation for standard Markov chains, but also because it allows us to formalize the intuitive ideas behind weak bisimulation for Interactive Markov chains. Extending the notion of ordinary lumping for Markov reward chains, we first define a notion of lumping for discontinuous Markov reward chains. Based on that, we define a notion of lumping for Markov reward chains with fast transitions, called τ-lumping. We justify the latter notion by showing that the following diagram commutes:



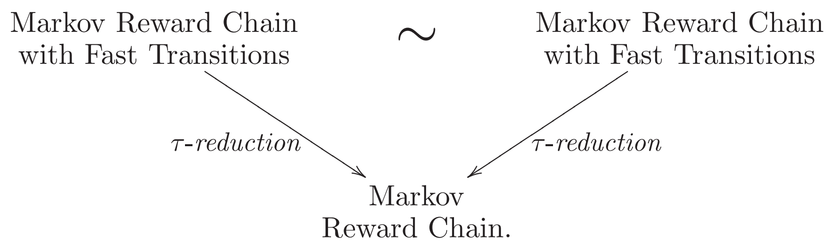
Next, we define a notion of lumping, called $\tau_{\sim}$-lumping, for Markov reward chains with silent transitions, and show that it is a proper lifting of τ-lumping to equivalence classes of Markov reward chains with fast transitions. In other words, we show that $\tau_{\sim}$-lumping induces a τ-lumping for each element of the class, and moreover, that the induced τ-lumped process does not depend on the representative from the class. That is, we show that the following diagram commutes:



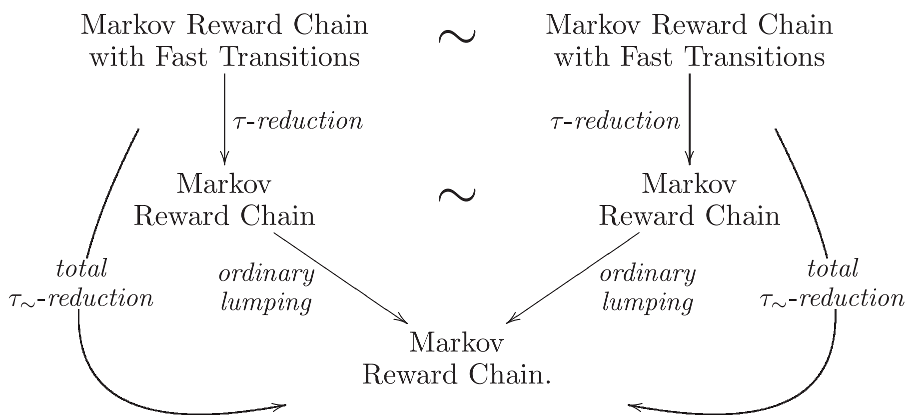
Aggregation by Reduction It is straightforward to obtain (e.g. by comparison of the matrix techniques used) that the methods for elimination of vanishing markings in generalized stochastic Petri nets given in [76, 56, 29, 77, 27] are equivalent to the reduction method in perturbation theory (cf. [32, 38]). We recall the results from this setting that allow us to reduce a discontinuous Markov chain to a Markov chain. Then we extend this technique to discontinuous Markov reward chains and Markov reward chains with fast transitions. The corresponding method for Markov reward chains with fast transitions is referred to as τ-reduction and the following diagram shows its structure:



Subsequently, we extend the notion of τ-reduction to Markov reward chains with silent transitions by lifting it to equivalence classes of Markov reward chains with fast transitions. The obtained aggregation method is called $\tau_{\sim}$-reduction. The main requirement for a class to be $\tau_{\sim}$-reducible is that its every representative Markov reward chain with fast transitions τ-reduces to a speed independent Markov chain. This is illustrated by the following diagram:



Motivated by the fact that $\tau_\sim$-reduction in general does not aggregate much, we introduce a new concept, called *total* $\tau_\sim$*-reduction*, that is a combination of τ-reduction and standard ordinary lumping on the τ-reduced representative Markov reward chain with fast transitions. The idea is to eliminate the effect of the speeds of fast transitions by lumping, and thus to aggregate more. The following diagram clarifies the structure of the method:



Comparison of the methods Each of the reduction methods is compared with its corresponding lumping method. We show that the reduction and the lumping methods for discontinuous Markov chains and Markov reward chains with fast transitions are incomparable but that the reduction method is superior, i.e. it aggregates more, if combined with standard lumping. We also show that, in case there are no silent transitions in the lumped process, $\tau_\sim$-reduction is a special case of $\tau_\sim$-lumping, and that $\tau_\sim$-lumping coincides with total $\tau_\sim$-reduction. Finally, we point out the differences between $\tau_\sim$-lumping and the weak bisimulation for Interactive Markov chains.

11.3 Outline

The mentioned extensions to Markov chains, i.e. Markov reward chains, discontinuous Markov reward chains, Markov reward chains with fast transitions and Markov reward chains with silent transitions, are introduced in Chapter 12, and necessary theorems are provided to establish the connections between them. In Chapter 13 we define the ordinary lumping for discontinuous Markov chains, and the notions of τ- and $\tau_{\sim}$-lumping. In Chapter 14 we recall the reduction method for discontinuous Markov chains, extend it to discontinuous Markov reward chains, and define τ-, $\tau_{\sim}$- and total $\tau_{\sim}$-reductions. The lumping and the reduction method are compared in Chapter 15.

Chapter 12

Markov Reward Chains with Discontinuities, and with Fast and Silent Transitions

This chapter introduces several extensions of standard Markov chains. We first recall the definition of a discontinuous Markov chain from [41, 32], i.e. of a Markov chain that can also exhibit non-continuous behavior, and extend it with rewards. Next, the standard Markov reward chain model is extended with special transitions called fast transitions. As explained in the introduction, this is to model probabilistic transitions. We show that Markov reward chains with fast transitions are asymptotically equivalent to discontinuous Markov reward chains. Finally, to model non-determinism we introduce Markov reward chains with silent transitions as Markov reward chains with fast transitions in which the speeds of the fast transitions are unknown.

12.1 Discontinuous Markov reward chains

The standard theory of Markov chains [42, 28, 66] always assumes continuity, i.e. that, when $t \to 0$, the probability of the process occupying at time t the same state as at time 0 is 1. However, as pointed out in [32], when working with instantaneous transitions we need to drop this requirement and work in the more general theory of discontinuous Markov chains introduced in [41]. In this section we give a definition of the notion of discontinuous Markov chain. We follow the approach of [32] but add initial probabilities and rewards.

The exposition is in terms of matrices and we give some preliminaries first. All vectors are column vectors if not indicated otherwise. $\mathbf{1}^n$ denotes the vector of n 1's. $\mathbf{0}^{n\times m}$ denotes the $n \times m$ zero matrix. I^n denotes the $n \times n$ identity matrix. We omit the n and m when they are clear from the context. We write $A > 0$ (resp. $A \geqslant 0$) when all elements of a matrix A are greater than (resp. greater than or equal to) zero. A matrix $A \in \mathbb{R}^{n\times m}$ is called *stochastic* if $A \geqslant 0$ and $A \cdot \mathbf{1} = \mathbf{1}$. By $\mathrm{diag}\,(A_1, \ldots, A_n)$ we denote the block matrix with blocks $A_1, \ldots, A_n$ on the diagonal and $\mathbf{0}$'s elsewhere.

A discontinuous Markov chain is a time-homogeneous finite-state stochastic process that satisfies the Markov property. The exact nature of a state is not important and we can always assume that the state space of a discontinuous Markov chain is the (linearly ordered) set $\mathcal{S} = \{1, \ldots, n\}$. It is known (see [41, 28, 32]) that a discontinuous Markov chain is then completely determined by a *transition matrix function* and a stochastic *row* vector that gives the starting probabilities of the process for each state (called the *initial probability vector*).

Definition 12.1.1 (Transition matrix function) A function $P : \mathbb{R}_{>0} \mapsto \mathbb{R}^{n\times n}$ is called a *transition matrix function* iff, for all $t > 0$,

1. $P(t) \geqslant 0$,
2. $P(t) \cdot \mathbf{1} = \mathbf{1}$ and
3. $P(t + s) = P(t) \cdot P(s)$ for all $s > 0$.

If $\lim_{t\to 0} P(t)$ is equal to the identity matrix, then P is called *continuous*, otherwise it is *discontinuous* (it is shown in [42] that this limit always exists). For any $t > 0$, we call the image $P(t)$ a *transition matrix*. As is standard practice, whenever we say transition matrix $P(t) = \ldots$ we actually mean transition matrix function P defined by $P(t) = \ldots$. □

The following theorem of [32, 62] gives a convenient characterization (independent on t) of the notion of transition matrix.

Theorem 12.1.2 Let $(\Pi, Q) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times n}$ be such that:

1. $\Pi \geqslant 0,\ \ \Pi \cdot \mathbf{1} = \mathbf{1},\ \ \Pi^2 = \Pi$,
2. $\Pi Q = Q\Pi = Q$,
3. $Q \cdot \mathbf{1} = \mathbf{0}$ and

4. $Q + c\Pi \geqslant 0$ for some $c \geqslant 0$.

Then $P(t) = \Pi e^{Qt} = \Pi \sum_{n=0}^{\infty} \frac{Q^n t^n}{n!}$ is a transition matrix. Moreover, the converse also holds: For any transition matrix $P(t)$ there exists a unique pair (Π, Q) that satisfies Conditions 1–4 and such that $P(t) = \Pi e^{Qt}$. □

Note that, since $\lim_{t\to 0} P(t) = \lim_{t\to 0} \Pi e^{Qt} = \Pi$, it follows that $P(t)$ is continuous iff $\Pi = I$. In this case Q is a *generator* matrix, i.e. a square matrix of which the non-diagonal elements are non-negative and each diagonal element is the additive inverse of the sum of the non-diagonal elements of the same row.

The discontinuous Markov chain determined by a transition matrix $P(t) = \Pi e^{Qt} \in \mathbb{R}^{n\times n}$ and an initial probability vector $\sigma \in \mathbb{R}^{1\times n}$ is denoted by (σ, Π, Q). Strictly speaking, different orderings on the set $\mathcal{S}$ give rise to different discontinuous Markov chains, but we will not make a distinction between them. This is because there is no real difference, the representing matrices are permutation equivalent. All our results can be easily shown to be insensitive to permutation. This allows us to always work with the numbering that is most convenient at the moment.

In the case when $\Pi = I$, the discontinuous Markov chain (σ, Π, Q) has no stochastic discontinuity and is a *standard* Markov chain. Since Q is then a generator matrix, the process has the standard visual representation (like in Figure 11.1a). We give an example.

Example 12.1.3 a. The matrix

$$P(t) = \begin{pmatrix} e^{-(\lambda+\mu)t} & \frac{\lambda}{\lambda+\mu}(1-e^{-(\lambda+\mu)t}) & \frac{\mu}{\lambda+\mu}(1-e^{-(\lambda+\mu)t}) \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

with $\lambda, \mu \geqslant 0$ and $\lambda + \mu \neq 0$, is a transition matrix. It is continuous because $\lim_{t\to 0} P(t) = I$. We obtain

$$\Pi = I \quad \text{and} \quad Q = \begin{pmatrix} -(\lambda+\mu) & \lambda & \mu \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

As expected, in this case Q is a generator matrix. For $\sigma = \begin{pmatrix} \pi & 1-\pi & 0 \end{pmatrix}$, the (standard) Markov chain (σ, I, Q) is visualized in Figure 11.1a from the introduction.

b. Let $0 < p < 1$ and $\lambda \geqslant 0$. Then

$$P(t) = \begin{pmatrix} (1-p) \cdot e^{-p\lambda t} & p \cdot e^{-p\lambda t} & 1-e^{-p\lambda t} \\ (1-p) \cdot e^{-p\lambda t} & p \cdot e^{-p\lambda t} & 1-e^{-p\lambda t} \\ 0 & 0 & 1 \end{pmatrix}$$

is a transition matrix. It is discontinuous because

$$\Pi = \lim_{t \to 0} P(t) = \begin{pmatrix} 1-p & p & 0 \\ 1-p & p & 0 \\ 0 & 0 & 1 \end{pmatrix} \neq I.$$

We also obtain

$$Q = \begin{pmatrix} -p(1-p)\lambda & -p^2\lambda & p\lambda \\ -p(1-p)\lambda & -p^2\lambda & p\lambda \\ 0 & 0 & 0 \end{pmatrix}.$$

Note that Π deviates from the identity matrix only in the first two rows. This is exactly where Q deviates from the form of a generator matrix. □

It is a known result (see e.g. [32]) that there is a numbering of $\mathcal{S}$ in which in a discontinuous Markov chain (σ, Π, Q), the matrix Π takes the following form:

$$\Pi = \begin{pmatrix} \Pi_1 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Pi_2 & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \Pi_M & \mathbf{0} \\ \overline{\Pi}_1 & \overline{\Pi}_2 & \dots & \overline{\Pi}_M & \mathbf{0} \end{pmatrix},$$

where for all $1 \leq K \leq M$, $\Pi_K = \mathbf{1} \cdot \mu_K$, and $\overline{\Pi}_K = \delta_K \cdot \mu_K$ for a *row* vector $\mu_K > 0$ such that $\mu_K \cdot \mathbf{1} = 1$ and a vector $\delta_K \geq 0$ such that $\sum_{i=1}^{M} \delta_K = \mathbf{1}$.

We now show that the form of Π divides the states into groups. First, we formalize the notion of partitioning.

Definition 12.1.4 (Partitioning) Let $\mathcal{S}$ be a set. A set $\mathcal{P} = \{S_1, \dots, S_N\}$ of subsets of $\mathcal{S}$ is called a *partitioning* of $\mathcal{S}$ if $\mathcal{S} = S_1 \cup \ldots \cup S_N$, $S_i \neq \emptyset$ and $S_i \cap S_j = \emptyset$ for all i, j, with $i \neq j$. The partitionings $\mathcal{P} = \big\{\mathcal{S}\big\}$ and $\mathcal{P} = \big\{\{i\} \mid i \in \mathcal{S}\big\}$ are called trivial. □

Given a set $\mathcal{S}$ and its partitioning $\mathcal{P} = \{S_1, \dots, S_N\}$, it is sometimes convenient to number the elements of $\mathcal{S}$ so that the elements from the same partitioning class are grouped together. Formally, we require that, for all

$i \in \mathcal{S}$, if $i \in S_K$ for some $1 \leqslant K \leqslant N$, then either $i+1 \in S_K$ or $j \notin S_K$ for all $j > i$. Any such numbering of $\mathcal{S}$ is called the numbering that makes the partitioning $\mathcal{P}$ explicit.

The form of Π induces a partitioning $\mathcal{E} = \{E_1, \ldots, E_M, T\}$ of $\mathcal{S} = \{1, \ldots, n\}$ into *ergodic* classes, $E_1, \ldots, E_M$, determined by $\Pi_1, \ldots, \Pi_M$, and into a class of *transient* states, T, determined by $\overline{\Pi}_1, \ldots, \overline{\Pi}_M$. The partitioning $\mathcal{E}$ is called the *ergodic partitioning* and the used numbering is making it explicit (not the additional requirement here; ergodic states must precede transient states). For every ergodic class E_K, the vector μ_K is the vector of *ergodic probabilities*. If an ergodic class E_K contains exactly one state, then $\mu_K = \big(1\big)$, and the state is called *regular*. The vector δ_K holds the *trapping probabilities* from transient states to the ergodic class E_K. Note that, although μ_K and δ_K are not indexed by $\{1, \ldots, n\}$, without introducing confusion, we will always use the implicit indexing. In other words, for any $i \in E_K$, we will write $\mu_K[i]$ to refer to the element of μ_K that corresponds to state i. Similarly, we write $\delta_K[i]$ for any $i \in T$.

Let us now explain the behavior of a discontinuous Markov chain as given in [41, 32]. The discontinuous Markov chain (σ, Π, Q) starts in some state with a probability that is determined by the initial probability vector σ. In an ergodic class with multiple states the process spends a non-zero amount of time switching rapidly (infinitely many times) among its elements. The probability that it is found in some state of this class is determined by the vector of ergodic probabilities of this class. The time the process spends in the class is exponentially distributed and determined by the matrix Q. If the ergodic class contains only one state i, i.e. if the process is in a regular state, then the row of Q corresponding to i has the form of a row in a generator matrix, and $Q[i,j]$ for $i \neq j$ is interpreted as the rate from i to j. In a transient state the process spends no time (with probability one) and goes immediately to some ergodic class (and stays trapped there for some amount of time). Note that $\delta_K[i] > 0$ iff $i \in T$ can be trapped in the ergodic class E_K. A standard Markov chain is a discontinuous Markov chain that has no transient states and only has regular (ergodic) states.

Sometimes we will also work with the matrix Π that is not in the above form, i.e. we will work in a numbering that does not make the ergodic partitioning explicit. Let us so explain the form of Π on the level of single elements. Note first that $\Pi[i,j] = 0$ for all $i \in \mathcal{S}$ and all $j \in T$. Next, note that if $i \in E_K$, $j \in E_L$ and $K \neq L$, then $\Pi[i,j] = 0$. If $i, j \in E_K$, then we have $\Pi[i,j] > 0$ and $\Pi[i,j] = \Pi[k,j]$ for all $k \in E_K$. In this case we also have that $\Pi[i,j] = \mu_K[j]$. For transient states we have that if $i \in T$ and $j \in E_K$, then $\Pi[i,j] = \delta_K[i] \cdot \Pi[k,j]$ for any $k \in E_K$.

We give examples of some discontinuous Markov chains and their ergodic partitionings.

Example 12.1.5 a. Let (σ, I, Q) be the standard Markov chain from Example 12.1.3a. Its ergodic partitioning is $\mathcal{E} = \{E_1, E_2, E_3\}$ where $E_1 = \{1\}$, $E_2 = \{2\}$ and $E_3 = \{3\}$. As expected, there are no transient states and all ergodic classes are singletons.

b. Let (σ, Π, Q) be the discontinuous Markov chain from Example 12.1.3b. This discontinuous Markov chain has two ergodic classes $E_1 = \{1, 2\}$ and $E_2 = \{3\}$ and no transient states. The corresponding ergodic probability vectors are $\mu_1 = \begin{pmatrix} 1-p & p \end{pmatrix}$ and $\mu_2 = \begin{pmatrix} 1 \end{pmatrix}$. In the first two states the process exhibits discontinuous behavior. It constantly switches among those states and it is found in the first one with probability $1-p$ and in the second one with probability p. The amount of time the process spends switching is exponentially distributed with rate $p\lambda$ (we will see later how this follows from Q). Note that also the rows of Q that correspond to states belonging to the same ergodic classes are equal. This indicates that those states all have the same behavior.

c. Let, for $0 < p < 1$ and $\lambda, \mu, \nu > 0$, Π and Q be defined as:

$$\Pi = \begin{pmatrix} 0 & p & 1-p & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad Q = \begin{pmatrix} 0 & -p\lambda & -(1-p)\mu & p\lambda + (1-p)\mu \\ 0 & -\lambda & 0 & \lambda \\ 0 & 0 & -\mu & \mu \\ \nu & 0 & 0 & -\nu \end{pmatrix}.$$

Let σ be an arbitrary stochastic row vector. The ergodic partitioning of the discontinuous Markov chain (σ, Π, Q) is $\mathcal{E} = \{E_1, E_2, E_3, T\}$ where $E_1 = \{2\}$, $E_2 = \{3\}$, $E_3 = \{4\}$ and $T = \{1\}$ (note that the numbering does not make the ergodic partitioning explicit since the transient state precedes the ergodic states). We have $\mu_i = \begin{pmatrix} 1 \end{pmatrix}$ for all $i = 1, 2, 3$, and $\delta_1 = \begin{pmatrix} p \end{pmatrix}$, $\delta_2 = \begin{pmatrix} 1-p \end{pmatrix}$ and $\delta_3 = \begin{pmatrix} 0 \end{pmatrix}$. If the process is in state 1, then with probability p it is trapped in state 2, the only state in the ergodic class E_1, and with probability $1-p$ it is trapped in state 3, the only state in the ergodic class E_2. $\square$

12.1.1 Adding rewards

We now add (state) rewards to our model. As we said in the introduction, this addition is of great practical importance. A reward is a number associated to a state that represents the rate at which gain is received while the process is in that state. We define a discontinuous Markov reward chain as a discontinuous Markov chain with an additional vector that holds a reward for each state.

Definition 12.1.6 (Discontinuous Markov Reward Chain) A *discontinuous Markov reward chain* is a quadruple (σ, Π, Q, ρ) where (σ, Π, Q) is a discontinuous Markov chain and $\rho \in \mathbb{R}^{n\times 1}$ is the *reward vector*. □

The *total reward (rate)* of the process up to time $t > 0$, denoted $R(t)$, is calculated as $R(t) = \sigma P(t)\rho$. It represents the core of the most important performance measure, i.e. of the *expected accumulated reward*, which is calculated by $\int_0^t R(s)ds$.

Note that the total reward remains unchanged if the reward vector ρ is replaced by $\Pi\rho$. To show this, note that $P(t) = P(t)\Pi$ (cf. [32]), so $\sigma P(t)\Pi\rho = \sigma P(t)\rho = R(t)$. Intuitively, the reward of an ergodic state can be replaced by the sum of the rewards of all states inside its ergodic class weighted according to their ergodic probabilities, and the reward in a transient state can be replaced by the sum of the rewards of the ergodic states that it can be trapped in weighted by the trapping probabilities. Note that this means that the reward in a transient states is actually irrelevant. This is expected since in a transient state the process spends no time nor does it ever come back to it. The technique of replacing the reward vector simplifies the reward structure which becomes important for the aggregation methods in the latter chapters. We give an illustration in the following example.

Example 12.1.7 a. Let (σ, I, Q, ρ) be the standard discontinuous Markov reward chain where (σ, I, Q) is as in Example 12.1.3a and $\rho = \begin{pmatrix} r_1 & r_2 & r_3 \end{pmatrix}$. Recall that the transition matrix is

$$P(t) = \begin{pmatrix} e^{-(\lambda+\mu)t} & \frac{\lambda}{\lambda+\mu}(1-e^{-(\lambda+\mu)t}) & \frac{\mu}{\lambda+\mu}(1-e^{-(\lambda+\mu)t}) \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Then, we calculate the total reward:

$$R(t) = \sigma P(t)\rho = \begin{pmatrix} \pi & 1-\pi & 0 \end{pmatrix} P(t) \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} =$$

$$= \pi r_1 e^{-(\lambda+\mu)t} + \pi \frac{\lambda r_2 + \mu r_3}{\lambda+\mu}(1-e^{-(\lambda+\mu)t}).$$

Note that all the rewards contribute to the total reward. This is because the process does not have transient states (cf. Example 12.1.5a).

b. Let (σ, Π, Q) be the discontinuous Markov chain from Example 12.1.3b (with $\sigma = \begin{pmatrix}\pi & 1{-}\pi & 0\end{pmatrix}$) and let $\rho = \begin{pmatrix}r_1 & r_2 & r_3\end{pmatrix}$. The total reward of the discontinuous Markov reward chain (σ, Π, Q, ρ) is:

$$R(t) = \sigma P(t)\rho = ((1{-}p)r_1 + pr_2 - r_3)\, e^{-p\lambda t} + r_3.$$

The same total reward is obtained when ρ is replaced by the reward vector $\rho' = \Pi\rho = \begin{pmatrix} (1{-}p)r_1{+}pr_2 \\ (1{-}p)r_1{+}pr_2 \\ r_3 \end{pmatrix}$. Note that the first two elements of ρ' are equal. This is because these two states belong to the same ergodic class (cf. Example 12.1.5b). As in the previous example, there are no transient states and hence all the rewards are important.

c. Let (σ, Π, Q) be the discontinuous Markov chain from Example 12.1.5c (with $\sigma = \begin{pmatrix}1 & 0 & 0\end{pmatrix}$) and let $\rho = \begin{pmatrix}r_1 & r_2 & r_3 & r_4\end{pmatrix}^{\mathsf{T}}$. The total reward of the discontinuous Markov reward chain (σ, Π, Q, ρ) does not depend on r_1 because state 1 is transient (cf. Example 12.1.5c). This is confirmed when ρ is replaced by $\rho' = \Pi\rho = \begin{pmatrix} pr_2{+}(1{-}p)r_3 \\ r_2 \\ r_3 \\ r_4 \end{pmatrix}$. □

12.2 Markov reward chain with fast transitions

We extend the standard Markov chain model by letting Markov chains contain two types of transitions, slow and fast. The behavior of a Markov reward chain with fast transitions is determined by a pair of generator matrices: the first matrix represents the normal (slow) transitions, whereas the second represents the (*speeds* of the) fast transitions. As we explained in the introduction, the role of speeds is to determine the probabilistic behavior in a state.

Definition 12.2.1 (Markov reward chain with fast transitions) The *Markov reward chain with fast transitions* (σ, Q_s, Q_f, ρ), determined by a stochastic row vector $\sigma \in \mathbb{R}^{1\times n}$, generator matrices $Q_s, Q_f \in \mathbb{R}^{n\times n}$ and a vector $\rho \in \mathbb{R}^{n\times 1}$, is the function that assigns to each $\tau > 0$ the Markov reward chain $(\sigma, I, Q_s + \tau Q_f, \rho)$. □

We depict a Markov reward chain with fast transitions (σ, Q_s, Q_f, ρ) as the corresponding Markov reward chain $(\sigma, I, Q_s + \tau Q_f, \rho)$ (see Figure 12.1).

The following theorem shows that when $\tau \to \infty$, i.e. when fast transitions become instantaneous, a Markov reward chain with fast transitions behaves as a discontinuous Markov reward chain.

Theorem 12.2.2 (Limit process) Let $P_\tau(t) = e^{(Q_s+\tau Q_f)t}$. Then, for all $t > 0$,

$$\lim_{\tau\to\infty} P_\tau(t) = \Pi e^{Qt}$$

where $\Pi = \lim_{t\to\infty} e^{Q_f t}$ and $Q = \Pi Q_s \Pi$. In addition, Π and Q satisfy Conditions 1–4 of Theorem 12.1.2. □

See [25] for the first proof of Theorem 12.2.2, or [69] for a proof written in more modern terms. See [32] for the proof that convergence is also uniform.

If Q is a generator matrix, then $\Pi = \lim_{t\to\infty} e^{Qt}$ is called the *ergodic projection* of Q. It is proven in [42] that the limit always exists; moreover, see e.g. [1] for the following result:

Theorem 12.2.3 The matrix $\Pi \in \mathbb{R}^{n\times n}$ is the ergodic projection of a generator matrix $Q \in \mathbb{R}^{n\times n}$, iff

$$\Pi \geqslant 0,\ \Pi \cdot \mathbf{1} = \mathbf{1},\ \Pi^2 = \Pi,\ \Pi Q = Q\Pi = \mathbf{0},$$

and $\mathrm{rank}(\Pi) + \mathrm{rank}(Q) = n$. □

Theorem 12.2.2 shows that the limit behavior of a Markov reward chain with fast transitions does not directly depend on the matrix that models fast transitions but only on its ergodic projection. In general, there are many generator matrices that have the same ergodic projection.

We say that the discontinuous Markov chain $(\sigma, \Pi, Q, \Pi\rho)$ is *the limit* of (σ, Q_s, Q_f, ρ) as $\tau \to \infty$, and indicate that by writing $(\sigma, Q_s, Q_f, \rho) \to_\infty (\sigma, \Pi, Q, \Pi\rho)$. The initial probability vector and the reward vector are not affected when $\tau \to \infty$ but it is convenient to replace the reward vector ρ by $\Pi\rho$ because of the facilitated representation of the lumping conditions in the following sections.

The ergodic partitioning of $(\sigma, \Pi, Q, \Pi\rho)$ is also said to be the ergodic partitioning of (σ, Q_s, Q_f, ρ). It is known that this corresponds with the partitioning induced by closed communicating classes of fast transitions. We write $i \to j$ if $Q_f[i,j] > 0$, i.e. if there is a direct fast transition from i to j. Let $\twoheadrightarrow$ denote the reflexive-transitive closure of $\to$. If $i \twoheadrightarrow j$ we say that j is *τ-reachable* from i. If $i \twoheadrightarrow j$ and $j \twoheadrightarrow i$ we say that i and j *τ-communicate* and write $i \leftrightarrows j$. In a slightly different context, it has been shown (see e.g. [42]) that every ergodic class is actually a closed class of

τ-communicating states, closed meaning that for all i inside the class there does not exist j outside the class such that $i \to j$. Moreover, for all states i and all ergodic states j, $i \twoheadrightarrow j$ iff $\Pi[i,j] > 0$.

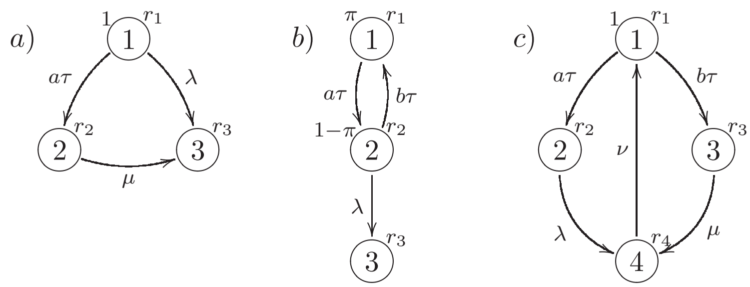


Figure 12.1: Markov reward chains with fast transitions – Example 12.2.4

Example 12.2.4 a. Consider the Markov reward chain with fast transitions (σ, Q_s, Q_f, ρ) depicted in Figure 12.1a. It is defined with

$$\sigma = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix},\ Q_s = \begin{pmatrix} -\lambda & 0 & \lambda \\ 0 & -\mu & \mu \\ 0 & 0 & 0 \end{pmatrix},\ Q_f = \begin{pmatrix} -a & a & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \text{ and } \rho = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix}.$$

The transition from state 1 to state 2 is fast and has speed a. The other two transitions are normal (slow).

The limit of (σ, Q_s, Q_f, ρ) is obtained as follows:

$$\Pi = \lim_{t\to\infty} e^{Q_f t} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$Q = \Pi Q_s \Pi = \begin{pmatrix} 0 & -\mu & \mu \\ 0 & -\mu & \mu \\ 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \Pi\rho = \begin{pmatrix} r_2 \\ r_2 \\ r_3 \end{pmatrix}.$$

The ergodic partitioning is $E_1 = \{2\}$, $E_2 = \{3\}$ and $T = \{1\}$. This is because, as we see it in Figure 12.1a, state 2 and state 3 each form a trivial τ-communicating class. The same can be obtained by observing the form of Π.

b. Consider the Markov reward chain with fast transitions depicted in Figure 12.1b. The limit of this Markov reward chain with fast transitions is the discontinuous Markov reward chain (σ, Π, Q, ρ') defined in Examples 12.1.3b and 12.1.7b (with $p = \frac{a}{a+b}$). From Figure 12.1b we can easily see that the process has two closed τ-communicating classes, i.e. two ergodic classes $E_1 = \{1, 2\}$ and $E_2 = \{3\}$, and no transient states. The same was established in Example 12.1.5b.

c. The limit of the Markov reward chain with fast transitions in Figure 12.1c is the discontinuous Markov reward chain (σ, Π, Q, ρ') defined in Examples 12.1.5c and 12.1.7c (with $p = \frac{a}{a+b}$). From Figure 12.1c we obtain that the ergodic partitioning is determined by $E_1 = \{2\}$, $E_2 = \{3\}$, $E_3 = \{4\}$ and $T = \{1\}$. This is confirmed by Example 12.1.5c. □

12.3 Markov reward chains with silent transitions

In this section we define Markov reward chains that can exhibit nondeterministic behavior and call them Markov reward chains with silent transitions. A Markov reward chain with silent transitions is a Markov reward chain with fast transitions in which the speeds of the fast transitions are considered unspecified. In other words, we define a Markov reward chain with silent transitions by abstracting from the speeds in a Markov reward chain with fast transitions. For this, we need to introduce a special equivalence relation on matrices.

Definition 12.3.1 (Matrix grammar) Two matrices $A, B \in \mathbb{R}^{n \times n}$ are said to have the *same grammar*, denoted by $A \sim B$, if for all $1 \leqslant i, j \leqslant n$, $A[i, j] = 0$ iff $B[i, j] = 0$. □

Example 12.3.2 The matrices $\begin{pmatrix} 2 & -3 \\ -5 & 0 \end{pmatrix}$ and $\begin{pmatrix} -1 & 2 \\ -4 & 0 \end{pmatrix}$ have the same grammar while the matrices $\begin{pmatrix} 2 & -3 \\ -5 & 0 \end{pmatrix}$ and $\begin{pmatrix} -1 & 0 \\ -4 & 0 \end{pmatrix}$ do not. □

The abstraction from speeds is achieved by identifying generator matrices with the same grammar. A Markov reward chain with silent transitions is defined as a Markov reward chain with fast transitions but instead of one matrix that models fast transitions we take the whole equivalence class induced by $\sim$. Note that we do not take elements of the matrix to be sets, but rather take the set of matrices instead. The consequence is that a Markov reward chain with silent transitions is not allowed to choose different speeds each time it enters some state. Our approach to resolving non-determinism therefore corresponds to the one of probabilistic, history independent, schedulers [92]. Having the quantification inside a matrix would lead to a much more complicated theory because it would force us to move from Markov chains to a model similar to *Markov set chains* [58].

Definition 12.3.3 (Markov reward chain with silent transitions) A *Markov reward chain with silent transitions* is a quadruple $(\sigma, Q_s, \mathcal{Q}_f, \rho)$ where $\mathcal{Q}_f$ is an equivalence class of $\sim$ and, for every $Q_f \in \mathcal{Q}_f$, (σ, Q_s, Q_f, ρ) is a Markov reward chain with fast transitions. □

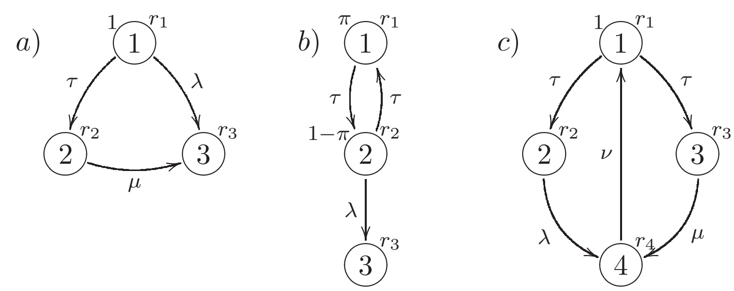


Figure 12.2: Markov reward chains with silent transitions corresponding to the Markov reward chains with fast transitions from Figure 12.1

A Markov reward chain with silent transitions $(\sigma, Q_s, \mathcal{Q}_f, \rho)$ is visualized as the Markov reward chain with fast transitions (σ, Q_s, Q_f, ρ), where $Q_f \in \mathcal{Q}_f$, but omitting the speeds of fast transitions. Figure 12.2 shows the Markov reward chains with silent transitions that correspond to the Markov reward chains with fast transitions from Figure 12.1.

Note that the notions of τ-reachability, τ-communication, and ergodic partitioning are speed independent, so they naturally carry over to Markov reward chains with silent transitions.

Chapter 13

Aggregation by Lumping

Lumping [67, 23, 82] is an aggregation method based on joining together states that exhibit equivalent behavior. In this chapter we introduce a notion of lumping for each of the Markovian models from Chapter 12. We first generalize the *ordinary lumping* method from standard Markov chains to discontinuous Markov reward chains. Then we introduce a lumping method for Markov reward chains with fast transitions, called τ*-lumping*, that assures that the limit process of the lumped Markov reward chain with fast transitions is the lumped version of the limit process of the original Markov reward chain with fast transitions. Finally, we lift τ-lumping to Markov reward chains with silent transitions and call it $\tau_\sim$*-lumping*. We show that $\tau_\sim$-lumping induces a τ-lumping for all possible speeds of fast transitions and, moreover, that the slow transitions in the induced τ-lumped process do not depend on those speeds.

13.1 Ordinary lumping

Partitioning is a central notion in the definition of lumping, so recall Definition 12.1.4. To define lumping in matrix terms it is standard to associate, with every partitioning $\mathcal{P} = \{C_1, \ldots, C_N\}$ of $\mathcal{S} = \{1, \ldots, n\}$, the following two matrices. A matrix $V \in \mathbb{R}^{n \times N}$ defined as

$$V[i,j] = \begin{cases} 0, & i \notin C_j \\ 1, & i \in C_j \end{cases}$$

is called the *collector* matrix for $\mathcal{P}$. Its j-th column has 1's for elements corresponding to states in C_j and has zeroes otherwise. Note that $V \cdot \mathbf{1} = \mathbf{1}$. For the trivial partitionings $\mathcal{P} = \{\mathcal{S}\}$ and $\mathcal{P} = \{\{i\} \mid i \in \mathcal{S}\}$, we have $V = \mathbf{1}$ and $V = I$ respectively.

A matrix $U \in \mathbb{R}^{N\times n}$ such that $U \geqslant 0$ and $UV = I^N$ is a *distributor* matrix for $\mathcal{P}$. It can be readily seen that to satisfy these conditions U must actually be a matrix of which the elements of the i-th row that correspond to elements in C_i sum up to one while the other elements of the row are 0. For the trivial partitioning $\mathcal{P} = \{S\}$ a distributor is any stochastic row vector; for the trivial partitioning $\mathcal{P} = \{\{i\} \mid i \in S\}$ there exists only one distributor, viz. I.

Example 13.1.1 Let $\mathcal{S} = \{1,2,3\}$ and $\mathcal{P} = \{\{1,2\},\{3\}\}$. Then $V = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$ is the collector for $\mathcal{P}$ and $U = \begin{pmatrix} \frac{1}{3} & \frac{2}{3} & 0 \\ 0 & 0 & 1 \end{pmatrix}$ is an example for a distributor. □

Aggregation by ordinary lumping partitions the state space into classes such that in all the states that are lumped together the process behaves in the same way when transiting to other partitioning classes. It is also required that states in the same lumping class have the same reward. We formalize this in matrix terms.

Definition 13.1.2 (Ordinary lumping) A partitioning $\mathcal{P}$ of $\{1,\ldots,n\}$ is called an *ordinary lumping* of a discontinuous Markov reward chain (σ,Π,Q,ρ) iff the following conditions hold:

$$VU\Pi V = \Pi V, \quad VUQV = QV, \quad \text{and} \quad VU\rho = \rho,$$

where V and U are respectively the collector and some distributor matrix for $\mathcal{P}$. □

The lumping conditions actually assure that the rows of ΠV (resp. QV and ρ) that correspond to the states of the same partitioning class are equal. Their representation in terms of a distributor matrix is convenient since it allows us to write them as matrix equations. We show that, indeed, the lumping conditions do not depend on the particular choice of the non-zero elements of U. Suppose that $VU\Pi V = \Pi V$ and that $U' \geqslant 0$ is such that $U'V = I$. Then $VU'\Pi V = VU'VU\Pi V = VU\Pi V = \Pi V$. Similarly, $VU'QV = QV$ and $VU'\rho = \rho$.

The trivial partitioning $\mathcal{P} = \{\{1\},\ldots,\{n\}\}$ is always an ordinary lumping. The other trivial partitioning $\mathcal{P} = \{S\}$, however, is an ordinary lumping only if the reward structure is trivial, i.e. if the reward vector ρ is comprised of equal elements.

The following theorem characterizes the lumped process, i.e. the process obtained after the aggregation by lumping.

Theorem 13.1.3 (Lumped process) Let (σ, Π, Q, ρ) be a discontinuous Markov reward chain and let $\mathcal{P} = \{C_1, \ldots, C_N\}$ be an ordinary lumping of (σ, Π, Q, ρ). Define

$$\hat{\sigma} = \sigma V, \quad \hat{\Pi} = U\Pi V, \quad \hat{Q} = UQV, \quad \text{and} \quad \hat{\rho} = U\rho.$$

Then $(\hat{\sigma}, \hat{\Pi}, \hat{Q}, \hat{\rho})$ is a discontinuous Markov reward chain. □

Proof First, we have $\hat{\sigma} \cdot \mathbf{1} = \sigma V \cdot \mathbf{1} = \sigma \cdot \mathbf{1} = \mathbf{1}$. Next, we show that the four conditions of Theorem 12.1.2 hold for $\hat{\Pi}$ and $\hat{Q}$.

1. Since $U \geqslant 0$, $V \geqslant 0$ and $\Pi \geqslant 0$, we have $\hat{\Pi} = U\Pi V \geqslant 0$. Also,

   $$\hat{\Pi} \cdot \mathbf{1} = U\Pi V \cdot \mathbf{1} = U\Pi \cdot \mathbf{1} = U \cdot \mathbf{1} = \mathbf{1}$$

   and, since $VU\Pi V = \Pi V$, we have

   $$\hat{\Pi}^2 = U\Pi V U\Pi V = U\Pi\Pi V = U\Pi V = \hat{\Pi}.$$

2. Using the lumping conditions and that $\Pi Q = Q\Pi = Q$, we have

   $$\hat{\Pi}\hat{Q} = U\Pi V UQV = U\Pi QV = UQV$$

   and, similarly,

   $$\hat{Q}\hat{\Pi} = UQVU\Pi V = UQ\Pi V = UQV = \hat{Q}.$$

3. We have

   $$\hat{Q} \cdot \mathbf{1} = UQV \cdot \mathbf{1} = UQ \cdot \mathbf{1} = U \cdot \mathbf{0} = \mathbf{0}.$$

4. Let c be such that $Q + c\Pi \geqslant 0$. Then

   $$\hat{Q} + c\hat{\Pi} = UQV + cU\Pi V = U(Q + c\Pi)V \geqslant 0.$$ ■

When the lumping conditions hold, then the definition of $(\hat{\sigma}, \hat{\Pi}, \hat{Q}, \hat{\rho})$ also does not depend on a particular distributor U. To show this, let U' be another distributor matrix for $\mathcal{P}$. Then $U'\Pi V = U'VU\Pi V = U\Pi V$. Similarly, $U'QV = UQV$ and $U'\rho = U\rho$.

The trivial partitioning $\mathcal{P} = \big\{\{1\}, \ldots, \{n\}\big\}$ leaves the original process intact. The other trivial partitioning, i.e. $\mathcal{P} = \big\{S\big\}$ gives the absorbing, one state, process as result.

If $\mathcal{P}$ is an ordinary lumping of (σ, Π, Q, ρ) and $\hat{\sigma}$, $\hat{\Pi}$, $\hat{Q}$ and $\hat{\rho}$ are defined as in Theorem 13.1.3, then we say that (σ, Π, Q, ρ) *lumps to* $(\hat{\sigma}, \hat{\Pi}, \hat{Q}, \hat{\rho})$ with respect to $\mathcal{P}$ and we write $(\sigma, \Pi, Q, \rho) \overset{\mathcal{P}}{\rightsquigarrow} (\hat{\sigma}, \hat{\Pi}, \hat{Q}, \hat{\rho})$.

Note that if (σ, Π, Q, ρ) is a Markov reward chain, then, since $\Pi = I$, $VU\Pi V = \Pi V$ always holds. Moreover, in this case, also $\hat{\Pi} = U\Pi V = UIV = I$ and so, by Theorem 12.1.2, $\hat{Q}$ is a generator matrix. Therefore, when restricted to the continuous case, our notion of ordinary lumping coincides with the standard definition proposed in [82].

Before we give some examples of ordinary lumping we give two important theorems that give the connection between the lumping and the transition matrix. We prove a lemma first.

Lemma 13.1.4 Let (σ, Π, Q, ρ) be a discontinuous Markov reward chain and let $\mathcal{P}$ be an ordinary lumping. Then,

1. $\Pi Q^n = Q^n$ for all $n \geqslant 1$,
2. $VUQ^nV = Q^nV$ for all $n \geqslant 0$, and
3. $(UQV)^n = UQ^nV$ for all $n \geqslant 0$. □

Proof We prove all the three cases by induction on n.

1. First we have $\Pi Q^1 = \Pi Q = Q = Q^1$ by definition. Assume that $\Pi Q^n = Q^n$ for $n \geqslant 1$. Then
$$\Pi Q^{n+1} = \Pi Q^n Q = Q^n Q = Q^{n+1}.$$
2. For $n = 0$ we have $VUQ^0V = VUV = VI = V = IV = Q^0V$. Assume that $VUQ^nV = Q^nV$ for $n \geqslant 0$. Then,
$$\begin{aligned} VUQ^{n+1}V = VUQ^nQV = VUQ^nVUQV = \\ = Q^nVUQV = Q^nQV = Q^{n+1}V. \end{aligned}$$
3. For $n = 0$ we have $(UQV)^0 = I = UV = UIV = UQ^0V$. Suppose that $(UQV)^n = UQ^nV$ for $n \geqslant 0$. Then
$$(UQV)^{n+1} = (UQV)^n UQV = UQ^nVUQV = UQ^nQV = UQ^{n+1}V.$$
■

The first theorem reflects the conditions of Definition 13.1.2 to the corresponding transition matrix.

Theorem 13.1.5 Let (σ, Π, Q, ρ) be a discontinuous Markov reward chain and let $P(t) = \Pi e^{Qt}$ $(t > 0)$, be its transition matrix. Let $\mathcal{P}$ be an ordinary lumping of (σ, Π, Q, ρ). Then

$$VUP(t)V = P(t)V. \qquad \square$$

Proof We have

$$VUP(t)V = VU\Pi e^{Qt}V = VU\Pi \sum_{n=0}^{\infty} \frac{Q^n t^n}{n!} V =$$
$$= VU\Pi \left(I + \sum_{n=1}^{\infty} \frac{Q^n t^n}{n!} \right) V = VU\Pi V + \sum_{n=1}^{\infty} \frac{VU\Pi Q^n V t^n}{n!}.$$

By Lemma 13.1.4(1), we have $\Pi Q^n = Q^n$ and so $VU\Pi Q^n V = VUQ^n V$. Furthermore, by Lemma 13.1.4(2), we have $VUQ^n V = Q^n V$. Using this and $VU\Pi V = \Pi V$, we continue the derivation as

$$VU\Pi V + \sum_{n=1}^{\infty} \frac{VU\Pi Q^n V t^n}{n!} = \Pi V + \sum_{n=1}^{\infty} \frac{Q^n V t^n}{n!} = \Pi V + \sum_{n=1}^{\infty} \frac{\Pi Q^n V t^n}{n!} =$$
$$= \Pi \left(I + \sum_{n=1}^{\infty} \frac{Q^n t^n}{n!} \right) V = \Pi \sum_{n=0}^{\infty} \frac{Q^n t^n}{n!} V = \Pi e^{Qt} V = P(t)V. \qquad \blacksquare$$

The second theorem shows that the transition matrix of the lumped process can also be obtained directly from the transition matrix of the original process.

Theorem 13.1.6 Let $(\sigma, \Pi, Q, \rho) \overset{\mathcal{P}}{\rightsquigarrow} (\hat{\sigma}, \hat{\Pi}, \hat{Q}, \hat{\rho})$. Let $P(t) = \Pi e^{Qt}$ and $\hat{P}(t) = \hat{\Pi} e^{\hat{Q}t}$ $(t > 0)$ be the transition matrices of (σ, Π, Q, ρ) and $(\hat{\sigma}, \hat{\Pi}, \hat{Q}, \hat{\rho})$ respectively. Then

$$\hat{P}(t) = UP(t)V. \qquad \square$$

Proof First we have

$$\hat{P}(t) = \hat{\Pi} e^{\hat{Q}t} = U\Pi V e^{UQVt} = U\Pi V \sum_{n=0}^{\infty} \frac{(UQV)^n t^n}{n!}.$$

By Lemma 13.1.4(3), we have $(UQV)^n = UQ^n V$, and so

$$U\Pi V \sum_{n=0}^{\infty} \frac{(UQV)^n t^n}{n!} = U\Pi V \sum_{n=0}^{\infty} \frac{UQ^n V t^n}{n!} = U\Pi \sum_{n=0}^{\infty} \frac{VUQ^n V t^n}{n!}.$$

Using Lemma 13.1.4(2), it further follows that

$$U\Pi\sum_{n=0}^{\infty}\frac{VUQ^nVt^n}{n!} = U\Pi\sum_{n=0}^{\infty}\frac{Q^nVt^n}{n!} =$$

$$= U\Pi\Big(\sum_{n=0}^{\infty}\frac{Q^nt^n}{n!}\Big)V = U\Pi e^{Qt}V = UP(t)V. \quad \blacksquare$$

Now we can also prove that the lumped process has the same total reward as the original process. Since the total reward is usually the most useful performance measure, this is a very important property of lumping.

Corollary 13.1.7 Let $(\sigma, \Pi, Q, \rho) \overset{\mathcal{P}}{\rightsquigarrow} (\hat{\sigma}, \hat{\Pi}, \hat{Q}, \hat{\rho})$ and let $R(t)$ and $\hat{R}(t)$ be the total reward of (σ, Π, Q, ρ) and $(\hat{\sigma}, \hat{\Pi}, \hat{Q}, \hat{\rho})$ respectively. Then $\hat{R}(t) = R(t)$.□

Proof Using Theorems 13.1.6 and 13.1.5, we have

$$\hat{R}(t) = \hat{\sigma}\hat{P}(t)\hat{\rho} = \sigma VUP(t)VU\rho = \sigma P(t)VU\rho = \sigma P(t)\rho = R(t). \quad \blacksquare$$

Remark 13.1.8 The definition of the lumped process must be correct according to the standard probabilistic intuition. This means that we need to show that the finite distribution of the lumped process is the same as the sum of the finite distributions of the original process over the states in the lumping classes. That is, we need to prove that the probability that the process is in a finite sequence of classes in a given sequence of time instances, is the same as the sum of the probabilities that the process is in the individual sequences of states from these classes in that time sequence. This can be easily proven (e.g. by induction on the length of the time sequence) using Theorems 13.1.5 and 13.1.6. □

We now give some examples.

Example 13.1.9 a. Let (σ, Π, Q, ρ) be the discontinuous Markov reward chain defined by

$$\sigma = \begin{pmatrix}\pi & 1-\pi & 0\end{pmatrix},\ \Pi = I,\ Q = \begin{pmatrix}-(\lambda+\mu) & \lambda & \mu \\ 0 & 0 & 0 \\ 0 & 0 & 0\end{pmatrix},\ \text{and } \rho = \begin{pmatrix}r_1 \\ r \\ r\end{pmatrix}.$$

This it the discontinuous Markov reward chain from Example 12.1.7a but with $r_2 = r_3 \overset{\text{def}}{=} r$. We show that the partitioning $\mathcal{P} =$

$\{\{1\},\{2,3\}\}$ is an ordinary lumping. From $\mathcal{P}$ we obtain

$$V = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad U = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \alpha & 1-\alpha \end{pmatrix},$$

for some $0 \leqslant \alpha \leqslant 1$. Now, we have

$$VUQV = \begin{pmatrix} -(\lambda{+}\mu) & \lambda{+}\mu \\ 0 & 0 \end{pmatrix} = QV$$

and

$$VU\rho = \begin{pmatrix} r_1 \\ \alpha r + (1{-}\alpha)r \\ \alpha r + (1{-}\alpha)r \end{pmatrix} = \begin{pmatrix} r_1 \\ r \\ r \end{pmatrix} = \rho.$$

The lumped process $(\hat{\sigma}, \hat{\Pi}, \hat{Q}, \hat{\rho})$ is defined by

$$\hat{\sigma} = \begin{pmatrix} 1 & 0 \end{pmatrix}, \; \hat{\Pi} = I, \; \hat{Q} = \begin{pmatrix} -(\lambda{+}\mu) & \lambda{+}\mu \\ 0 & 0 \end{pmatrix} \quad \text{and} \quad \hat{\rho} = \begin{pmatrix} r_1 \\ r \end{pmatrix}.$$

The total reward of the process (σ, Π, Q, ρ) from Example 12.1.7a reduces to $R(t) = r_1 e^{-(\lambda+\mu)t} + r(1 - e^{-(\lambda+\mu)t})$ when $r_2 = r_3 = r$. As proven in Corollary 13.1.7, the same total reward can be calculated by

$$\hat{\sigma}\hat{P}(t)\hat{\rho} = \hat{\sigma} e^{\hat{Q}t}\hat{\rho} = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} e^{-(\lambda+\mu)t} & 1{-}e^{-(\lambda+\mu)t} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} r_1 \\ r \end{pmatrix} =$$

$$= r_1 e^{-(\lambda+\mu)t} + r(1 - e^{-(\lambda+\mu)t}).$$

This example illustrated an ordinary lumping of a standard Markov chain.

b. Let (σ, Π, Q, ρ) be defined by $\sigma = \begin{pmatrix} \pi & 1{-}\pi & 0 \end{pmatrix}$ and

$$\Pi = \begin{pmatrix} 1{-}p & p & 0 \\ 1{-}p & p & 0 \\ 0 & 0 & 1 \end{pmatrix}, \; Q = \begin{pmatrix} -p(1{-}p)\lambda & -p^2\lambda & p\lambda \\ -p(1{-}p)\lambda & -p^2\lambda & p\lambda \\ 0 & 0 & 0 \end{pmatrix}, \text{ and } \rho = \begin{pmatrix} r \\ r \\ r_3 \end{pmatrix}.$$

This is the same discontinuous Markov reward chain as in Example 12.1.7b but with $r_1 = r_2 \stackrel{\text{def}}{=} r$. We show that $\mathcal{P} = \{\{1,2\},\{3\}\}$ is an ordinary lumping. This easily follows after looking at the corresponding rows of ρ and of the following matrices:

$$\Pi V = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, \; QV = \begin{pmatrix} -p\lambda & p\lambda \\ -p\lambda & p\lambda \\ 0 & 0 \end{pmatrix}.$$

The lumped process $(\hat{\sigma}, \hat{\Pi}, \hat{Q}, \hat{\rho})$ is defined by:

$$\hat{\sigma} = \begin{pmatrix} 1 & 0 \end{pmatrix}, \ \hat{\Pi} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \hat{Q} = \begin{pmatrix} -p\lambda & p\lambda \\ 0 & 0 \end{pmatrix} \quad \text{and} \quad \hat{\rho} = \begin{pmatrix} r \\ r_3 \end{pmatrix}.$$

Note that, in this case, the lumped process is a Markov reward chain.

By setting $r_1 = r_2 = r$ in the total reward from Example 12.1.7b we have $R(t) = ((1-p)r_1 + pr_2 - r_3)\, e^{-p\lambda t} + r_3 = (r - r_3)e^{-p\lambda t} + r_3$. We calculate

$$\hat{R}(t) = \hat{\sigma}\hat{P}(t)\hat{\rho} = \hat{\sigma}e^{\hat{Q}t}\hat{\rho} = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} e^{-p\lambda t} & 1-e^{-p\lambda t} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} r \\ r_3 \end{pmatrix} =$$
$$= (r - r_3)e^{-p\lambda t} + r_3 = R(t).$$

In this example a whole ergodic class constitutes a lumping class. It is not hard to show that an ergodic class is always a correct lumping class when the states inside all have the same reward. By lumping a whole ergodic class we obtain a regular state in the lumped process. By observing its entry in $\hat{Q}$ we can see the time that the original process spends switching among the states in this ergodic class. The time is always exponential; in this example it is with rate $p\lambda$.

Note that we always obtain a reward vector with equal elements for states belonging to the same ergodic class after multiplying the original reward vector by Π (cf. Example 12.1.7b). Recall that nothing is lost by this operation if only the total reward is to be calculated.

c. Let (σ, Π, Q, ρ) be defined by $\sigma = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}$,

$$\Pi = \begin{pmatrix} 0 & p & 1-p & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \ Q = \begin{pmatrix} 0 & -p\lambda & -(1-p)\mu & p\lambda + (1-p)\mu \\ 0 & -\lambda & 0 & \lambda \\ 0 & 0 & -\lambda & \lambda \\ \nu & 0 & 0 & -\nu \end{pmatrix},$$

and

$$\rho = \begin{pmatrix} r_1 \\ r \\ r \\ r_4 \end{pmatrix}.$$

The partitioning $\mathcal{P} = \big\{\{1\},\{2,3\},\{4\}\big\}$ is an ordinary lumping and (σ, Π, Q, ρ) lumps (with respect to $\mathcal{P}$) to $(\hat{\sigma}, \hat{\Pi}, \hat{Q}, \hat{\rho})$ defined by:

$$\hat{\sigma} = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}, \ \hat{\Pi} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \ \hat{Q} = \begin{pmatrix} 0 & -\lambda & \lambda \\ 0 & -\lambda & \lambda \\ \nu & 0 & -\nu \end{pmatrix} \quad \text{and} \quad \hat{\rho} = \begin{pmatrix} r_1 \\ r \\ r_4 \end{pmatrix}.$$

This is an example when the lumped process is not a Markov reward chain.

The partitioning $\mathcal{P} = \big\{\{1,2,3\},\{4\}\big\}$ is also an ordinary lumping. With respect to this partitioning (σ, Π, Q, ρ) lumps to $(\hat{\sigma}, \hat{\Pi}, \hat{Q}, \hat{\rho})$ defined as:

$$\hat{\sigma} = \begin{pmatrix} 1 & 0 \end{pmatrix}, \ \hat{\Pi} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \ \hat{Q} = \begin{pmatrix} -\lambda & \lambda \\ \nu & -\nu \end{pmatrix}, \ \text{and } \hat{\rho} = \begin{pmatrix} r \\ r_4 \end{pmatrix}$$

which is a standard Markov reward chain.

This example shows how transient states are lumped together with ergodic states. It is not hard to show that if a transient state can be trapped only in one ergodic class, then it can always be lumped with states from that ergodic class. Note that, when the reward vector is multiplied by Π, the original reward on the transient state becomes irrelevant because it becomes the same as the new reward of the ergodic class. Also, if a transient state can be trapped in more than one ergodic class, and if the lumping class that contains this transient state also contains some states from one of these ergodic classes, then this lumping class must contain states from all of these ergodic classes.

Note that (σ, Π, Q, ρ) is the discontinuous Markov reward chain from Example 12.1.7c when $\lambda = \mu$ and $r_2 = r_3 \stackrel{\text{def}}{=} r$. We show that without these restrictions, the discontinuous Markov reward chain from Example 12.1.7c cannot be properly lumped. State 1 is transient and it can be trapped in the ergodic states 2 and 3. This state can only be joined with state 2 or with state 3 if these two states are both in the same lumping class. States 2 and 3, however, cannot belong to the same class because they either have different rates leading to state 4 or they have different rewards. □

13.2 τ-lumping

In this section we introduce a notion of lumping for Markov reward chains with fast transitions. This notion is based on the ordinary lumping for discontinuous Markov reward chains: a partitioning is a lumping of a Markov reward chain with fast transitions if it is an ordinary lumping of its limit.

Definition 13.2.1 (τ-lumping) A partitioning $\mathcal{P}$ of a Markov reward chain with fast transitions (σ, Q_s, Q_f, ρ) is called a *τ-lumping* if it is an ordinary lumping of the discontinuous Markov chain (σ, Π, Q, ρ), where $(\sigma, Q_s, Q_f, \rho) \to_\infty (\sigma, \Pi, Q, \rho)$. □

We give a definition of the lumped process by multiplying σ, Q_s, Q_f and ρ with the collector matrix and a distributor matrix, similarly as we did for discontinuous Markov chains. This technique ensures that the lumped versions of Q_s and Q_f are also generator matrices and that, consequently, we obtain a Markov reward chain with fast transitions as a result. However, since the lumping condition does not hold for Q_s and Q_f (i.e. we *do not* necessarily have that $VUQ_sV = Q_sV$ and $VUQ_fV = Q_fV$, but only that $VU\Pi V = \Pi V$ and $VUQV = QV$), we cannot guarantee that the definition of the lumped process does not depend on the choice for a distributor. We define a class of special distributors, called τ-distributors, that give a lumped process of which the limit is the lumped version of the limit of the original Markov reward chain with fast transitions.

Before we present the definition of τ-distributors, we state a lemma that provides a connection between a τ-lumping and the ergodic classes. Intuitively, if two lumping classes contain states from a same ergodic class, then whenever one of the lumping classes contains states from another ergodic class, the other must also contain states from that ergodic class.

Lemma 13.2.2 Let (σ, Q_s, Q_f, ρ) be a Markov reward chain with fast transitions. Let $\mathcal{E} = \{E_1, \ldots, E_M, T\}$ be its ergodic partitioning and let $\mathcal{P} = \{C_1, \ldots, C_N\}$ be a τ-lumping. Then, for all $1 \leq I, J \leq M$ and all $1 \leq K, L \leq N$, if $E_I \cap C_K \neq \emptyset$, $E_J \cap C_K \neq \emptyset$ and $E_I \cap C_L \neq \emptyset$, then $E_J \cap C_L \neq \emptyset$. □

Proof Suppose i, j and k are such that $i \in E_I$, $i \in C_K$, $j \in E_J$, $j \in C_K$, $k \in E_I$ and $k \in C_L$. Let Π be the ergodic projection of Q_f. Note first, because k is an ergodic state, from the form of Π, we have that $\Pi[k,k] > 0$. Since $k \in C_L$, this implies that $(\Pi V)[k, L] > 0$. Now, let $\ell \in E_J$. From the form of Π again, it follows that $(\Pi V)[\ell, L] = (\Pi V)[j, L]$ because j and ℓ are in the same ergodic class. Since j and i belong to the same lumping class, we have $(\Pi V)[j, L] = (\Pi V)[i, L]$. As before, by the form of Π, $(\Pi V)[i, L] = (\Pi V)[k, L]$. We conclude that $(\Pi V)[\ell, L] > 0$. This means that there exists an $\ell' \in C_L$ such that $\Pi[\ell, \ell'] > 0$. This is only possible if ℓ' and ℓ are in the same ergodic class, i.e. if $\ell' \in E_J$. We conclude that $\ell' \in E_J \cap C_L$. ■

Now, we can give the definition of a τ-distributor and of a τ-lumped Markov reward chain with fast transitions.

Definition 13.2.3 (τ-distributor) Let (σ, Q_s, Q_f, ρ) be a Markov reward chain with fast transitions. Let $\mathcal{P} = \{C_1, \ldots, C_N\}$ be its τ-lumping and $\mathcal{E} = \{E_1, \ldots, E_M, T\}$ its ergodic partitioning. Let Π be the ergodic projection of Q_f. Put $e(K) = \{L \mid C_K \cap E_L \neq \emptyset\}$. Let $\alpha_{KL} > 0$ if $L \in e(K)$ be arbitrary, subject only to $\sum_{L \in e(K)} \alpha_{KL} = 1$ and $\alpha_{KL} = \alpha_{K'L}$. Let $\beta_{Ki} > 0$ for $i \in C_K$ and $e(K) = \emptyset$ be also arbitrary, subject only to $\sum_{i \in C_K} \beta_{Ki} = 1$. Then a *$\tau$-distributor* $W \in \mathbb{R}^{N \times n}$ is defined as

$$W[K, i] = \begin{cases} 0, & i \notin C_K \\ \alpha_{KL}|e(K)| \frac{\Pi[i,i]}{\sum_{k \in C_K} \Pi[k,k]}, & i \in C_K \cap E_L \\ 0, & i \in C_K \cap T, e(K) \neq \emptyset \\ \beta_{Ki}, & i \in C_K \cap T, e(K) = \emptyset. \end{cases}$$

Define

$$\hat{\sigma} = \sigma V, \quad \hat{Q}_s = W Q_s V, \quad \hat{Q}_f = W Q_f V, \quad \text{and} \quad \hat{\rho} = W \rho,$$

for some τ-distributor W. We say that (σ, Q_s, Q_f, ρ) *τ-lumps* to $(\hat{\sigma}, \hat{Q}_s, \hat{Q}_f, \hat{\rho})$ with respect to $\mathcal{P}$ and write $(\sigma, Q_s, Q_f, \rho) \overset{\mathcal{P}}{\rightsquigarrow}_\tau (\hat{\sigma}, \hat{Q}_s, \hat{Q}_f, \hat{\rho})$. □

Note that $W \geqslant 0$. In the special case that $\alpha_{KL} = 1/|e_K|$ for all $1 \leqslant L \leqslant M$, it is clear that W is indeed a distributor matrix for $\mathcal{P}$. The proof that it is a distributor also in the general case will be given later (see Theorem 13.2.8).

Let us explain the form of a τ-distributor. As a distributor, it is a matrix that assigns weights to the rows of $Q_s V$ and $Q_f V$, and then sums them up. Because of Lemma 13.2.2 the lumping and the ergodic classes can be grouped in such a way that every lumping class shares states with every ergodic class of the group and no other. The group of ergodic classes that have common states with the lumping class C_K are given by $e(K)$. The weights $\alpha_{KL} > 0$, for $L \in e(K)$, can be arbitrarily distributed amongst the ergodic classes that share the same lumping classes. They must sum up to one to ensure the form of a distributor. The condition $\alpha_{KL} = \alpha_{K'L}$ assures that the states from the same ergodic class are treated in the same way (it is because of this condition that Lemma 13.2.2 is crucial for the correct definition of a τ-distributor). The weights are multiplied by $|e(K)|$ because the normalization constant $\sum_{k \in C_k} \Pi[k,k]$ is a sum over all states of the $|e(K)|$ shared ergodic classes. As transient states have no ergodic probabilities ($\Pi[i,i] = 0$ when $i \in T$), they are assigned weight 0 when lumped together with ergodic states. We can assign arbitrary weights when lumping only transient states since by the lumping conditions their trapping probabilities to lumped ergodic classes must be equal.

Note that because there are several choices for the parameters in the definition of τ-distributors, there are, in general, several Markov reward chains with fast transitions that the original Markov reward chain with fast transitions τ-lumps to. We will show later that all these processes are equivalent in the limit and, moreover, that in some special cases, they are exactly equivalent.

We now give some examples; first some in which the τ-lumped process is unique.

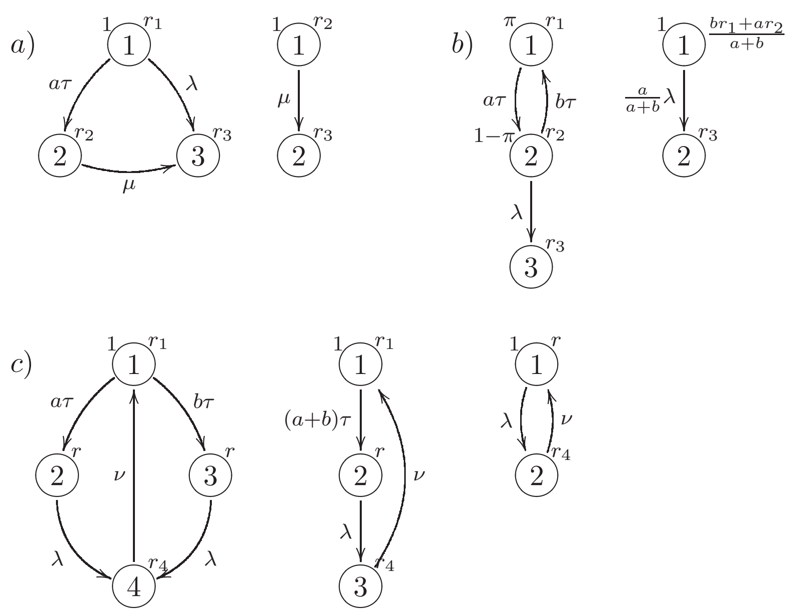


Figure 13.1: τ-lumpings with unique τ-lumped processes – Example 13.2.4

Example 13.2.4 a. Consider the Markov reward chain with fast transitions depicted in Figure 13.1a on the left. Its ergodic partitioning is $\mathcal{E} = \{E_1, E_2, T\}$ with $E_1 = \{2\}$, $E_2 = \{3\}$ and $T = \{1\}$. We show that $\mathcal{P} = \{C_1, C_2\}$, with $C_1 = \{1, 2\}$ and $C_2 = \{3\}$, is a τ-lumping and that the process τ-lumps to the one in Figure 13.1a on the right. To show that the lumping conditions hold we first obtain

$$\Pi = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad V = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Then

$$\Pi V = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, \ \Pi Q_s \Pi V = \begin{pmatrix} -\mu & \mu \\ -\mu & \mu \\ 0 & 0 \end{pmatrix}, \text{ and } \Pi\rho = \begin{pmatrix} r_2 \\ r_2 \\ r_3 \end{pmatrix}.$$

It is clear that the conditions for τ-lumping hold (the rows corresponding to states in the same lumping class are equal).

We now construct a τ-distributor. We have $e(1) = \{1\}$ and $e(2) = \{2\}$. From this, $\alpha_{11} = 1$, $\alpha_{22} = 1$, and there are no other parameters. We now obtain the only τ-distributor

$$W = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

The τ-lumped process is now defined by the following.

$$\hat{\sigma} = \sigma V = \begin{pmatrix} 1 & 0 \end{pmatrix}, \ \hat{Q}_s = W Q_s V = \begin{pmatrix} -\mu & \mu \\ 0 & 0 \end{pmatrix},$$

$$\hat{Q}_f = W Q_f V = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \text{ and } \hat{\rho} = W\rho = \begin{pmatrix} r_2 \\ r_3 \end{pmatrix}.$$

The process $(\hat{\sigma}, \hat{Q}_s, \hat{Q}_f, \rho)$ is indeed the one depicted in Figure 13.1a on the right.

This example illustrates how, in transient states, fast transitions have priority over slow transitions; the transition labeled with λ is irrelevant. Because there is only one τ-distributor, i.e. it does not depend on the parameters, we have a unique τ-lumped process.

b. Consider the Markov reward chain with fast transitions depicted in Figure 13.1b on the left. The limit of this process was calculated in Example 12.2.4b, and in Example 13.1.9b we showed that $\mathcal{P} = \{C_1, C_2\}$, with $C_1 = \{1,2\}$ and $C_2 = \{3\}$, is an ordinary lumping of the limit. By definition, $\mathcal{P}$ is then a τ-lumping.

We construct a τ-distributor. Recall that

$$\Pi = \begin{pmatrix} \frac{b}{a+b} & \frac{a}{a+b} & 0 \\ \frac{b}{a+b} & \frac{a}{a+b} & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

The ergodic partitioning is $\mathcal{E} = \{E_1, E_2\}$ where $E_1 = \{1,2\}$ and $E_2 = \{3\}$. We have $e(1) = \{1\}$ and $e(2) = \{2\}$. From this, $\alpha_{11} = 1$ and $\alpha_{22} = 1$, and we first obtain

$$W = \begin{pmatrix} \frac{b}{a+b} & \frac{a}{a+b} & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

and then

$$\hat{Q}_s = \begin{pmatrix} -\frac{a\,\lambda}{a+b} & \frac{a\,\lambda}{a+b} \\ 0 & 0 \end{pmatrix}, \ \hat{Q}_f = \mathbf{0}, \text{ and } \hat{\rho} = \begin{pmatrix} \frac{br_1+ar_2}{a+b} \\ r_3 \end{pmatrix}.$$

The process τ-lumps to the one in Figure 13.1b on the right. As in the previous case, we only have one τ-distributor, and hence only one τ-lumped process.

This example shows that when two ergodic states with different slow transition rates are lumped together, the resulting state is ergodic and it can perform the same slow transition but with an adapted rate. The example also shows that, in the limit, the Markov reward chain with fast transitions of Figure 13.1b on the left spends an exponentially distributed amount of time with rate $\frac{a\,\lambda}{a+b}$ in the class $\{1,2\}$. This is the time that it spends switching between state 1 and state 2.

c. Consider the Markov reward chain with fast transitions depicted in Figure 13.1c on the left. The limit of this process was calculated in Example 12.2.4c. Example 13.1.9c then shows that the partitionings $\mathcal{P} = \{C_1, C_2, C_3\}$, with $C_1 = \{1\}$, $C_2 = \{2,3\}$, and $C_3 = \{4\}$, and $\mathcal{P} = \{C_1, C_2\}$, with $C_1 = \{1,2,3\}$ and $C_2 = \{4\}$, are τ-lumpings. The ergodic partitioning of this Markov reward chain with fast transitions is $\mathcal{E} = \{E_1, E_2, E_3, T\}$ where $E_1 = \{2\}$, $E_2 = \{3\}$, $E_3 = \{4\}$ and $T = \{1\}$.

For the first partitioning we have $e(1) = \emptyset$, $e(2) = \{1, 2\}$ and $e(3) = \{3\}$. We then have $\alpha_{21} \stackrel{\text{def}}{=} \alpha$ to be an arbitrary number between 0 and 1, $\alpha_{22} = 1 - \alpha_{21} = 1 - \alpha$, and $\alpha_{33} = 1$. This now gives the following τ-distributor and the τ-lumped process:

$$W = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \alpha & 1-\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \hat{Q}_s = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -\lambda & \lambda \\ \nu & 0 & -\nu \end{pmatrix},$$

$$\hat{Q}_f = \begin{pmatrix} -(a+b) & a+b & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \hat{\rho} = \begin{pmatrix} r_1 \\ r \\ r_4 \end{pmatrix}.$$

The τ-lumped process is depicted in Figure 13.1c in the middle. This example shows that τ-lumping need not eliminate all silent transitions. It also shows that even if there are several valid choices for the parameters in τ-distributors, in some cases there is only one possible τ-lumped process.

For the second partitioning we similarly obtain

$$W = \begin{pmatrix} 0 & \alpha & 1-\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \; \hat{Q}_s = \begin{pmatrix} -\lambda & \lambda \\ \nu & -\nu \end{pmatrix}, \; \hat{Q}_f = \mathbf{0}, \; \text{and } \hat{\rho} = \begin{pmatrix} r \\ r_4 \end{pmatrix}.$$

The lumped Markov reward chains with fast transitions is depicted in Figure 13.1c on the right. This example shows how transient states can be lumped with ergodic states, resulting in an ergodic state. □

In the previous example all the lumping classes always contained some ergodic states, and moreover, there were not constructed from states of different ergodic classes. This is why none of the τ-lumped Markov reward chains with fast transitions depended on the particular choice of parameters in the τ-distributor. The next example shows that this is not always the case.

Example 13.2.5 a. Consider the left Markov reward chain with fast transitions depicted in Figure 13.2a on the left. It is defined by

$$\sigma = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}, \; Q_s = \begin{pmatrix} 0 & 0 & 0 & 0 \\ -\lambda & 0 & 0 & \lambda \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

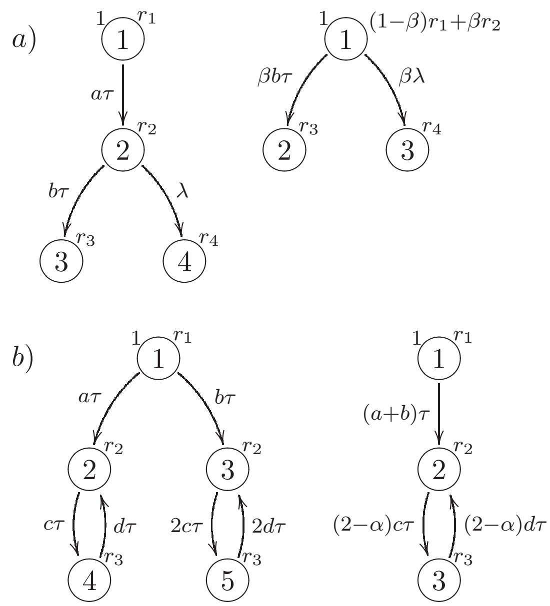


Figure 13.2: τ-lumping where the τ-lumped process depends on the parameters in the τ-distributor – Example 13.2.5

$$Q_f = \begin{pmatrix} -a & a & 0 & 0 \\ 0 & -b & b & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \rho = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{pmatrix}.$$

It is not hard to show that $\mathcal{P} = \{\{1,2\},\{3\},\{4\}\}$ is a τ-lumping of this Markov reward chain with fast transitions. We only show that it τ-lumps to the Markov reward chain with fast transitions depicted in Figure 13.2a on the right. We obtain

$$\Pi = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad V = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

States 1 and 2 are both transient and constitute a lumping class. Because of this we have

$$W = \begin{pmatrix} 1-\beta & \beta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{ for some } 0 < \beta < 1,$$

and so

$$\hat{\sigma} = \sigma V = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}, \; \hat{Q}_s = W Q_s V = \begin{pmatrix} -\beta\lambda & 0 & \beta\lambda \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$\hat{Q}_f = W Q_f V = \begin{pmatrix} -\beta b & \beta b & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \; \hat{\rho} = \begin{pmatrix} (1-\beta) r_1 + \beta r_2 \\ r_3 \\ r_4 \end{pmatrix}.$$

This Markov reward chain with fast transitions is indeed the right one in Figure 13.2a. The reason why it depends on the parameters in W is because there is a lumping class, in this case the first one, that contains transient states only.

b. Consider now the Markov reward chain with fast transitions depicted in Figure 13.2b on the left. It is defined by

$$\sigma = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \; Q_s = \mathbf{0},$$

$$Q_f = \begin{pmatrix} -(a+b) & a & b & 0 & 0 \\ 0 & -c & 0 & c & 0 \\ 0 & 0 & -2c & 0 & 2c \\ 0 & d & 0 & -d & 0 \\ 0 & 0 & 2d & 0 & -2d \end{pmatrix} \quad \text{and} \quad \rho = \begin{pmatrix} r_1 \\ r_2 \\ r_2 \\ r_3 \\ r_3 \end{pmatrix}.$$

It is not hard to show that $\mathcal{P} = \{\{1\}, \{2,3\}, \{4,5\}\}$ is a τ-lumping of this Markov reward chain with fast transitions. We only show that it τ-lumps to the Markov reward chain with fast transitions depicted in Figure 13.2b on the right. We obtain

$$\Pi = \begin{pmatrix} 0 & \frac{a\,d}{(a+b)\,(c+d)} & \frac{b\,d}{(a+b)\,(c+d)} & \frac{a\,c}{(a+b)\,(c+d)} & \frac{b\,c}{(a+b)\,(c+d)} \\ 0 & \frac{d}{c+d} & 0 & \frac{c}{c+d} & 0 \\ 0 & 0 & \frac{d}{c+d} & 0 & \frac{c}{c+d} \\ 0 & \frac{d}{c+d} & 0 & \frac{c}{c+d} & 0 \\ 0 & 0 & \frac{d}{c+d} & 0 & \frac{c}{c+d} \end{pmatrix}.$$

From Π and $\mathcal{P}$ we have

$$W = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha & 1-\alpha & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha & 1-\alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \text{ for some } 0 < \alpha < 1.$$

Note that the same parameter α appears, both in the row corresponding to class $\{2,3\}$ and in the row corresponding to $\{4,5\}$. This is because these two classes belong to the same group, i.e. they share states with the same ergodic classes.

Now, $\hat{\sigma} = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}$, $\hat{Q}_s = \mathbf{0}$,

$$\hat{Q}_f = \begin{pmatrix} -(a+b) & a+b & 0 \\ 0 & -(2-\alpha)c & (2-\alpha)c \\ 0 & (2-\alpha)d & -(2-\alpha)d \end{pmatrix}, \quad \text{and} \quad \hat{\rho} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix}.$$

This Markov reward chain with fast transitions is indeed the one in Figure 13.2b on the right. The reason why it depends on the parameters in W is because the second and the third lumping class contain states from multiple ergodic classes, but do not contain complete ergodic classes. □

The following example shows some Markov reward chains with fast transitions that are minimal in the sense that they only admit the trivial τ-lumpings.

Example 13.2.6 We show that, in non-special cases, the Markov reward chains with fast transitions from Figure 13.3 admit only the trivial τ-lumpings regardless of the reward structure. For this reason the rewards are omitted from the picture.

a. Consider the Markov reward chain with fast transitions depicted in Figure 13.3a. Its limit was obtained in Example 12.1.7c and in Example 13.1.9c we explained why the limit does not have a non-trivial lumping when $\lambda \neq \mu$. Therefore, by the definition of τ-lumping, the Markov reward chain with fast transitions from Figure 13.3a has no proper τ-lumpings when $\lambda \neq \mu$.

b. Consider the Markov reward chain with fast transitions from Figure 13.3b. We show that states 1 and 2 cannot be in the same lumping class. Let $\mathcal{P} = \{\{1,2\},\{3\},\{4\}\}$. We obtain

$$\Pi = \begin{pmatrix} 0 & 0 & \frac{a\,c+b\,(c+d)}{(a+b)\,(c+d)} & \frac{a\,d}{(a+b)\,(c+d)} \\ 0 & 0 & \frac{c}{c+d} & \frac{d}{c+d} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad V = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Then

$$\Pi V = \begin{pmatrix} 0 & \frac{a\,c+b\,(c+d)}{(a+b)\,(c+d)} & \frac{a\,d}{(a+b)\,(c+d)} \\ 0 & \frac{c}{c+d} & \frac{d}{c+d} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

In order for the lumping condition to hold for Π we must have $\frac{a\,d}{(a+b)\,(c+d)} = \frac{d}{c+d}$ which is impossible because $\frac{a}{a+b} < 1$ always.

States 3 and 4 can be in the same lumping class only if $\lambda = \mu$. It is also easy to see that states 2 and 3 cannot be in the same class because otherwise $\frac{c}{c+d} = 1$ which is impossible.

c. Consider the Markov reward chain with fast transitions in Figure 13.3c. This Markov reward chain with fast transitions has a nontrivial lumping only when $b = c$ (with the assumption that $\lambda \neq \mu$). We show that states 1 and 2 can be in the same lumping class only in this case. Let $\mathcal{P} = \{\{1,2\},\{3\},\{4\}\}$. We obtain

$$\Pi = \begin{pmatrix} 0 & 0 & \frac{a}{a+b} & \frac{b}{a+b} & 0 \\ 0 & 0 & \frac{a}{a+c} & \frac{c}{a+c} & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad V = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

As in the previous example for the lumping condition to hold we must have that $\frac{a}{a+b} = \frac{a}{a+c}$. This is only possible when $b = c$. $\square$

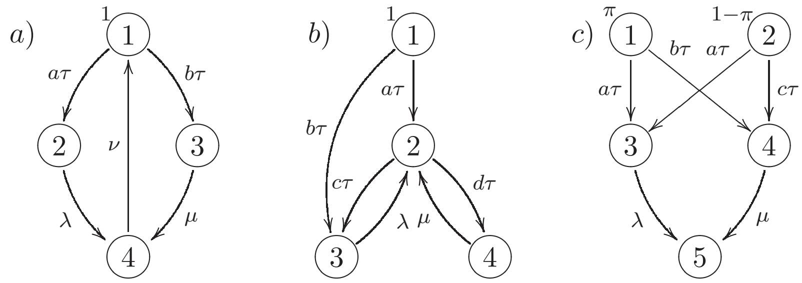


Figure 13.3: Markov reward chains with fast transitions without non-trivial τ-lumpings – Example 13.2.6

Definition 13.2.3 of τ-lumping and Definition 13.1.2 of ordinary lumping induce the following diagram:

Markov Reward Chain with Fast Transitions	$\xrightarrow[\tau\to\infty]{}$	Discontinuous Markov Reward Chain
↓ *τ-lumping*		↓ *ordinary lumping*
τ-lumped Markov Reward Chain with Fast Transitions		lumped Discontinuous Markov Reward Chain

We now show that the diagram can be closed, i.e. that

τ-lumped Markov Reward Chain with Fast Transitions	$\xrightarrow[\tau\to\infty]{}$	lumped Discontinuous Markov Reward Chain

This property is very important since it proves the definition of τ-lumping correct by showing that τ-lumping preserves limit behavior.

To establish correctness we first show that $\Pi V W \Pi = \Pi V W$. Intuitively, this equality states that W distributes the lumped ergodic states according to their re-normalized ergodic probabilities. For a smooth proof of this property we introduce a convenient numbering of states. This numbering also allows us to prove that W is a distributor for any choice of the parameters.

By Lemma 13.2.2 we can introduce a convenient arrangement of the ergodic and lumping classes.

Let $\mathcal{E} = \{E_1, \ldots, E_M, T\}$ and $\mathcal{P} = \{C_1, \ldots, C_N\}$ be the ergodic partitioning and a τ-lumping respectively of some Markov reward chain with fast transitions. Let $1 \leqslant L \leqslant N$ be the number of lumping classes that contain ergodic states and let the lumping classes be arranged such that $C_1, \ldots, C_L$ contain states from ergodic classes (and possibly some transient states too), while $C_{L+1}, \ldots, C_N$ consist exclusively of transient states. Then, there exist $1 \leqslant S \leqslant \min(L, M)$, $c_1, \ldots, c_S$, and $e_1, \ldots, e_S$, such that $L = \sum_{i=1}^{S} c_i$ and that $C_1, \ldots, C_L$ and $E_1, \ldots, E_M$ can be further arranged and divided into S blocks $E_{i1}, \ldots, E_{ie_i}$ and $C_{i1}, \ldots, C_{ic_i}$ where, for all $1 \leq j \leq e_i$, $1 \leq k \leq c_i$, $E_{ij} \cap C_{ik} \neq \emptyset$, and that E_{ij} has no common elements with other lumping classes.

We further number the states to make the above arrangement explicit (assuming the lexicographic order). Additionally, we divide transient states into those that are lumped together with some ergodic states and those that are lumped only with other transient states, and then number them so that those that belong to the first group precede those from the second group. We give an example of this (re)numbering.

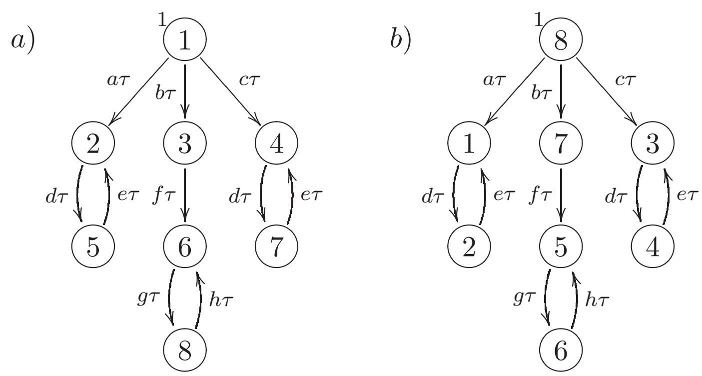


Figure 13.4: Markov reward chain with fast transitions before and after the renumbering of states – Example 13.2.7

Example 13.2.7 Consider the Markov reward chain with fast transitions depicted in Figure 13.4a (we omit the reward structure, but assume that the reward vector is permuted accordingly). Then $\mathcal{E} = \{E_1, E_2, E_3, T\}$, with $E_1 = \{2, 5\}$, $E_2 = \{6, 8\}$, $E_3 = \{4, 7\}$ and $T = \{1, 3\}$, is its ergodic partitioning. It is not hard to show that the partitioning $\mathcal{P} = \{C_1, C_2, C_3, C_4\}$, where $C_1 = \{1\}$, $C_2 = \{2, 4\}$, $C_3 = \{5, 7\}$ and $C_4 = \{3, 6, 8\}$, is a τ-lumping. Note that the ergodic classes E_1 and E_3 share states from the lumping classes C_2 and C_3, and that E_2 shares states only with C_4. So, $L = 3$ and $S = 2$. We now renumber ergodic and lumping classes as $E_1 \mapsto E_{11}$, $E_3 \mapsto E_{12}$, $C_2 \mapsto C_{11}$, $C_3 \mapsto C_{12}$, $E_2 \mapsto E_{21}$, $C_4 \mapsto C_{21}$ and $C_1 \mapsto C_3$. Note that the transient state 3 lumps together with the ergodic states 6 and 8, and that the transient state 1 lumps alone. We renumber states as $2 \mapsto 1$, $5 \mapsto 2$, $4 \mapsto 3$, $7 \mapsto 4$, $6 \mapsto 5$, $8 \mapsto 6$, $3 \mapsto 7$, and $1 \mapsto 8$. The permuted Markov reward chain with fast transitions is depicted in 13.4b. □

We now present the matrices Π, V and W in the new numbering. First we have

$$\Pi = \begin{pmatrix} \Pi_1 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Pi_2 & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \Pi_S & \mathbf{0} & \mathbf{0} \\ \overline{\Pi}_1 & \overline{\Pi}_2 & \dots & \overline{\Pi}_S & \mathbf{0} & \mathbf{0} \\ \widetilde{\Pi}_1 & \widetilde{\Pi}_2 & \dots & \widetilde{\Pi}_S & \mathbf{0} & \mathbf{0} \end{pmatrix} \quad \begin{array}{ll} \Pi_i = \mathrm{diag}\left(\Pi_{i1}, \dots, \Pi_{ie_i}\right) & \Pi_{ij} = \mathbf{1}^{|E_{ij}|} \cdot \mu_{ij} \\ \overline{\Pi}_i = \left(\overline{\Pi}_{i1} \ \dots \ \overline{\Pi}_{ie_i}\right) & \overline{\Pi}_{ij} = \overline{\delta}_{ij} \cdot \mu_{ij} \\ \widetilde{\Pi}_i = \left(\widetilde{\Pi}_{i1} \ \dots \ \widetilde{\Pi}_{ie_i}\right) & \widetilde{\Pi}_{ij} = \widetilde{\delta}_{ij} \cdot \mu_{ij}. \end{array}$$

The matrices Π_i correspond to the groups of classes that share states with the same ergodic classes. The vector μ_{ij} is the ergodic probability vector for the ergodic class E_{ij}. The matrices $\overline{\Pi}_i$ and $\widetilde{\Pi}_i$ respectively correspond to the transient states that are lumped together with ergodic classes and to those that are lumped only with other transient states. The vectors $\overline{\delta}_{ij}$ and $\widetilde{\delta}_{ij}$ are the corresponding restrictions of the vector δ_{ij}, the vector of trapping probabilities for the ergodic class E_{ij}.

The collector matrix V associated with $\mathcal{P}$ has the following form:

$$V = \begin{pmatrix} V_1 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & V_2 & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & V_S & \mathbf{0} \\ \overline{V}_1 & \overline{V}_2 & \dots & \overline{V}_S & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \widetilde{V} \end{pmatrix} \quad V_i = \begin{pmatrix} V_{i1} \\ \vdots \\ V_{ie_i} \end{pmatrix}$$

$$\begin{aligned} V_{ij} &= \mathrm{diag}\left(\mathbf{1}^{|E_{ij} \cap C_{i1}|}, \dots, \mathbf{1}^{|E_{ij} \cap C_{ic_i}|}\right) \\ \overline{V}_i &= \mathrm{diag}\left(\mathbf{1}^{|T \cap C_{i1}|}, \dots, \mathbf{1}^{|T \cap C_{ic_i}|}\right) \\ \widetilde{V} &= \mathrm{diag}\left(\mathbf{1}^{|T \cap C_{L+1}|}, \dots, \mathbf{1}^{|T \cap C_N|}\right). \end{aligned}$$

Note that V_i and $\widetilde{V}$ are always collector matrices. The matrices $\overline{V}_i$ are not necessarily collectors; they are allowed to have zero columns.

Let $\mu_{ij}^{(k)}$ denote the restriction of μ_{ij} to the elements of C_{ik}. The vector $\mu_{ij}^{(k)}$ is never empty because $C_{ik} \cap E_{ij} \neq \emptyset$. Then we can express $\Pi_i V_i$ in terms of these vectors as follows:

$$\Pi_i V_i = \begin{pmatrix} \Pi_{i1} V_{i1} \\ \vdots \\ \Pi_{ie_i} V_{ie_i} \end{pmatrix} = \begin{pmatrix} \mathbf{1}^{|E_{i1}|} \cdot \mu_{i1}^{(1)} \cdot \mathbf{1} & \dots & \mathbf{1}^{|E_{i1}|} \cdot \mu_{i1}^{(c_i)} \cdot \mathbf{1} \\ \vdots & & \vdots \\ \mathbf{1}^{|E_{ie_i}|} \cdot \mu_{ie_i}^{(1)} \cdot \mathbf{1} & \dots & \mathbf{1}^{|E_{ie_i}|} \cdot \mu_{ie_i}^{(c_i)} \cdot \mathbf{1} \end{pmatrix}.$$

From the lumping condition it follows that the rows of $\Pi_i V_i$ that correspond to the same lumping class are equal. This implies that

$$\mu_{ij}^{(\ell)} \cdot \mathbf{1} = \mu_{ik}^{(\ell)} \cdot \mathbf{1},$$

for all $1 \leq j, k \leq e_i$, $1 \leq \ell \leq c_i$. Define a row vector $\phi_i \in \mathbb{R}^{1 \times c_i}$ as

$$\phi_i[\ell] = \mu_{ij}^{(\ell)} \cdot \mathbf{1}$$

(for any $1 \leq j \leq e_i$). Then

$$\mu_{ij} V_{ij} = \phi_i \text{ for every } 1 \leq j \leq e_i, \text{ and } \Pi_i V_i = \mathbf{1} \cdot \phi_i.$$

The matrix W of Definition 13.2.3 has the following form:

$$W = \begin{pmatrix} W_1 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & W_2 & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & W_S & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \widetilde{W} \end{pmatrix} \quad \begin{array}{l} W_i = \begin{pmatrix} W_{i1} & \dots & W_{ie_i} \end{pmatrix} \\ \widetilde{W} = \operatorname{diag}(\widetilde{w}_{L+1}, \dots, \widetilde{w}_N) \end{array}$$

where

$$W_{ij} = \operatorname{diag}\left(\frac{\alpha_{ij} e_i \mu_{ij}^{(1)}}{\sum_{k=1}^{e_i} \mu_{ik}^{(1)} \cdot \mathbf{1}}, \dots, \frac{\alpha_{ij} e_i \mu_{ij}^{(c_i)}}{\sum_{k=1}^{e_i} \mu_{ik}^{(c_i)} \cdot \mathbf{1}} \right).$$

and

$$\widetilde{w}_i = \begin{pmatrix} \beta_{i1} & \dots & \beta_{i|C_i|} \end{pmatrix}, \ 0 < \beta_{ij} < 1.$$

Using the definition of ϕ_i, we have:

$$\begin{aligned} W_{ij} &= \operatorname{diag}\left(\frac{\alpha_{ij} e_i \mu_{ij}^{(1)}}{\sum_{k=1}^{e_i} \mu_{ik}^{(1)} \cdot \mathbf{1}}, \dots, \frac{\alpha_{ij} e_i \mu_{ij}^{(c_i)}}{\sum_{k=1}^{e_i} \mu_{ik}^{(c_i)} \cdot \mathbf{1}}\right) \\ &= \alpha_{ij} e_i \cdot \operatorname{diag}\left(\frac{\mu_{ij}^{(1)}}{\sum_{k=1}^{e_i} \phi_i[1]}, \dots, \frac{\mu_{ij}^{(c_i)}}{\sum_{k=1}^{e_i} \phi_i[c_i]}\right) \\ &= \frac{\alpha_{ij} e_i}{e_i} \cdot \operatorname{diag}\left(\frac{\mu_{ij}^{(1)}}{\phi_i[1]}, \dots, \frac{\mu_{ij}^{(c_i)}}{\phi_i[c_i]}\right) \\ &= \alpha_{ij} \cdot \operatorname{diag}\left(\frac{\mu_{ij}^{(1)}}{\phi_i[1]}, \dots, \frac{\mu_{ij}^{(c_i)}}{\phi_i[c_i]}\right). \end{aligned}$$

Let us now prove that every τ-distributor is a distributor.

Theorem 13.2.8 Let W be a τ-distributor as defined in Definition 13.2.3. Then W is a distributor. □

Proof That $W \geqslant 0$ follows directly from Definition 13.2.3; we only prove that $WV = I$. Using the above forms for W and V, we have

$$WV = \begin{pmatrix} W_1V_1 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & W_2V_2 & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & W_SV_S & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \widetilde{W}\widetilde{V} \end{pmatrix}, \quad W_iV_i = \textstyle\sum_{j=1}^{e_i} W_{ij}V_{ij}$$

and

$$\widetilde{W}\widetilde{V} = \operatorname{diag}\left(\widetilde{w}_{L+1} \cdot \mathbf{1}^{|T \cap C_{L+1}|}, \dots, \widetilde{w}_N \cdot \mathbf{1}^{|T \cap C_N|}\right).$$

We first have

$$\begin{aligned} W_{ij}V_{ij} &= \alpha_{ij} \cdot \operatorname{diag}\left(\frac{\mu_{ij}^{(1)}}{\phi_i[1]}, \dots, \frac{\mu_{ij}^{(c_i)}}{\phi_i[c_i]}\right) \cdot \operatorname{diag}(\mathbf{1}, \dots, \mathbf{1}) \\ &= \alpha_{ij} \cdot \operatorname{diag}\left(\frac{\mu_{ij}^{(1)} \cdot \mathbf{1}}{\phi_i[1]}, \dots, \frac{\mu_{ij}^{(c_i)} \cdot \mathbf{1}}{\phi_i[c_i]}\right) \\ &= \alpha_{ij} \cdot \operatorname{diag}\left(\frac{\phi_i[1]}{\phi_i[1]}, \dots, \frac{\phi_i[c_i]}{\phi_i[c_i]}\right) = \alpha_{ij} I. \end{aligned}$$

Now, using that $\sum_{j=1}^{e_i} \alpha_{ij} = 1$,

$$W_i V_i = \sum_{j=1}^{e_i} W_{ij} V_{ij} = \sum_{j=1}^{e_i} \alpha_{ij} I = I.$$

Also, for all $L+1 \leqslant K \leqslant N$,

$$\widetilde{w}_K \cdot \mathbf{1}^{|T \cap C_K|} = \sum_{k=1}^{|C_K|} \beta_{ik} = 1.$$

■

We now prove an important property of a τ-distributor.

Lemma 13.2.9 Let Π, V and W be as in Definition 13.2.3. Then

$$\Pi V W \Pi = \Pi V W.$$

□

Proof Using the structure of Π, V and W, after a simple block-matrix calculation it follows that $\Pi V W \Pi = \Pi V W$ iff, for all $1 \leqslant i \leqslant S$,

$$X_i V_i W_i \Pi_i = X_i V_i W_i \text{ for } X_i \in \{\Pi_i, \overline{\Pi}_i, \widetilde{\Pi}_i\}.$$

Going one level deeper in the matrix structure, we obtain that $X_i V_i W_i \Pi_i = X_i V_i W_i$ iff

$$\mu_{ij} V_{ij} W_{ik} \Pi_{ik} = \mu_{ij} V_{ij} W_{ik}$$

for all $1 \leqslant j, k \leqslant e_i$. Furthermore, from the definition of ϕ_i it follows that

$$\begin{aligned} \mu_{ij} V_{ij} &= \left(\mu_{ij}^{(1)} \ \ldots \ \mu_{ij}^{(c_i)}\right) \cdot \operatorname{diag}\left(\mathbf{1}^{|E_{ij} \cap C_{i1}|}, \ldots, \mathbf{1}^{|E_{ij} \cap C_{ic_i}|}\right) = \\ &= \left(\mu_{ij}^{(1)} \cdot \mathbf{1}^{|E_{ij} \cap C_{i1}|} \ \ldots \ \mu_{ij}^{(c_i)} \cdot \mathbf{1}^{|E_{ij} \cap C_{ic_i}|}\right) = \left(\phi_i[1] \ \ldots \ \phi_i[c_i]\right) = \phi_i. \end{aligned}$$

Therefore, the equality $\mu_{ij} V_{ij} W_{ik} \Pi_{ik} = \mu_{ij} V_{ij} W_{ik}$ holds iff

$$\phi_i W_{ik} \Pi_{ik} = \phi_i W_{ik}$$

holds. We first calculate

$$\phi_i W_{ik} = (\phi_i[1] \ldots \phi_i[c_i]) \cdot \alpha_{ik} \cdot \operatorname{diag}\left(\frac{\mu_{ik}^{(1)}}{\phi_i[1]}, \ldots, \frac{\mu_{ik}^{(c_i)}}{\phi_i[c_i]}\right) = \alpha_{ik} \cdot \mu_{ik},$$

and then

$$\phi_i W_{ik} \Pi_{ik} = \alpha_{ik} \cdot \mu_{ik} \cdot \mathbf{1} \cdot \mu_{ik} = \alpha_{ik} \cdot 1 \cdot \mu_{ik} = \phi_i W_{ik}.$$

■

It is not hard to show that the converse of this lemma also holds in a special case. Any distributor W that has only non-zero elements associated to the transient states that are lumped only with other transient states, and that satisfies $\Pi V W \Pi = \Pi V W$, is a τ-distributor.

The property $\Pi V W \Pi = \Pi V W$ is crucial in the proof that $\hat{Q}_s$ and $\hat{\rho}$ are correctly defined. We now introduce some notions and prove a lemma that plays an important role in the proof that $\hat{Q}_f$ is also correctly defined.

A matrix $G \in \mathbb{R}^{n \times n}$ such that $G \cdot \mathbf{1} \leqslant \mathbf{0}$ and $G + cI \geqslant \mathbf{0}$ for some $c > 0$ is called a *semi-generator* (matrix). In other words, a semi-generator is a matrix in which a negative element can only be on the diagonal, and the absolute value of this element is bigger than or equal to the sum of the other elements in the row. A semi-generator is called *indecomposable* if it cannot be represented (after any permutation) as $\left(\begin{smallmatrix} Q & \mathbf{0} \\ X & Y \end{smallmatrix}\right)$ where Q is a generator matrix.

Lemma 13.2.10 Let $G \in \mathbb{R}^{n \times n}$ be an indecomposable semi-generator. Then

a. G is invertible, i.e. of full rank; and

b. $UGV \in \mathbb{R}^{N \times N}$ is an indecomposable semi-generator for any collector matrix $V \in \mathbb{R}^{n \times N}$ and any distributor $U \in \mathbb{R}^{N \times n}$ associated to V such that $V[i, K] = 1$ implies $U[K, i] > 0$, for all $1 \leqslant i \leqslant n$ and $1 \leqslant K \leqslant N$. □

Proof a. Suppose that G is not invertible. We construct a numbering in which G= $\left(\begin{smallmatrix} Q & \mathbf{0} \\ X & Y \end{smallmatrix}\right)$ and Q is a generator matrix. Let $r_1, \ldots r_n \in \mathbb{R}^{1 \times n}$ be the row vectors that correspond to the rows of G. Let the rows with elements that sum up to 0 precede those of which this sum is less than 0, i.e. let the numbering of states be such that, for some $1 \leqslant k \leqslant n$, we have $r_i \cdot \mathbf{1} = 0$, for $1 \leqslant i \leqslant k$, and $r_i \cdot \mathbf{1} < 0$, for $k+1 \leqslant i \leqslant n$. Since G is not invertible, there exists an $1 \leqslant \ell \leqslant n$ such that $\alpha_\ell r_\ell = \alpha_1 r_1 + \cdots + \alpha_{\ell-1} r_{\ell-1} + \alpha_{\ell+1} r_{\ell+1} + \cdots + \alpha_n r_n$ for some $\alpha_1, \ldots, \alpha_n$ with $\alpha_\ell = 1$. We can now apply Theorem 2.1 of [45] which imposes restrictions on rows of a singular diagonally-dominant matrix G that are in the span of the other rows. By this theorem we directly have that $r_\ell \cdot \mathbf{1} = 0$, i.e. that $\ell \leqslant k$, that $\alpha_i = 0$ for all $k+1 \leqslant i \leqslant n$, and that $G[i, j] = 0$ for all $1 \leqslant i \leqslant k$ and all $k+1 \leqslant j \leqslant n$. This means that $G = \left(\begin{smallmatrix} Q & \mathbf{0} \\ X & Y \end{smallmatrix}\right)$ where $Q = \begin{pmatrix} r_1 \\ \vdots \\ r_k \end{pmatrix}$ satisfies $Q \cdot \mathbf{1} = \mathbf{0}$ and hence is a generator matrix.

b. The proof is by contraposition. Suppose that in some numbering of classes $UGV = \left(\begin{smallmatrix} Q & \mathbf{0} \\ X & Y \end{smallmatrix}\right)$ and Q is a generator matrix. Assume that the states are numbered such that those that belong to classes that correspond to Q precede the other states. Then

$$UGV = \begin{pmatrix} U_1 & \mathbf{0} \\ \mathbf{0} & U_2 \end{pmatrix} \begin{pmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{pmatrix} \begin{pmatrix} V_1 & \mathbf{0} \\ \mathbf{0} & V_2 \end{pmatrix} = \begin{pmatrix} Q & \mathbf{0} \\ X & Y \end{pmatrix},$$

which in turn implies $U_1 G_{11} V_1 = Q$ and $U_1 G_{12} V_2 = \mathbf{0}$.

We first prove that $G_{12} = \mathbf{0}$. Multiplying the equation $U_1 G_{12} V_2 = \mathbf{0}$ from the right by $\mathbf{1}$ we obtain $U_1 G_{12} \cdot \mathbf{1} = \mathbf{0}$. Define $x \in \mathbb{R}^n$ by $x = G_{12} \cdot \mathbf{1}$. Since $G_{12} \geqslant \mathbf{0}$, also $x \geqslant \mathbf{0}$. Suppose $x[k] > 0$ for some $1 \leqslant k \leqslant n$. Then from $U_1 x = \mathbf{0}$ it follows that $U[K,k] = 0$ for all $1 \leqslant K \leqslant N$. This is not possible because of the requirement that $U[K,k] > 0$ for the index K such that $V[k,K] = 1$. We conclude that $x = \mathbf{0}$ which implies $G_{12} = \mathbf{0}$.

We now prove that G_{11} is a generator matrix. Note that it is a semi-generator, so we only need to show that $G_{11} \cdot \mathbf{1} = \mathbf{0}$. Multiplying the equation $U_1 G_{11} V_1 = Q$ from the right by $\mathbf{1}$ we obtain $U_1 G_{11} \cdot \mathbf{1} = Q \cdot \mathbf{1} = \mathbf{0}$ because Q is a generator. Define $x \in \mathbb{R}^n$ by $x = G_{11} \cdot \mathbf{1}$. Note that $x \leqslant 0$. Suppose $x[k] < 0$ for some $1 \leqslant k \leqslant n$. Since $U_1 x = \mathbf{0}$ it follows that $U[K,k] = 0$ for all $1 \leqslant K \leqslant N$. As in the previous case, this is not possible because $U[K,k] > 0$ when $V[k,K] = 1$. We conclude that $x = \mathbf{0}$ and, therefore, that G_{11} is a generator. ■

The second notion we introduce is the notion of irreducible generator. A matrix is called *irreducible* if there is no permutation after which it is represented as $\left(\begin{smallmatrix} A' & A'' \\ \mathbf{0} & B \end{smallmatrix}\right)$ for some (non-empty) square matrices A' and B.

Lemma 13.2.11 Let $Q \in \mathbb{R}^{n \times n}$ be an irreducible generator matrix. Then $UQV \in \mathbb{R}^{N \times N}$ is also an irreducible generator matrix for any collector matrix $V \in \mathbb{R}^{n \times N}$, and any distributor $U \in \mathbb{R}^{N \times n}$ associated to V such that $V[i,K] = 1$ implies $U[K,i] > 0$, for all $1 \leqslant i \leqslant n$ and $1 \leqslant K \leqslant N$. □

Proof The proof is by contraposition. Suppose that $\hat{Q} = UQV$ is not irreducible. Then $\hat{Q} = \left(\begin{smallmatrix} \hat{Q}'_1 & \hat{Q}''_1 \\ \mathbf{0} & \hat{Q}_2 \end{smallmatrix}\right)$ in some numbering of classes. After an adequate renumbering of states we have

$$UQV = \begin{pmatrix} U_1 & \mathbf{0} \\ \mathbf{0} & U_2 \end{pmatrix} \begin{pmatrix} Q'_1 & Q''_1 \\ Q'_2 & Q''_2 \end{pmatrix} \begin{pmatrix} V_1 & \mathbf{0} \\ \mathbf{0} & V_2 \end{pmatrix} = \begin{pmatrix} \hat{Q}'_1 & \hat{Q}''_1 \\ \mathbf{0} & \hat{Q}_2 \end{pmatrix}$$

which implies that $U_2 Q_2' V_1 = \mathbf{0}$. Since $Q_2' \geqslant 0$, after the same reasoning as in the proof of Lemma 13.2.10, we obtain that $Q_2' = \mathbf{0}$. From this it follows that Q is not irreducible. ■

We are now ready for the correctness proof.

Theorem 13.2.12 Let (σ, Q_s, Q_f, ρ) be a Markov reward chain with fast transitions. Suppose $(\sigma, Q_s, Q_f, \rho) \overset{\mathcal{P}}{\rightsquigarrow}_\tau (\hat{\sigma}, \hat{Q}_s, \hat{Q}_f, \hat{\rho})$, $(\sigma, Q_s, Q_f, \rho) \to_\infty (\sigma, \Pi, Q, \rho')$ and $(\sigma, \Pi, Q, \rho') \overset{\mathcal{P}}{\rightsquigarrow} (\sigma, \hat{\Pi}, \hat{Q}, \hat{\rho}')$. Then

$$(\sigma, Q_s, Q_f, \rho) \to_\infty (\sigma, \hat{\Pi}, \hat{Q}, \hat{\rho}'). \qquad \square$$

Proof We need to show that $\hat{\Pi}$ is the ergodic projection of $\hat{Q}_f$, that $\hat{\Pi}\hat{Q}_s\hat{\Pi} = \hat{Q}$ and that $\hat{\Pi}\hat{\rho} = \hat{\rho}'$.

For the second part, using the lumping conditions and the property $\Pi V W = \Pi V W \Pi$ proven in Lemma 13.2.9, we have the following derivations:

$$\hat{\Pi}\hat{Q}_s\hat{\Pi} = U\Pi V W Q_s V U \Pi V = U \Pi V W \Pi Q_s \Pi V = \\ = U\Pi V W Q V = U \Pi Q V = U Q V = \hat{Q},$$

and, since $\rho' = \Pi\rho$, we have $\Pi\rho' = \rho'$, and then

$$\hat{\Pi}\hat{\rho} = U\Pi V W \rho = U\Pi V W \Pi \rho = U \Pi V W \rho' = U \Pi \rho' = U\rho' = \hat{\rho}'.$$

It remains to show that $\hat{\Pi}$ is the ergodic projection of $\hat{Q}_f$. By Theorem 12.2.3 it is enough to show that $\hat{\Pi} \geqslant 0$, $\hat{\Pi} \cdot \mathbf{1} = \mathbf{1}$, $\hat{\Pi}^2 = \hat{\Pi}$, $\hat{\Pi}\hat{Q}_f = \hat{Q}_f\hat{\Pi} = \mathbf{0}$, and $\mathrm{rank}(\hat{\Pi}) + \mathrm{rank}(\hat{Q}_f) = N$. In Theorem 13.1.3 we showed that $\hat{\Pi}$ satisfies the conditions of Theorem 12.1.2, so we have $\hat{\Pi} \geqslant 0$, $\hat{\Pi} \cdot \mathbf{1} = \mathbf{1}$ and $\hat{\Pi}^2 = \hat{\Pi}$. We also derive

$$\hat{\Pi}\hat{Q}_f = U\Pi V W Q_f V = U \Pi V W \Pi Q_f V = \mathbf{0}$$

using that $\Pi Q_f = \mathbf{0}$. Similarly,

$$\hat{Q}_f\hat{\Pi} = W Q_f V U \Pi V = W Q_f \Pi V = \mathbf{0}$$

because $Q_f\Pi = \mathbf{0}$. We prove that $\mathrm{rank}(\hat{\Pi}) + \mathrm{rank}(\hat{Q}_f) = N$.

First, we compute $\hat{\Pi}$:

$$\hat{\Pi} = W\Pi V = \begin{pmatrix} W_1\Pi_1V_1 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & W_2\Pi_2V_2 & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & W_S\Pi_SV_S & \mathbf{0} \\ \widetilde{W}\widetilde{\Pi}_1V_1 & \widetilde{W}\widetilde{\Pi}_2V_2 & \dots & \widetilde{W}\widetilde{\Pi}_SV_S & \mathbf{0} \end{pmatrix}$$

where $W_i \Pi_i V_i = W_i \cdot \mathbf{1} \cdot \rho_i = \mathbf{1} \cdot \rho_i$.

Since $\hat{\Pi}$ is idempotent, i.e. $\hat{\Pi}^2 = \hat{\Pi}$, its rank is equal to its trace and so:

$$\operatorname{rank}(\hat{\Pi}) = \operatorname{trace}(\hat{\Pi}) = \sum_{i=1}^{S} \operatorname{trace}(W_i \Pi_i V_i) = \sum_{i=1}^{S} \operatorname{trace}(\mathbf{1} \cdot \rho_i) = S \cdot 1 = S.$$

We now show that $\operatorname{rank}(\hat{Q}_f) = N - S$.

It is well-known (cf. [39]) that, in a numbering that makes the ergodic partitioning explicit (and our numbering is just a more refined one), Q_f has the following form:

$$Q_f = \begin{pmatrix} Q_1 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & Q_2 & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & Q_S & \mathbf{0} & \mathbf{0} \\ \overline{Q}_1 & \overline{Q}_2 & \dots & \overline{Q}_S & \overline{Q} & \overline{Q}' \\ \widetilde{Q}_1 & \widetilde{Q}_2 & \dots & \widetilde{Q}_S & \widetilde{Q} & \widetilde{Q}' \end{pmatrix} \qquad Q_i = \operatorname{diag}(Q_{i1}, \dots, Q_{ie_i}),$$

where Q_{ij} are irreducible generators and $\begin{pmatrix} \overline{Q} & \overline{Q}' \\ \widetilde{Q} & \widetilde{Q}' \end{pmatrix}$ is an indecomposable semi-generator. Note that it follows that $\widetilde{Q}'$ must also be an indecomposable semi-generator.

We compute $\hat{Q}_f$:

$$\hat{Q}_f = W Q_f V = \begin{pmatrix} W_1 Q_1 V_1 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & W_2 Q_2 V_2 & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & W_S Q_S V_S & \mathbf{0} \\ \widetilde{W} \begin{pmatrix} \widetilde{Q}_1 V_1 \\ + \\ \widetilde{Q} \overline{V}_1 \end{pmatrix} & \widetilde{W} \begin{pmatrix} \widetilde{Q}_2 V_2 \\ + \\ \widetilde{Q} \overline{V}_2 \end{pmatrix} & \dots & \widetilde{W} \begin{pmatrix} \widetilde{Q}_S V_S \\ + \\ \widetilde{Q} \overline{V}_S \end{pmatrix} & \widetilde{W} \widetilde{Q}' \widetilde{V} \end{pmatrix}$$

and

$$W_i Q_i V_i = \sum_{j=1}^{e_i} W_{ij} Q_{ij} V_{ij}.$$

Since Q_{ij} is an irreducible generator, and since W_{ij} and V_{ij} satisfy the conditions of Lemma 13.2.11, we obtain that $W_{ij} Q_{ij} V_{ij}$ is also an irreducible generator. It is easy to prove that the sum of two irreducible generators is again an irreducible generator. We conclude that $W_i Q_i V_i$ is an irreducible generator.

Since $\widetilde{Q}'$ is an indecomposable semi-generator, and since $\widetilde{W}$ and $\widetilde{V}$ satisfy the conditions of Lemma 13.2.10, we obtain that $\widetilde{W}\widetilde{Q}'\widetilde{V}$ is an indecomposable semi-generator matrix.

It is known that the rank of an irreducible generator of dimension n is $n-1$. We have also proven in Lemma 13.2.10a that an indecomposable semi-generator matrix has full rank. Then $\text{rank}(\hat{Q}_f) = \sum_{i=1}^{S}(c_i - 1) + N - (L+1) + 1 = L - S + N - L = N - S$. ■

Recall that depending on the parameters in the τ-distributor there are, in general, many processes to which a τ-lumpable Markov reward chain with fast transitions τ-lumps to. The previous theorem showed that all these processes have equal limits. The next theorem shows that they are actually equal if all fast transitions were eliminated by τ-lumping, i.e., when the matrix that models fast transitions aggregates to zero matrix.

Theorem 13.2.13 Suppose $(\sigma, Q_s, Q_f, \rho) \overset{\mathcal{P}}{\rightsquigarrow}_\tau (\hat{\sigma}, \hat{Q}_s, \hat{Q}_f, \hat{\rho})$, suppose W is the τ-distributor used, and suppose $\hat{Q}_f = \mathbf{0}$. If W' is another τ-distributor (with a different choice of parameters), then $W'Q_sV = \hat{Q}_s$, $W'Q_fV = \mathbf{0}$ and $W'\rho = \hat{\rho}$. □

Proof Let $(\sigma, Q_s, Q_f, \rho) \to_\infty (\sigma, \Pi, Q, \Pi\rho)$. Since $\hat{Q}_f = WQ_fV = \mathbf{0}$, by Theorem 13.2.12 we have $W\Pi V = I$. Multiplying by V from the left and using that $VW\Pi V = \Pi V$, we obtain $\Pi V = V$. From Lemma 13.2.9, we have that $\Pi VW\Pi = \Pi VW$ and $\Pi VW'\Pi = \Pi VW'$. Since $\Pi V = V$, we have $VW\Pi = VW$ and $VW'\Pi = VW'$. Multiplying by W from the left, we get $W\Pi = W$ and $W'\Pi = W'$.

First, $W'Q_fV = W'\Pi Q_fV = \mathbf{0}$ because $\Pi Q_f = \mathbf{0}$ (as Π is the ergodic projection of Q_f). Next, using that UQV is the same for every distributor U, we have $\hat{Q}_s = WQ_sV = W\Pi Q_s\Pi V = WQV = W'QV = W'\Pi Q_s\Pi V = W'Q_sV$. Similarly, $W\rho = W\Pi\rho = W'\Pi\rho = W'\rho$. ■

13.3 $\tau_\sim$-lumping

In this section we introduce a notion of lumping for Markov reward chains with silent transitions, called $\tau_\sim$-lumping, by lifting τ-lumping to equivalence classes induced by the relation $\sim$ (recall Definition 12.3.1). Intuitively, we want a partitioning $\mathcal{P}$ of a Markov reward chain with silent transitions $(\sigma, Q_s, \mathcal{Q}_f, \rho)$ to be a $\tau_\sim$-lumping iff it is a τ-lumping for every Markov reward chain with fast transitions (σ, Q_s, Q_f, ρ) with $Q_f \in \mathcal{Q}_f$. Moreover, to have a proper lifting, we also want that $\hat{Q}_s = WQ_sV$ and $\hat{\rho} = W\rho$ do not depend on the choice of representative from $\mathcal{Q}_f$. This is crucial for the definition of slow transitions and rewards in the $\tau_\sim$-lumped process. Finally, to be able to define $\hat{\mathcal{Q}}_f$, we need that $WQ_fV \sim W'Q_f'V$ for all $Q_f, Q_f' \in \mathcal{Q}_f$, and that the non-zero elements of WQ_fV range over all positive real numbers (with Q_f ranging through $\mathcal{Q}_f$).

Before we give a definition that satisfies the above requirements, we give an example that shows that not every τ-lumping can be taken as $\tau_\sim$-lumping.

Example 13.3.1 a. Consider the Markov reward chain with silent transitions depicted in Figure 13.5a. Example 13.2.4b shows that the partitioning $\mathcal{P} = \big\{\{1,2\},\{3\}\big\}$ is a τ-lumping for all possible speeds given to the silent transitions. However, the slow transition in the τ-lumped process always depends on those speeds (cf. Figure 13.1b).

b. Consider the Markov reward chain with silent transitions depicted in Figure 13.5b. As Example 13.2.6c shows, if we assign the speeds a, b, a and c to the four silent steps respectively, we cannot have a proper τ-lumping. □

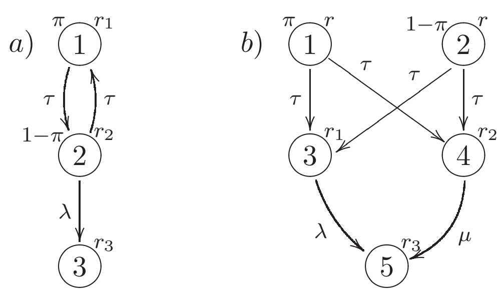


Figure 13.5: Not every τ-lumping can be $\tau_\sim$-lumping – Example 13.3.1

We define $\tau_\sim$-lumping by carefully restricting to the cases when τ-lumping is "speed independent", i.e. forbidding the situations from Example 13.3.1. For the definition we need to introduce some notation. We define $\mathrm{erg}(i) = \{j \in E \mid i \twoheadrightarrow j\}$ to be the set of all ergodic states reachable from state i and, for $X \subseteq \{1,\ldots,n\}$, we define $\mathrm{erg}(X) = \bigcup_{i\in X}\mathrm{erg}(i)$. Note that $j \in \mathrm{erg}(i)$ iff $\Pi[i,j] > 0$. Let E_L be some ergodic class. Then, for all $i \in E_L$, we have $\mathrm{erg}(i) = E_L$. Recall that $\delta_L[i] > 0$ iff $i \in T$ can be trapped in E_L. Therefore, $\delta_L[i] = 1$ iff $\mathrm{erg}(i) = E_L$.

The definition of $\tau_\sim$-lumping now follows.

Definition 13.3.2 ($\tau_\sim$-lumping) Let $(\sigma, Q_s, \mathcal{Q}_f, \rho)$ be a Markov reward chain with silent transitions. Let $\{E_1, \ldots, E_M, T\}$ be its ergodic partitioning and let $E = \bigcup_{1 \leqslant K \leqslant M} E_K$ be the set of ergodic states. A partitioning $\mathcal{P}$ is a $\tau_\sim$-*lumping* of $(\sigma, Q_s, \mathcal{Q}_f, \rho)$ iff:

1. for all $C \in \mathcal{P}$ at least one of the following holds:
   (a) $\text{erg}(C) \subseteq D$, for some $D \in \mathcal{P}$,
   (b) $\text{erg}(C) = E_L$, for some $1 \leq L \leq M$, or
   (c) $C \subseteq T$ and $i \to i'$, for *exactly* one $i \in C$, where $i' \notin C$;
2. for all $C \in \mathcal{P}$, for all $i, j \in C \cap E$, and for all $D \in \mathcal{P}$ such that $D \neq C$, $\sum_{\ell \in D} Q_s[i, \ell] = \sum_{\ell \in D} Q_s[j, \ell]$;
3. $\rho[i] = \rho[j]$ for all $i, j \in C \cap E$. □

Let us explain what these conditions mean. Condition 1 ensures that the lumping condition holds for the ergodic projection Π of every matrix from $\mathcal{Q}_f$. Condition 1a says that the ergodic states reachable by silent transitions from the states in C are all in the same lumping class. Condition 1b says that the ergodic states reachable by silent transitions from the states in C constitute an ergodic class. Condition 1c says that C is a set of transient states with precisely one (silent) exit. Note that Conditions 1a and 1b overlap when $E_i \subseteq D$. If, in addition, C contains only transient states and has only one exit, all the three conditions overlap. Condition 1 forbids lumping classes to contain parts of different ergodic classes in order to eliminate the effect of the ergodic probabilities. It also forbids the case where transient states of some lumping class lead to multiple ergodic classes that are not all subsets of the same lumping class (except in the case where there are only transient states in the lumping class and the class has only one exit). This is to eliminate the effect of the trapping probabilities (in the above exceptional case the trapping probabilities of the elements from the lumping class are all equal). Note that Condition 1 was violated in Example 13.3.1b. This is because states 3 and 4 were not in a lumping class nor in an ergodic class, and because the lumping class $\{1, 2\}$ had two exits.

Condition 2 says that every ergodic state in C must have the same accumulative rate to every other $\tau_\sim$-lumping class. This condition is needed to avoid the situation in Example 13.3.1a where a slow transition in the τ-lumped process depends on speeds. Condition 3 says that every ergodic state that belongs to the same lumping class must have the same reward.

The idea is the same as in Condition 2 but applied to the reward vector. The condition ensures that the rewards in the lumped process do not depend on speeds. Note that no condition is imposed on Q_s and ρ that concerns transient states.

We now show that the notion of $\tau_\sim$-lumping from Definition 13.3.2 exactly meets our requirements set in the beginning.

Theorem 13.3.3 Let $(\sigma, Q_s, \mathcal{Q}_f, \rho)$ be a Markov reward chain with silent transitions and let $\mathcal{P}$ be a partitioning. Then $\mathcal{P}$ is a $\tau_\sim$-lumping iff it is a τ-lumping for every Markov reward chain with fast transitions (σ, Q_s, Q_f, ρ) with $Q_f \in \mathcal{Q}_f$, and, moreover, for every $Q_f, Q'_f \in \mathcal{Q}_f$, $W'Q_sV = WQ_sV$ and $W'\rho = W\rho$, where W and W' are τ-distributors for Q_f and Q'_f respectively, and have the same values for the parameters. □

Proof ($\Rightarrow$) We prove that if the conditions of Definition 13.3.2 hold, then $\mathcal{P}$ is a τ-lumping for all representative matrices $Q_f \in \mathcal{Q}_f$.

First we show that the lumping condition on Π holds (where Π is the ergodic projection of Q_f). Recall that $VU\Pi V = \Pi V$ iff the rows of ΠV that correspond to states in the same partitioning class are equal. So it suffices to prove that, for all $C, D \in \mathcal{P}$, $\sum_{d \in D} \Pi[i,d] = \sum_{d \in D} \Pi[j,d]$ for all $i, j \in C$.

Suppose first that Condition 1a holds, i.e. that $\mathrm{erg}(C) \subseteq C'$ for some $C' \in \mathcal{P}$. Then, for all $i \in C$, $\mathrm{erg}(i) \subseteq C'$. From this it easily follows (by contradiction) that $\Pi[i,d] = 0$ for all $d \notin C'$. Let $D \in \mathcal{P}$ be some lumping class. If $D \neq C'$, then $\sum_{d \in D} \Pi[i,d] = 0$. Since Π is a stochastic matrix, its rows sum up to one, and so we also have $\sum_{d \in C'} \Pi[i,d] = 1$. We conclude that $\sum_{d \in D} \Pi[i,d]$ does not depend on $i \in C$.

Suppose second that Condition 1b holds, i.e. that $\mathrm{erg}(C) = E_L$ for some $1 \leqslant L \leqslant M$. Then, for all $i \in C$, $\mathrm{erg}(i) \subseteq E_L$. From this it follows that $\Pi[i,d] = 0$ for all $d \notin E_L$. Suppose first that $i \in E$. Then $\sum_{d \in D} \Pi[i,d] = \sum_{d \in D \cap E_L} \Pi[i,d] = \sum_{d \in D \cap E_L} \Pi[d,d]$. Suppose next that $i \in T$. Then from $\mathrm{erg}(i) \subseteq E_L$ it follows that $\delta_L[i] = 1$. Now, $\sum_{d \in D} \Pi[i,d] = \sum_{d \in D \cap E_L} \Pi[i,d] = \sum_{d \in D \cap E_L} \delta_L[i]\Pi[d,d] = \sum_{d \in D \cap E_L} \Pi[d,d]$. We conclude that $\sum_{d \in D} \Pi[i,d]$ does not depend on $i \in C$.

Assume finally that Condition 1c holds. Let $k \in C$ be the unique state in $C \subseteq T$ such that $k \to k'$ for some $k' \notin C$. Since $C \subseteq T$, we have $i \twoheadrightarrow k$. Note that this implies that $\delta_L[i] = \delta_L[k]$, for all $1 \leqslant L \leqslant M$. Let $D \in \mathcal{P}$. We have $\sum_{d \in D} \Pi[i,d] = \sum_{d \in D \cap E} \Pi[i,d] = \sum_{L : D \cap E_L \neq \emptyset} \sum_{d \in D \cap E_L} \Pi[i,d] = \sum_{L : D \cap E_L \neq \emptyset} \sum_{d \in D \cap E_L} \delta_L[i]\Pi[d,d] = \sum_{L : D \cap E_L \neq \emptyset} \sum_{d \in D \cap E_L} \delta_L[k]\Pi[d,d]$, and so $\sum_{d \in D} \Pi[i,d]$ does not depend on $i \in C$.

To show that $VU\Pi Q_s \Pi V = \Pi Q_s \Pi V$ and $VU\Pi\rho = \Pi\rho$ we use matrix manipulation. Let the numbering be such that it makes the division between ergodic and transient states explicit. Moreover, let the lumping classes be arranged so that the classes that contain ergodic states precede those that contain only transient states. This numbering gives the following forms for Π, Q_s, ρ and V:

$$\Pi = \begin{pmatrix} \Pi_E & \mathbf{0} \\ \Pi_T & \mathbf{0} \end{pmatrix}, \quad Q_s = \begin{pmatrix} Q_E & Q_{ET} \\ Q_{TE} & Q_T \end{pmatrix}, \quad \rho = \begin{pmatrix} \rho_E \\ \rho_T \end{pmatrix}, \quad V = \begin{pmatrix} V_E & \mathbf{0} \\ V_{TE} & V_T \end{pmatrix}.$$

Note that

$$\Pi Q_s = \begin{pmatrix} \Pi_E Q_E & \Pi_E Q_{ET} \\ \Pi_T Q_E & \Pi_T Q_{ET} \end{pmatrix} = \Pi \left(\begin{smallmatrix} Q_E & Q_{ET} \\ \mathbf{0} & \mathbf{0} \end{smallmatrix} \right), \quad \Pi\rho = \begin{pmatrix} \Pi_E \rho_E \\ \Pi_T \rho_E \end{pmatrix} = \Pi \left(\begin{smallmatrix} \rho_E \\ \mathbf{0} \end{smallmatrix} \right)$$

and

$$\Pi V = \begin{pmatrix} \Pi_E V_E & \mathbf{0} \\ \Pi_T V_E & \mathbf{0} \end{pmatrix} = \Pi \left(\begin{smallmatrix} V_E & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{smallmatrix} \right).$$

Condition 2 of Definition 13.3.2 imposes the lumping condition on the ergodic states. It can be rewritten in matrix form as:

$$V_E U_E \left(\begin{smallmatrix} Q_E & Q_{ET} \end{smallmatrix} \right) V = \left(\begin{smallmatrix} Q_E & Q_{ET} \end{smallmatrix} \right) V,$$

where U_E is a distributor matrix corresponding to (the collector matrix) V_E. Using that $VU\Pi V = \Pi V$ we compute:

$$\begin{aligned} VUQV &= VU\Pi Q_s \Pi V = VU\Pi \left(\begin{smallmatrix} Q_E & Q_{ET} \\ \mathbf{0} & \mathbf{0} \end{smallmatrix} \right) \Pi V = \\ &= VU\Pi \left(\begin{smallmatrix} Q_E & Q_{ET} \\ \mathbf{0} & \mathbf{0} \end{smallmatrix} \right) VU\Pi V = VU\Pi \left(\begin{smallmatrix} V_E U_E Q_E & V_E U_E Q_{ET} \\ \mathbf{0} & \mathbf{0} \end{smallmatrix} \right) VU\Pi V = \\ &= VU\Pi \left(\begin{smallmatrix} V_E & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{smallmatrix} \right) \left(\begin{smallmatrix} U_E Q_E & U_E Q_{ET} \\ \mathbf{0} & \mathbf{0} \end{smallmatrix} \right) VU\Pi V = \\ &= VU\Pi V \left(\begin{smallmatrix} U_E Q_E & U_E Q_{ET} \\ \mathbf{0} & \mathbf{0} \end{smallmatrix} \right) VU\Pi V = \Pi V \left(\begin{smallmatrix} U_E Q_E & U_E Q_{ET} \\ \mathbf{0} & \mathbf{0} \end{smallmatrix} \right) VU\Pi V = \\ &= \Pi Q_s \Pi V = QV. \end{aligned}$$

Condition 3 of Definition 13.3.2 is written in matrix form as:

$$V_E U_E \rho_E = \rho_E.$$

Similarly as we did for Q, we compute

$$\begin{aligned} VU\Pi\rho &= VU\Pi \left(\begin{smallmatrix} \rho_E \\ \mathbf{0} \end{smallmatrix} \right) = VU\Pi \left(\begin{smallmatrix} V_E U_E \rho_E \\ \mathbf{0} \end{smallmatrix} \right) = VU\Pi \left(\begin{smallmatrix} V_E & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{smallmatrix} \right) \left(\begin{smallmatrix} U_E \rho_E \\ \mathbf{0} \end{smallmatrix} \right) = \\ &= VU\Pi V \left(\begin{smallmatrix} U_E \rho_E \\ \mathbf{0} \end{smallmatrix} \right) = \Pi V \left(\begin{smallmatrix} U_E \rho_E \\ \mathbf{0} \end{smallmatrix} \right) = \Pi \left(\begin{smallmatrix} V_E U_E \rho_E \\ \mathbf{0} \end{smallmatrix} \right) = \Pi \left(\begin{smallmatrix} \rho_E \\ \mathbf{0} \end{smallmatrix} \right) = \Pi\rho. \end{aligned}$$

We show that $\hat{Q}_s$ does not depend on the representative Q_f. Let $\hat{Q}_s = WQ_sV$ for some τ-distributor W. Suppose we take $Q'_f \sim Q_f$ instead of Q_f

and let W' be the τ-distributor for Q'_f that has the same parameters as W (note that the number of parameters depends only on the grammar of Q_f). We show that $\hat{Q}_s = W'Q_sV$.

The matrices W and W' have the following form:

$$W = \begin{pmatrix} W_E & \mathbf{0} \\ \mathbf{0} & W_T \end{pmatrix} \quad \text{and} \quad W' = \begin{pmatrix} W'_E & \mathbf{0} \\ \mathbf{0} & W_T \end{pmatrix}.$$

Note that W and W' have the same block that corresponds to the classes that contain only transient states. This is because this block only depends on the parameters and not on Q_f. Now,

$$W \left(\begin{smallmatrix} \mathbf{0} & \mathbf{0} \\ Q_{TE} & Q_T \end{smallmatrix} \right) = W' \left(\begin{smallmatrix} \mathbf{0} & \mathbf{0} \\ Q_{TE} & Q_T \end{smallmatrix} \right).$$

Since W_E and W'_E are distributors for V_E, we also have

$$W_E \left(\begin{smallmatrix} Q_E & Q_{ET} \end{smallmatrix} \right) V = W'_E \left(\begin{smallmatrix} Q_E & Q_{ET} \end{smallmatrix} \right) V,$$

which implies

$$W \left(\begin{smallmatrix} Q_E & Q_{ET} \\ \mathbf{0} & \mathbf{0} \end{smallmatrix} \right) V = W' \left(\begin{smallmatrix} Q_E & Q_{ET} \\ \mathbf{0} & \mathbf{0} \end{smallmatrix} \right) V.$$

We now compute:

$$\begin{aligned} \hat{Q}_s = WQ_sV = W \begin{pmatrix} Q_E & Q_{ET} \\ Q_{TE} & Q_T \end{pmatrix} V = W \left(\begin{smallmatrix} Q_E & Q_{ET} \\ \mathbf{0} & \mathbf{0} \end{smallmatrix} \right) V + W \left(\begin{smallmatrix} \mathbf{0} & \mathbf{0} \\ Q_{TE} & Q_T \end{smallmatrix} \right) V = \\ = W' \left(\begin{smallmatrix} Q_E & Q_{ET} \\ \mathbf{0} & \mathbf{0} \end{smallmatrix} \right) V + W' \left(\begin{smallmatrix} \mathbf{0} & \mathbf{0} \\ Q_{TE} & Q_T \end{smallmatrix} \right) V = W'Q_sV = \hat{Q}'_s. \end{aligned}$$

To show that the reward vector of the lumped process does not depend on the representative Q_f note that $V_E U_E \rho_E = \rho_E$. From this it follows that $W_E\rho_E = W'_E\rho_E$ which directly implies

$$W\rho = \begin{pmatrix} W_E\rho_E \\ W_T\rho_T \end{pmatrix} = \begin{pmatrix} W'_E\rho_E \\ W_T\rho_T \end{pmatrix} = W'\rho.$$

($\Leftarrow$) First we show that Condition 1 of Definition 13.3.2 must hold if the lumping condition on Π is to hold for every $Q_f \in \mathcal{Q}_f$. The proof is by contraposition. Suppose Conditions 1a, 1b and 1c do not hold. Let $C \in \mathcal{P}$. We show that there is always a $D \in \mathcal{P}$ such that $\sum_{d\in D} \Pi[i,d]$ is not the same for every $i \in C$. We distinguish two cases, when $C \cap E \neq \emptyset$ and when $C \subseteq T$.

Suppose $C \cap E \neq \emptyset$. Let the ergodic classes be arranged so that there is a $1 \leqslant P \leqslant M$ such that $E_K \cap C \neq \emptyset$ for $K \leqslant P$, and $E_K \cap C = \emptyset$ for $K \geqslant P+1$. Since Condition 1b does not hold, we have $P \geqslant 2$. We show that not $E_K \subseteq C$ for all $1 \leqslant K \leqslant P$.

Suppose not, i.e., that $E_K \subseteq C$ for all $1 \leqslant K \leqslant P$. We prove that then $\mathrm{erg}(C) \subseteq C$, which gives us contradiction because Condition 1a does not hold. If $C \cap T = \emptyset$, then $C \subseteq E$ and so $\mathrm{erg}(C) \subseteq C$ follows directly. Suppose now that $C \cap T \neq \emptyset$ and let $i \in C \cap T$. We show that $\mathrm{erg}(i) \subseteq C$. Suppose not. Then there is an $k \in E$ such that $i \twoheadrightarrow k$ and $k \notin (E_1 \cup \cdots \cup E_P)$. Let $D \in \mathcal{P}$ be such that $k \in D$ and let $\ell \in E_L$ for some $1 \leqslant L \leqslant P$. Then $\sum_{d \in D} \Pi[i,d] > 0$ and $\sum_{d \in D} \Pi[\ell,d] = 0$, and so the lumping condition does not hold. We conclude that $\mathrm{erg}(i) \subseteq C$. From this it follows that $\mathrm{erg}(C) \subseteq C$. We conclude that not $E_K \subseteq C$ for all $1 \leqslant K \leqslant P$.

Let $1 \leqslant I, J \leqslant P$ be such that $E_I \cap C \neq \emptyset$, $E_J \cap C \neq \emptyset$ and $E_I \not\subseteq C$. Then there is a $D \in \mathcal{P}$ such that $E_I \cap D \neq \emptyset$. By Lemma 13.2.2 it follows that $E_J \cap D \neq \emptyset$. Let $i \in C \cap E_I$. Then $\sum_{d \in D} \Pi[i,d] = \sum_{d \in D} \Pi[i,d] \notin \{0,1\}$. Similarly, for some $j \in C \cap E_I$ we have $\sum_{d \in D} \Pi[j,d] = \sum_{d \in D} \Pi[j,d] \notin \{0,1\}$. Now, we can always choose a Q_f so that the ergodic probabilities of E_I and E_J are such that $\sum_{d \in D} \Pi[i,d] \neq \sum_{d \in D} \Pi[j,d]$.

Suppose now that $C \subseteq T$. Let $i_1, \ldots, i_p \in C$ be such that, for all $1 \leqslant k \leqslant p$, we have $i_k \to i'_k$ for some $i'_k \notin C$. Since Condition 1c does not hold, we have $p \geqslant 2$. Let $C_1, \ldots, C_P \in \mathcal{P}$ be all lumping classes such that $\mathrm{erg}(i_k) \cap C_K \neq \emptyset$ for some $1 \leqslant k \leqslant p$ and all $1 \leqslant K \leqslant P$. Note first that, because of the lumping condition, $\mathrm{erg}(i_k) \cap C_K \neq \emptyset$ for *all* $1 \leqslant k \leqslant p$. Note second that $P \geqslant 2$, because otherwise we would have $\mathrm{erg}(C) \subseteq C_1$ which does not hold because Condition 1a does not hold. Let $D \in \{C_1, \ldots, C_P\}$. We cannot find Π and $i, j \in \{i_1, \ldots, i_p\}$ such that $\sum_{d \in D} \Pi[i,d] \neq \sum_{d \in D} \Pi[j,d]$ only if there exists an ergodic class E_L such that $\mathrm{erg}(i_K) \subseteq E_L$ for all $1 \leqslant k \leqslant p$. This, however, is not possible because it would imply that $\mathrm{erg}(C) \subseteq E_L$ which does not hold because Condition 1b does not hold.

We conclude that Condition 1 holds. Using this, we now only show that Condition 3 holds. For Condition 2 the proof is essentially the same and is omitted.

Let $C_K \in \mathcal{P}$, let $i, j \in C \cap E$ and let $i \in E_I$ and $j \in E_J$ for some ergodic classes E_I and E_J. From what we proved before it follows that $E_I \subseteq C$ and $E_J \subseteq C$. We distinguish two cases, when $I = J$ and when $I \neq J$.

Suppose $I = J$. Let W be a τ-distributors associated to Q_f such that the parameters α_{JL} in Definition 13.2.3 are equal to $\frac{1}{e_J}$. Then

$$(W\rho)[K] = \sum_{k \in C_K} W[K,k]\rho[k] = \sum_{k \in C_K \cap E} \frac{\Pi[k,k]}{\sum_{\ell \in C_K} \Pi[\ell,\ell]} \rho_k.$$

Define Π' to be the same as Π but with $\Pi'[\ell,i] = \Pi[\ell,i] + \varepsilon$ for all $\ell \in E_I$, and $\Pi'[\ell,j] = \Pi[\ell,j] - \varepsilon$ for all $\ell \in E_J$, where $0 < \varepsilon < \Pi[j,j]$. Clearly, Π' is of the right form and it satisfies the lumping condition because $E_I = E_J \subseteq C$. We can always find $Q'_f \sim Q_f$ such that Π' is its ergodic projection. Let W' be a τ-distributors associated to Q'_f again such that the parameters α_{JL} are all the same. After some simple calculation, we obtain that $(W'\rho)[K] - (W\rho)[K] = \varepsilon(\rho[i] - \rho[j])$. Therefore, if $\rho[i] \neq \rho[j]$, then $(W\rho)[K] \neq (W'\rho)[K]$. We conclude that $\rho[i] = \rho[j]$.

Suppose now that $I \neq J$. If $|E_I| = |E_J| = 1$, then

$$(\Pi\rho)[i] = \sum_k \Pi[i,k]\rho[k] = \sum_{k\in E_I} \Pi[i,k]\rho[k] = \rho[i]$$

and similarly $(\Pi\rho)[j] = \rho[j]$. Therefore, $\rho[i] = \rho[j]$. Suppose $|E_I| > 1$. We define a matrix Π' to be the same as Π except that $\Pi'[k,i] = \Pi[k,i] + \varepsilon$ for all $k \in E_I$, and $\Pi'[\ell,j] = \Pi[\ell,j] - \varepsilon$ for all $\ell \in E_J$, with $0 < \varepsilon < \Pi[j,j]$. As before it easily follows that the lumping condition still holds for Π' and that Π' is of the right form. Now, since $(\Pi\rho)[i] = (\Pi\rho)[j]$, $(\Pi'\rho)[i] = (\Pi'\rho)[j]$ and $(\Pi'\rho)[j] = (\Pi\rho)[j]$, we have $(\Pi'\rho)[i] = (\Pi\rho)[i]$. From this it easily follows that $\rho[\ell] = \rho[i]$ for all $\ell \in E_I$. Then, if $|E_J| = 1$, we have $\rho[i] = \rho[\ell]$. If not, with the same reasoning as for E_I, we can obtain that $\rho[\ell] = \rho[j]$, for all $\ell \in E_J$. Now,

$$\begin{aligned}\rho[i] = \rho[i]\sum_{k\in E_I}\Pi[i,k] &= \sum_{k\in C_K}\Pi[i,k]\rho[k] = \\ = \sum_{k\in C_K}\Pi[j,k]\rho[k] &= \sum_{k\in E_J}\Pi[j,k]\rho[k] = \rho[j]\sum_{k\in E_I}\Pi[j,k] = \rho[j]. \quad \blacksquare\end{aligned}$$

As we said in the beginning, for the definition of $\tau_\sim$-lumping to be considered correct we must also establish that $WQ_fV \sim W'Q'_fV$, and that the non-zero elements of WQ_fV range over all positive real numbers. The proof of this is easy (it follows from $W \sim W'$ and the fact that non-zero elements in Π can take any value less than 1), however cumbersome, and is therefore omitted.

Now, if $\mathcal{P}$ is a $\tau_\sim$-lumping and if $(\sigma, Q_s, Q_f, \rho) \overset{\mathcal{P}}{\rightsquigarrow}_\tau (\hat{\sigma}, \hat{Q}_s, \hat{Q}_f, \hat{\rho})$ for some $Q_f \in \mathcal{Q}_f$, then we say that $(\sigma, Q_s, \mathcal{Q}_f, \rho)$ *$\tau_\sim$-lumps* (with respect to $\mathcal{P}$) to $(\hat{\sigma}, \hat{Q}_s, \hat{\mathcal{Q}}_f, \hat{\rho})$ where $\hat{\mathcal{Q}}_f = [\hat{Q}_f]_\sim$ and write $(\sigma, Q_s, \mathcal{Q}_f, \rho) \overset{\mathcal{P}}{\rightsquigarrow}_{\tau_\sim} (\hat{\sigma}, \hat{Q}_s, \hat{\mathcal{Q}}_f, \hat{\rho})$. Note that, as for τ-lumping, there can be several Markov reward chains with silent transitions to which $(\sigma, Q_s, Q_f \sim, \rho)$ $\tau_\sim$-lumps to (unless there are no fast transitions in the lumped process).

We give some examples of $\tau_\sim$-lumpings.

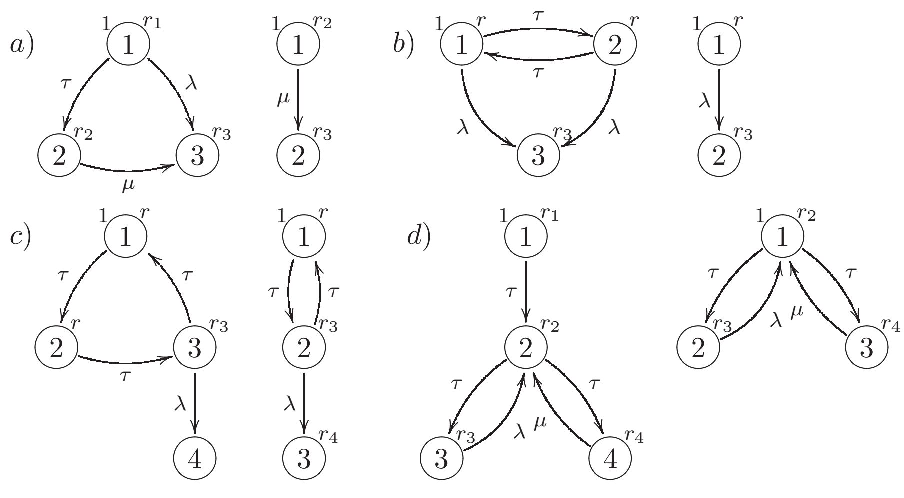


Figure 13.6: $\tau_\sim$-lumpings – Example 13.3.4

Example 13.3.4 Consider the Markov reward chains with silent transitions depicted in Figure 13.6 on the left sides. For each of them we give a $\tau_\sim$-lumping and for each lumping class we show which option of Condition 1 of Definition 13.3.2 holds. The corresponding lumped Markov reward chains with silent transitions are depicted in Figure 13.6 on the right sides.

a. For the Markov reward chain with silent transitions depicted in Figure 13.6a the partitioning $\mathcal{P} = \big\{\{1,2\},\{3\}\big\}$ is a $\tau_\sim$-lumping. For the lumping class $\{1,2\}$ Condition 1a in Definition 13.3.2 is satisfied. For the class $\{3\}$ both Conditions 1a and 1b are satisfied.

b. For the Markov reward chain with silent transitions in Figure 13.6b $\mathcal{P} = \big\{\{1,2\},\{3\}\big\}$ is a $\tau_\sim$-lumping. For both lumping classes Conditions 1a and 1b are satisfied.

c. For the Markov reward chain with silent transitions in Figure 13.6c $\mathcal{P} = \big\{\{1,2\},\{3\},\{4\}\big\}$ is a $\tau_\sim$-lumping. For the lumping classes $\{1,2\}$ and $\{4\}$ both Conditions 1a and 1b are satisfied. For the class $\{3\}$ only Condition 1b is satisfied. Note that the partitioning $\mathcal{P} = \big\{\{1,2,3\},\{4\}\big\}$ is not a $\tau_\sim$-lumping even when $r_r = r$ because it violates Condition 2.

d. For the Markov reward chain with silent transitions in Figure 13.6d $\mathcal{P} = \big\{\{1,2\},\{3\},\{4\}\big\}$ is a $\tau_\sim$-lumping. For the classes $\{3\}$ and $\{4\}$

both Conditions 1a and 1b are satisfied. Since $\{1, 2\}$ contains only transient states, for this class only Condition 1c is satisfied. □

Chapter 14

Aggregation by Reduction

In this chapter we first consider the specific aggregation (and disaggregation) method of [39, 32] and extend it with initial probabilities and rewards. This method reduces a discontinuous Markov chain to a Markov chain, eliminating instantaneous states while keeping the same distributions on the set of regular states. Then, we adapt this method for the setting of Markov reward chains with fast transitions. We call this method *τ-reduction* as it eliminates all fast transitions and reduces a Markov reward chain with fast transitions to a Markov reward chain. We develop two corresponding methods in the setting of Markov reward chains with silent transitions; the first is called *$\tau_\sim$-reduction* and the second is *total $\tau_\sim$-reduction*.

14.1 Reduction to a Markov reward chain

The reduction of a discontinuous Markov reward chain to a Markov reward chain of [39, 32] requires the notion of canonical product decomposition. Recall that

$$\Pi = \begin{pmatrix} \Pi_1 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Pi_2 & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \Pi_M & \mathbf{0} \\ \overline{\Pi}_1 & \overline{\Pi}_2 & \dots & \overline{\Pi}_M & \mathbf{0} \end{pmatrix}$$

where $\Pi_K = \mathbf{1} \cdot \mu_K$ and $\overline{\Pi}_K = \delta_K \cdot \mu_K$ for a *row* vector $\mu_K > 0$ such that $\mu_K \cdot \mathbf{1} = 1$ and a vector $\delta_K \geq 0$ such that $\sum_{i=1}^{M} \delta_K = \mathbf{1}$. The canonical product decomposition decomposes Π into the product of two matrices; one containing the μ_K's only, the other the δ_K's only.

Definition 14.1.1 (Canonical product decomposition) Let (σ, Π, Q) be a discontinuous Markov chain with a numbering that makes the ergodic partitioning explicit. The *canonical product decomposition* of Π is given by the matrices $L \in \mathbb{R}^{M \times n}$ and $R \in \mathbb{R}^{n \times M}$, defined as follows:

$$L = \begin{pmatrix} \mu_1 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mu_2 & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mu_M & \mathbf{0} \end{pmatrix} \qquad R = \begin{pmatrix} \mathbf{1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{1} \\ \delta_1 & \delta_2 & \dots & \delta_M \end{pmatrix}.$$

The dimension of the K-th vector $\mathbf{1}$ in R is the same as the dimension of its corresponding row vector μ_K; the dimension of the K-th row vector $\mathbf{0}$ in L is the same as the dimension of its corresponding δ_K. Note that then $RL = \Pi$ and $LR = I$. □

In case the numbering does not make the ergodic partitioning explicit, we need to renumber the states first, then construct L and R, and then renumber back to the original numbering. An example follows.

Example 14.1.2 a. Let

$$\Pi = \begin{pmatrix} 1-p & p & 0 \\ 1-p & p & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

The numbering is as needed and we obtain

$$L = \begin{pmatrix} 1-p & p & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad R = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

b. Let now

$$\Pi = \begin{pmatrix} 0 & p & 1-p & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

This numbering does not make the ergodic partitioning explicit. We renumber states to obtain

$$\Pi' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ p & 1-p & 0 & 0 \end{pmatrix}.$$

From this,

$$L' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad \text{and} \quad R' = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ p & 1-p & 0 \end{pmatrix}.$$

After renumbering back we have

$$L = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad R = \begin{pmatrix} p & 1-p & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

□

The method of [39, 32] masks the stochastic discontinuity in a discontinuous Markov chain and transforms it into a standard Markov chain that has the same behavior in regular states. We extend this method with an initial probability vector and with a reward vector. If (σ, Π, Q, ρ) is a discontinuous Markov reward chain, then the reduced Markov reward chain $(\hat{\sigma}, I, \hat{Q}, \hat{\rho})$ is defined by

$$\hat{\sigma} = \sigma R, \ \hat{Q} = LQR, \ \text{ and } \ \hat{\rho} = L\rho.$$

The states of the simplified process are exactly the ergodic classes of the original process. The transient states are eliminated. Intuitively, they are split probabilistically between the ergodic classes according to their trapping probabilities. In case a transient state is also an initial state, the initial state probabilities are split according to their trapping probabilities. Similarly, the joint reward is the sum of the individual rewards from the ergodic class weighted by their ergodic probabilities.

Under certain conditions we can obtain the original process from the reduced one. The transition matrix of the aggregated process has been shown in [32] to satisfy $\hat{P}(t) = e^{LQRt} = LP(t)R$, for $t > 0$. Since $\Pi P(t) = P(t)\Pi = P(t)$, if Π of the original process is known, and if $\sigma\Pi = \sigma$ and $\Pi\rho = \rho$, then there is a disaggregation procedure $\sigma = \hat{\sigma}L$, $P(t) = R\hat{P}(t)L$ and $\rho = R\hat{\rho}$.

Like lumping, the reduction procedure also preserves the total reward:

$$\hat{R}(t) = \hat{\sigma}\hat{P}(t)\hat{\rho} = \sigma RLP(t)RL\rho = \sigma\Pi P(t)\Pi\rho = \sigma P(t)\rho = R(t).$$

In case the original process has no stochastic discontinuity, i.e. $\Pi = I$, the aggregated process is equal to the original since then $L = R = I$.

We give an example.

Example 14.1.3 a. Consider the discontinuous Markov chain (σ, Π, Q, ρ) defined by $\sigma = \begin{pmatrix}\pi & 1{-}\pi & 0\end{pmatrix}$ and

$$\Pi = \begin{pmatrix} 1{-}p & p & 0 \\ 1{-}p & p & 0 \\ 0 & 0 & 1 \end{pmatrix}, \ Q = \begin{pmatrix} -p(1{-}p)\lambda & -p^2\lambda & p\lambda \\ -p(1{-}p)\lambda & -p^2\lambda & p\lambda \\ 0 & 0 & 0 \end{pmatrix}, \text{ and } \rho = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix}.$$

The matrix Π is the one from Example 14.1.2a which gives us L and R. Now,

$$\hat{\sigma} = \sigma R = \begin{pmatrix} 1 & 0 \end{pmatrix}, \ \hat{\rho} = L\rho = \begin{pmatrix} (1{-}p)r_1 + pr_2 \\ r_3 \end{pmatrix}$$

and

$$\hat{Q} = LQR = \begin{pmatrix} -p\lambda & p\lambda \\ 0 & 0 \end{pmatrix}.$$

The reduced Markov reward chain $(\hat{\sigma}, I, \hat{Q}, \hat{\rho})$ is depicted in Figure 14.1a.

b. Let (σ, Π, Q, ρ) be defined by $\sigma = \begin{pmatrix}1 & 0 & 0\end{pmatrix}$,

$$\Pi = \begin{pmatrix} 0 & p & 1{-}p & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \ Q = \begin{pmatrix} 0 & -p\lambda & -(1{-}p)\mu & p\lambda + (1{-}p)\mu \\ 0 & -\lambda & 0 & \lambda \\ 0 & 0 & -\lambda & \lambda \\ \nu & 0 & 0 & -\nu \end{pmatrix},$$

and $\rho = \begin{pmatrix} r_1 & r_2 & r_3 & r_4 \end{pmatrix}^{\mathsf{T}}$. The matrix Π of this process is the one from Example 14.1.2b which gives us L and R. We have

$$\hat{\sigma} = \begin{pmatrix} p & 1{-}p & 0 \end{pmatrix}, \ \hat{Q} = \begin{pmatrix} -\lambda & 0 & \lambda \\ 0 & -\mu & \mu \\ p\nu & (1{-}p)\nu & -\nu \end{pmatrix} \text{ and } \hat{\rho} = \begin{pmatrix} r_2 \\ r_3 \\ r_4 \end{pmatrix}.$$

The Markov reward chain $(\hat{\sigma}, I, \hat{Q}, \hat{\rho})$ is depicted in Figure 14.1b. □

14.2 τ-reduction

Since we are interested only in the case when fast transitions are instantaneous, in the part on lumping we were justifying all operations only in the limit. We do the same here for the reduction method. We adapt the aggregation method from the previous section to reduce a Markov reward chain

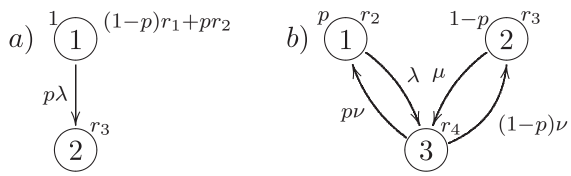
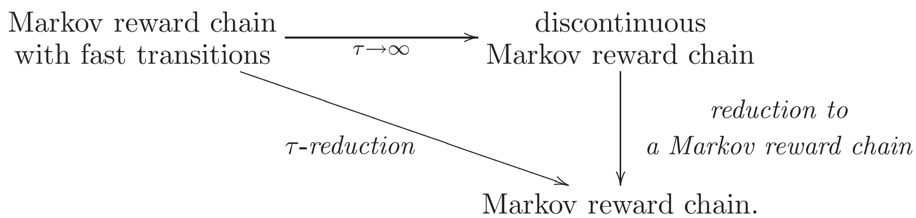


Figure 14.1: Markov reward chains obtained by reduction - Example 14.1.3

with fast transitions to an asymptotically equivalent Markov chain. The τ-reduced Markov reward chain with fast transitions is naturally defined to be the Markov chain obtained by reducing the limit discontinuous Markov reward chain. The definition is clarified by the following diagram:



We give a definition of τ-reduction.

Definition 14.2.1 (τ-reduction) Let (σ, Q_s, Q_f, ρ) be a Markov reward chain with fast transitions and let $(\sigma, Q_s, Q_f, \rho) \to_\infty (\sigma, \Pi, Q, \Pi\rho)$. Assume that $\Pi = RL$ is the canonical product decomposition of Π. Then the *τ-reduct* of (σ, Q_s, Q_f, ρ) is the Markov reward chain $(\hat{\sigma}, I, \hat{Q}, \hat{\rho})$ defined by

$$\hat{\sigma} = \sigma R, \ \hat{Q} = LQ_sR, \quad \text{and} \quad \hat{\rho} = L\rho. \qquad \square$$

Note that the definition corresponds to the above diagram because

$$LQR = L\Pi Q_s \Pi R = LQ_sR \quad \text{and} \quad L\Pi\rho = L\rho.$$

We give some examples.

Example 14.2.2 a. Let (σ, Q_s, Q_f, ρ) be the Markov reward chain with fast transitions from Figure 14.2a on the left. The limit of this Markov reward chain with fast transitions was calculated in Example 12.2.4a and we had

$$\Pi = \lim_{t\to\infty} e^{Q_f t} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

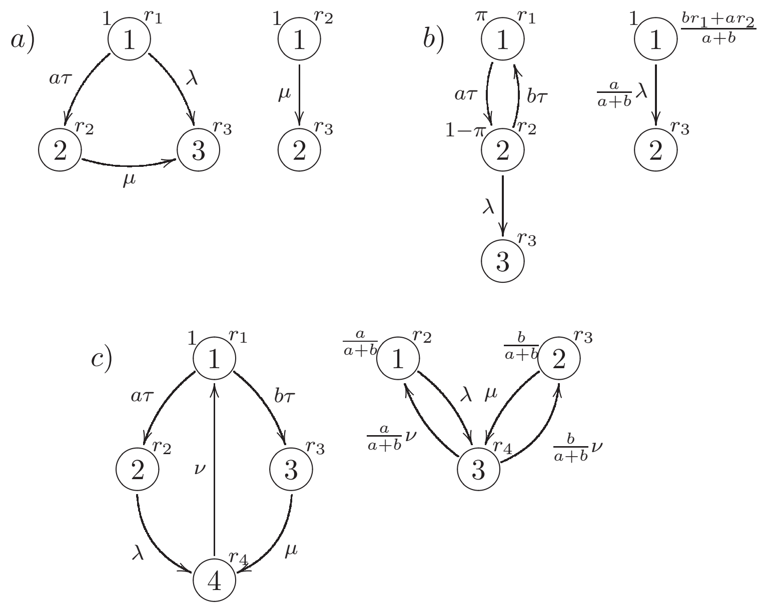


Figure 14.2: τ-reduction – Example 14.2.2

From this

$$L = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad R = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

We obtain

$$\hat{\sigma} = \sigma R = \begin{pmatrix} 1 & 0 \end{pmatrix}, \; \hat{Q}_s = LQ_sR = \begin{pmatrix} -\mu & \mu \\ 0 & 0 \end{pmatrix} \text{ and } \hat{\rho} = L\rho = \begin{pmatrix} r_2 \\ r_3 \end{pmatrix}.$$

The Markov reward chain $(\hat{\sigma}, I, \hat{Q}_s, \hat{\rho})$ is depicted in Figure 14.2a on the right.

b. Consider now the Markov reward chain with fast transitions from Figure 14.2b on the left. Note that the limit of this Markov reward chain with fast transitions is the discontinuous Markov reward chain from Example 14.1.3a when $p = \frac{a}{a+b}$. According to the definition of τ-reduction, both of these processes reduce to the same Markov reward chain. We depict the τ-reduced process in Figure 14.2b on the right.

c. As in the previous case, the limit of the Markov reward chain with fast transitions from Figure 14.2c on the left is the discontinuous Markov reward chain from Example 14.1.3b for $p = \frac{a}{a+b}$. This automatically gives us the τ-reduced process depicted in Figure 14.2c on the right.□

14.3 $\tau_\sim$-reduction and total $\tau_\sim$-reduction

In this section we extend the technique of τ-reduction to Markov reward chains with silent transitions. Two methods for reduction are given. The first, called $\tau_\sim$-reduction, is a direct lifting of τ-reduction to the set of Markov reward chains with fast transitions. The second method, called total $\tau_\sim$-reduction, combines τ-reduction with ordinary lumping for standard Markov reward chains to achieve better aggregation.

As we did for $\tau_\sim$-lumping, we want to define $\tau_\sim$-reduction by properly lifting the notion of τ-reduction. Intuitively, we want to say that $(\sigma, Q_s, \mathcal{Q}_f, \rho)$ is $\tau_\sim$-reducible iff σR, LQ_sR and $L\rho$ do not depend on the choice of the representative $Q_f \in \mathcal{Q}_f$, where RL is the canonical product decomposition of the ergodic projection of Q_f. As Example 14.2.2 shows, not every Markov reward chain with silent transitions is $\tau_\sim$-reducible (cf. Figure 14.2b and 14.2c).

We give a definition that characterizes $\tau_\sim$-reduction.

Definition 14.3.1 ($\tau_\sim$-reduction) Let $(\sigma, Q_s, \mathcal{Q}_f, \rho)$ be a Markov reward chain with silent transitions, let $\{E_1, E_2, \dots, E_M, T\}$ be its ergodic partitioning, and let $E = \bigcup_{1 \leqslant K \leqslant M} E_K$. Then $(\sigma, Q_s, \mathcal{Q}_f, \rho)$ is $\tau_\sim$-*reducible* iff the following conditions hold:

1. for all $i \in T$, either $\sigma[i] = 0$ or $\mathrm{erg}(i) = E_L$ for some $1 \leqslant L \leqslant M$;

2. (a) for all $j \in T$, either $Q_s[i,j] = 0$ for all $i \in E$, or $\mathrm{erg}(j) = E_L$ for some $1 \leqslant L \leqslant M$; and

   (b) for all $1 \leqslant K, L \leqslant M$ and all $i, j \in E_K$,

$$\sum_{\ell:\mathrm{erg}(\ell)=E_L} Q_s[i,\ell] = \sum_{\ell:\mathrm{erg}(\ell)=E_L} Q_s[j,\ell];$$

3. for all $1 \leqslant K \leqslant M$ and all $i, j \in E_K$, $\rho[i] = \rho[j]$. □

Condition 1 makes sure that an initial transient state can be trapped only in one ergodic class. Allowing it to be trapped in more classes would cause the initial vector of the reduced process to depend on the trapping probabilities, i.e., on speeds (cf. Example 14.2.2c). Condition 2a is the same but instead of an initial state we consider a state that has a slow transition leading to it. This is to forbid the situation where, due to the state splitting, the transition rates in the reduced process depend on speeds (see again Example 14.2.2c). Note that the reduction aggregates whole ergodic classes and performs weighted summing of all rates that lead out of the states in these classes. This sum is speed independent only if all these rates are equal (otherwise we have the situation as in Example 14.2.2b). This is ensured by Condition 2b. Finally, Condition 3 says that states from the same ergodic class must have equal rewards. This is needed because, as for the slow transitions, the new reward is a weighted sum of the rewards from the ergodic class (see Example 14.2.2b).

We prove two lemmas that will help us prove that Definition 14.3.1 meets all our requirements from the beginning, i.e., that $\tau_\sim$-reduction induces a speed independent τ-reduction for all representatives.

Lemma 14.3.2 Let $A \in \mathbb{R}^{n\times m}$ be such that $A \geqslant 0$. Then the following two statements are equivalent:

- μA is the same for any vector $\mu \in \mathbb{R}^{1\times n}$ such that $\mu > 0$ and $\mu \cdot \mathbf{1} = 1$;
- $A = \mathbf{1} \cdot a$ for some $a \in \mathbb{R}^{1\times m}$. □

Proof (⇒) Let μ be such that $\mu > 0$ and $\mu \cdot \mathbf{1} = \mathbf{1}$. Let $k, l \in \{1, \ldots, n\}$ be arbitrary and let ε be such that $0 < \varepsilon < \mu[l]$. Define $\mu' \in \mathbb{R}^{1\times n}$ as $\mu'[k] = \mu[k] + \varepsilon$, $\mu'[l] = \mu[l] - \varepsilon$, and $\mu'[i] = \mu[i]$ for all $i \neq k, l$. By definition, $\mu' > 0$ and $\mu' \cdot \mathbf{1} = 1$. From $\mu A = \mu' A$ we obtain that, for all $j \in \{1, \ldots, m\}$, $\varepsilon A[k, j] - \varepsilon A[l, j] = 0$. Since $\varepsilon > 0$, we have $A[k, j] = A[l, j]$ for all $j \in \{1, \ldots, m\}$. Because k and l were arbitrary, we conclude that all rows in A are equal, i.e. that $A = \mathbf{1} \cdot a$ for some $a \in \mathbb{R}^{1\times m}$.

(⇐) Suppose $A = \mathbf{1} \cdot a$ for some $a \in \mathbb{R}^{1\times m}$. Clearly, $\mu A = \mu \mathbf{1} a = a$ does not depend on μ. ■

Lemma 14.3.3 Let $A \in \mathbb{R}^{m\times n}$ be such that $A \geqslant 0$. Let $\delta \in \mathbb{R}^{n\times 1}$ be such that $\delta \geqslant 0$ and $\delta - \mathbf{1} \leqslant 0$. Then the following two statements are equivalent:

- $A\delta = A\delta'$ for all $\delta' \in \mathbb{R}^{n\times 1}$ such that $\delta' \sim \delta$ and $(\delta' - \mathbf{1}) \sim (\delta - \mathbf{1})$;
- for all $1 \leqslant j \leqslant n$, either $A[i, j] = 0$ for all $1 \leqslant i \leqslant m$, or $\delta[j] \in \{0, 1\}$.□

Proof ($\Rightarrow$) Let $j \in \{1,\ldots,n\}$ be such that $\delta[j] \notin \{0,1\}$ (if such j does not exists, the theorem vacuously holds). Define $\delta' \in \mathbb{R}^{n\times 1}$ by $\delta'[k] = \delta[k]$ for all $k \neq j$, and by $\delta'[j] = \delta[j] + \varepsilon$, for some ε such that $0 < \varepsilon < 1 - \delta[j]$. Clearly, $\delta' \sim \delta$ and $(\delta' - \mathbf{1}) \sim (\delta - \mathbf{1})$ because δ and δ' are different only in one element that is neither zero nor one. Now, from $A\delta = A\delta'$ we obtain that $A[i,j]\delta[j] = A[i,j](\delta[j]+\varepsilon)$ for all $i \in \{1,\ldots,m\}$. Since $\varepsilon > 0$, this implies that $A[i,j] = 0$ for all $i \in \{1,\ldots,m\}$.

($\Leftarrow$) Let $\delta' \in \mathbb{R}^{n\times 1}$ be such that $\delta' \sim \delta$ and $(\delta' - \mathbf{1}) \sim (\delta - \mathbf{1})$. Note that this means that δ and δ' have zeroes and ones on exactly the same positions. Using that $A[i,j] = 0$ whenever $\delta[j] \notin \{0,1\}$, we have, for any $i \in \{1,\ldots,m\}$, that

$$(A\delta')[i] = \sum_{j=1}^{n} A[i,j]\delta'[j] = \sum_{j:\delta[j]=0,1} A[i,j]\delta'[j] =$$

$$= \sum_{j:\delta[j]=0,1} A[i,j]\delta[j] = \sum_{j=1}^{n} A[i,j]\delta[j] = (A\delta)[i]. \quad \blacksquare$$

We can now prove that Definition 14.3.1 induces exactly the notion that we want.

Theorem 14.3.4 Let $(\sigma, Q_s, \mathcal{Q}_f, \rho)$ be a Markov reward chain with silent transitions. It is $\tau_\sim$-reducible iff, for all $Q_f, Q_f' \in \mathcal{Q}_f$,

$$\sigma R = \sigma R', \; LQ_sR = L'Q_sR', \; \text{ and } \; L\rho = L'\rho,$$

where RL and $R'L'$ are canonical product decompositions of the ergodic projections of Q_f and Q_f' respectively. $\square$

Proof The theorem is proven only from right to left but, as the proof is based on Lemmas 14.3.2 and 14.3.3, the other direction can be constructed easily.

Let the numbering be such that it makes the ergodic partitioning explicit. Then

$$\sigma = \begin{pmatrix}\sigma_1 & \ldots & \sigma_M & \sigma_T\end{pmatrix}, \; Q_s = \begin{pmatrix} Q_{11} & \ldots & Q_{1M} & X_1 \\ \vdots & \ddots & \vdots & \vdots \\ Q_{M1} & \ldots & Q_{MM} & X_M \\ Y_1 & \ldots & Y_M & Z \end{pmatrix}, \; \rho = \begin{pmatrix}\rho_1 \\ \vdots \\ \rho_M \\ \rho_T\end{pmatrix}.$$

Let $Q_f \in \mathcal{Q}_f$. We obtain

$$L = \begin{pmatrix} \mu_1 & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \dots & \mu_M & \mathbf{0} \end{pmatrix}, R = \begin{pmatrix} \mathbf{1} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{1} \\ \delta_1 & \dots & \delta_M \end{pmatrix}.$$

We have $\sigma R = \big((\sigma_1 \cdot \mathbf{1} + \sigma_T \cdot \delta_1) \ \dots \ (\sigma_M \cdot \mathbf{1} + \sigma_T \cdot \delta_M)\big)$. Let δ'_L be such that $\delta'_L \sim \delta_L$ and $(\delta'_L - \mathbf{1}) \sim (\delta_L - \mathbf{1})$. Let R' be the same as R but with δ'_L instead of δ_L. From $\sigma R = \sigma R'$ we obtain $\sigma_T \cdot \delta_L = \sigma_T \cdot \delta'_L$. We can always find $Q'_f \sim Q_f$ such that $R'L$ is the canonical product decomposition of its ergodic projection. This means that $\sigma_T \cdot \delta_L = \sigma_T \cdot \delta'_L$ actually holds for all δ'_L of the above form. Now, by Lemma 14.3.3 (with $A = \sigma_T$) this can only be if, for all $1 \leqslant i \leqslant n$, either $\sigma_T[i] = 0$ or $\delta_L[i] \in \{0, 1\}$ for all $1 \leqslant L \leqslant M$. Since $R \cdot \mathbf{1} = \mathbf{1}$, the latter is only possible when there exists an $1 \leqslant K \leqslant M$ such that $\delta_K[i] = 1$. Recall that $\delta_K[i] = 1$ iff $\text{erg}(i) = E_K$. This proves that the first condition in Definition 14.3.1 holds.

We now show that Condition 2a holds. We have

$$LQ_sR = \begin{pmatrix} \mu_1 Q_{11}\mathbf{1} + \mu_1 X_1 \delta_1 & \dots & \mu_1 Q_{1M}\mathbf{1} + \mu_1 X_1 \delta_M \\ \vdots & \ddots & \vdots \\ \mu_M Q_{M1}\mathbf{1} + \mu_M X_M \delta_M & \dots & \mu_M Q_{MM}\mathbf{1} + \mu_M X_M \delta_M \end{pmatrix}.$$

From $LQ_sR = LQ_sR'$ we obtain $\mu_K X_K \delta_L = \mu_K X_K \delta'_L$. By Lemma 14.3.3, it follows that, for all $1 \leqslant K \leqslant M$ and all $1 \leqslant j \leqslant n$, either $(\mu_K X_K)[j] = 0$ or $\delta_L[j] \in \{0, 1\}$ for all $1 \leqslant L \leqslant M$. Note that, since $\mu_K > 0$, $(\mu_K X_K)[j] = 0$ iff $X_K[i, j] = 0$ for all $i \in E_K$. As before, $\delta_L[j] \in \{0, 1\}$ for all $1 \leqslant L \leqslant M$ only if $\delta_{L'}[j] = 1$, i.e. if $\text{erg}(i) = E_{L'}$, for some $1 \leqslant L' \leqslant M$.

To prove Condition 2b let μ'_K be a stochastic vector such that $\mu'_K \sim \mu_K$. Let L' be formed as L but with μ'_K instead of μ_K. From $LQ_sR = L'Q_sR$ we have $\mu_K(Q_{KL}\mathbf{1} + X_K\delta_L) = \mu'_K(Q_{KL}\mathbf{1} + X_K\delta_L)$. As before, we can always find $Q'_f \sim Q_f$ such that RL' is the canonical product decomposition of its ergodic projection. By Lemma 14.3.2, it follows that $Q_{KL}\mathbf{1} + X_K\delta_L = \alpha \cdot \mathbf{1}$ for some constant α. In other words, it follows that the rows of $Q_{KL}\mathbf{1} + X_K\delta_L$ are all the same. From what we showed before (in the proof of Condition 2a) $(X_K\delta_L)[i] = \sum_{\ell:\text{erg}(\ell)=E_L} X_K[i, \ell]$. Thus

$$\sum_{\ell \in E_L} Q_{KL}[i, \ell] + \sum_{\ell \in T:\text{erg}(\ell)=E_L} X_K[i, \ell] = \sum_{\ell \in E_L} Q_{KL}[j, \ell] + \sum_{\ell \in T:\text{erg}(\ell)=E_L} X_K[j, \ell]$$

for all $i, j \in E_K$. Since $\mathrm{erg}(\ell) = E_L$ when $\ell \in E_L$, we have

$$\sum_{\ell:\mathrm{erg}(\ell)=E_L} Q_s[i,\ell] = \sum_{\ell:\mathrm{erg}(\ell)=E_L} Q_s[j,\ell],$$

for all $i, j \in E_K$. This is Condition 2b.

For the reward vector we have $L\rho = \begin{pmatrix} \mu_1\rho_1 \\ \vdots \\ \mu_M\rho_M \\ \mathbf{0} \end{pmatrix}$. From $L\rho = L'\rho$ we obtain $\mu_K\rho_K = \mu'_K\rho_K$. From Lemma 14.3.2 it follows that $\rho_K = \mathbf{1} \cdot x_K$ for some row vector x_K. Note that this exactly means that $\rho[i] = \rho[j]$ for all $i, j \in E_K$. This is Condition 3 of Definition 14.3.1. ■

If $(\sigma, Q_s, \mathcal{Q}_f, \rho)$ is $\tau_{\sim}$-reducible, then we say that it $\tau_{\sim}$-*reduces* to the Markov reward chain $(\sigma R, I, LQ_sR, \rho R)$, where RL is the canonical product decomposition of the ergodic projection of some $Q_f \in \mathcal{Q}_f$. Theorem 14.3.4 guarantees that this definition is correct.

We now give some examples of $\tau_{\sim}$-reductions.

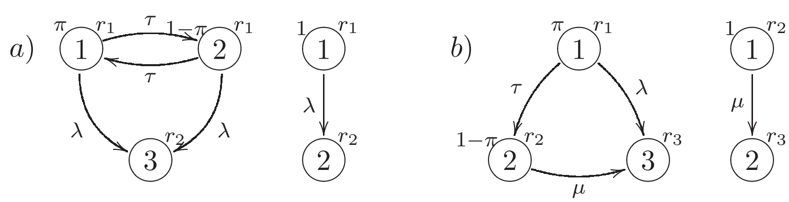


Figure 14.3: $\tau_{\sim}$-reduction – Example 14.3.5

Example 14.3.5 a. Consider the Markov reward chain with silent transitions depicted in Figure 14.3a on the left. This process can be τ-reduced because it does not have transient states and because every state in the ergodic class $\{1, 2\}$ does λ to the other ergodic class $\{3\}$. The process τ-reduces to the Markov reward chain depicted in Figure 14.3a on the right.

b. Consider the Markov reward chain with silent transitions depicted in Figure 14.3b on the left. This process can be τ-reduced because its ergodic classes are singletons, and because its only transient state, i.e., state 1, gets trapped only in the state 2. The τ-reduced process is depicted in Figure 14.3b on the right. □

We also give an example of Markov reward chains with silent transitions that are not $\tau_\sim$-reducible.

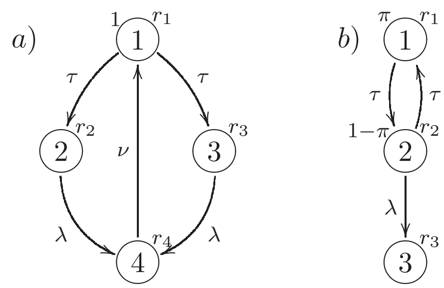


Figure 14.4: Markov reward chains with silent transitions that are not $\tau_\sim$-reducible – Example 14.3.6

Example 14.3.6 Consider the Markov reward chains with silent transitions from Figure 14.3c and Figure 14.3d. These Markov reward chains with silent transitions cannot be τ-reduced because they violate the first, resp. the second and third, condition of Definition 14.3.1. □

Note that the conditions of Definition 14.3.1 are very restrictive, and so not many Markov reward chains with silent transitions are $\tau_\sim$-reducible. The reason is that in most cases τ-reduction of a Markov reward chain with fast transitions will produce a Markov reward chain in which transitions do depend on the speeds of the fast transitions. The problem with the parameterized slow transitions can however, in some cases, be "repaired" by performing an ordinary lumping on the resulting Markov reward chain. In other words, even if LQ_sR depends on Q_f, it might be the case that its lumped version ULQ_sRV does not. We give an example.

Example 14.3.7 Consider the Markov reward chain with silent transitions from Figure 14.5a. First, we take a representative Markov reward chain with fast transitions such as the one from Figure 14.5b. Note that this Markov reward chain with fast transitions τ-reduces to the Markov reward chain in Figure 14.5c. This Markov reward chain depends on the parameters a and b. However, the states 1 and 2 can form a lumping class. The resulting lumped Markov reward chain is in Figure 14.5d. Note that the lumping removed the dependencies on the parameters. □

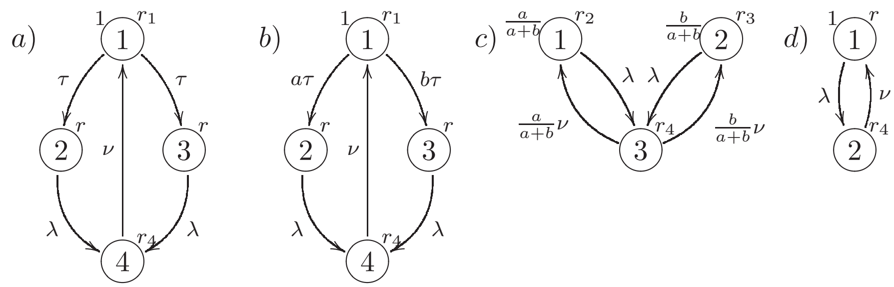


Figure 14.5: A total $\tau_{\sim}$ reduction – Example 14.3.7

We define a reduction method that combines τ-reduction with lumping and call it total $\tau_{\sim}$-reduction. In the definition we need to use the function called flat that gives a set of elements from a set of sets. Formally, if $C \in \mathcal{P}$, then $\mathrm{flat}(C) = \bigcup_{S \in C} S$.

Definition 14.3.8 (Total $\tau_{\sim}$-reduction) Let $(\sigma, Q_s, \mathcal{Q}_f, \rho)$ be a Markov reward chain with silent transitions. Let $\{E_1, \dots, E_M, T\}$ be its ergodic partitioning, and let $E = \bigcup_{1 \leqslant K \leqslant M} E_K$. Let $\mathcal{P}$ be a partitioning of $\{E_1, \dots, E_M\}$. Then $(\sigma, Q_s, \mathcal{Q}_f, \rho)$ is totally $\tau_{\sim}$-reducible (with respect to $\mathcal{P}$) if the following conditions hold:

1. for all $i \in T$, either $\sigma[i] = 0$ or $\mathrm{erg}(i) \subseteq \mathrm{flat}(C)$ for some $C \in \mathcal{P}$;
2. (a) for all $j \in T$, either $Q_s[i, j] = 0$ for all $i \in E$, or $\mathrm{erg}(j) \subseteq \mathrm{flat}(C)$ for some $C \in \mathcal{P}$;

   (b) for all $C, D \in \mathcal{P}$, $C \neq D$, and all $i, j \in \mathrm{flat}(C)$,

$$\sum_{\ell : \mathrm{erg}(\ell) \subseteq \mathrm{flat}(D)} Q_s[i, \ell] = \sum_{\ell : \mathrm{erg}(\ell) \subseteq \mathrm{flat}(D)} Q_s[j, \ell];$$

3. $\rho[i] = \rho[j]$ for every $i, j \in \mathrm{flat}(C)$. □

Note that the conditions for total $\tau_{\sim}$-reduction are very similar to those for τ-reduction. The only difference is that instead of an ergodic class E_L we work with the whole lumping class that contains it (that is why instead of $\mathrm{erg}(i) = E_L$ we have $\mathrm{erg}(i) \subseteq \mathrm{flat}(C)$). We note that in the trivial case when LQ_sR already does not depend on the choice from $\mathcal{Q}_f$, it is sufficient to use the trivial lumping induced by $V = I$. Then a total τ-reduction degrades to a τ-reduction.

The following theorem gives a characterization of total $\tau_\sim$-reduction, i.e. it shows that total $\tau_\sim$-reduction meets our requirements.

Theorem 14.3.9 Let $(\sigma, Q_s, \mathcal{Q}_f, \rho)$ be a Markov reward chain with silent transitions, and let $\mathcal{E} = \{E_1, \ldots, E_M, T\}$ be its ergodic partitioning. Let $\mathcal{P}$ be a partitioning of $\{E_1, \ldots, E_M\}$. Then $(\sigma, Q_s, \mathcal{Q}_f, \rho)$ is totally $\tau_\sim$-reducible with respect to $\mathcal{P}$ iff:

1. $VULQ_sRV = LQ_sRV$ and $VUL\rho = L\rho$, for every $Q_f \in \mathcal{Q}_f$; and

2. $\sigma RV = \sigma R'V$, $ULQ_sRV = UL'Q_sR'V$ and $UL\rho = UL'\rho$ for every $Q_f, Q_f' \in \mathcal{Q}_f$,

where RL and $R'L'$ are canonical product decompositions of the ergodic projections of Q_f and Q_f' respectively, V is the collector for $\mathcal{P}$, and U is a distributor for V. □

Proof Let the numbering be such that the partitioning $\mathcal{P} = \{C_1, \ldots, C_N\}$ is made explicit and then, inside every class also the ergodic partitioning $\mathcal{E}$ is made explicit. This is achieved by first numbering the ergodic classes as $E_{11}, \ldots, E_{1c_1}, \ldots, E_{N1}, \ldots, E_{Nc_N}$ with $C_K = \{E_{K1}, \ldots, E_{Kc_K}\}$ for $1 \leqslant K \leqslant N$. Then states are numbered to make the ergodic classes in each lumping class explicit.

We obtain the following forms for σ, Q_s, ρ, U and V:

$$\sigma = \begin{pmatrix}\sigma_1 & \ldots & \sigma_N & \sigma_T\end{pmatrix}, \ \sigma_K = \begin{pmatrix}\sigma_{K1} & \ldots & \sigma_{Kc_K}\end{pmatrix},$$

$$Q_s = \begin{pmatrix} Q_{11} & \ldots & Q_{1N} & X_1 \\ \vdots & \ddots & \vdots & \vdots \\ Q_{N1} & \ldots & Q_{NN} & X_N \\ Y_1 & \ldots & Y_N & Z \end{pmatrix}, \ \rho = \begin{pmatrix}\rho_1 \\ \vdots \\ \rho_N \\ \rho_T\end{pmatrix}, \ \rho_K = \begin{pmatrix}\rho_{11} \\ \vdots \\ \rho_{Kc_K}\end{pmatrix},$$

$$U = \begin{pmatrix} u_1 & \ldots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \ldots & u_N \end{pmatrix}, \ V = \begin{pmatrix} v_1 & \ldots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \ldots & v_N \end{pmatrix}, \quad \begin{matrix} u_K = \begin{pmatrix}u_{11} & \ldots & u_{Kc_K}\end{pmatrix}, \\ v_K = \begin{pmatrix}\mathbf{1} \\ \vdots \\ \mathbf{1}\end{pmatrix} \end{matrix} .$$

Let $Q_f \in \mathcal{Q}_f$. Then

$$L = \begin{pmatrix} \mu_1 & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \dots & \mu_N & \mathbf{0} \end{pmatrix}, \; R = \begin{pmatrix} R_1 & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & R_N \\ \delta_1 & \dots & \delta_N \end{pmatrix},$$

$$\mu_K = \begin{pmatrix} \mu_{K1} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mu_{Kc_K} \end{pmatrix}, \; R_K = \begin{pmatrix} \mathbf{1} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{1} \end{pmatrix}, \; \delta_K = \begin{pmatrix} \delta_{K1} & \dots & \delta_{Kc_K} \end{pmatrix}.$$

Define

$$\bar{L} = UL = \begin{pmatrix} m_1 & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \dots & m_N & \mathbf{0} \end{pmatrix}, \; m_K = \begin{pmatrix} u_{K1}\mu_{K1} & \dots & u_{Kc_K}\mu_{Kc_K} \end{pmatrix}$$

and

$$\bar{R} = RV = \begin{pmatrix} \mathbf{1} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{1} \\ d_1 & \dots & d_N \end{pmatrix}, \; d_K = \delta_K V_K = \sum_{\ell=1}^{c_K} \delta_{K\ell}.$$

($\Rightarrow$) First, we show that the lumping condition holds. We do this by showing that the rows of $LQ_s\bar{R}$, resp. $L\rho$, that correspond to the elements of the same class are equal.

It is not hard to show that Condition 2 of Definition 14.3.8 implies that, for all $1 \leqslant K, L \leqslant N$, all elements of the vector $Q_{KL}\mathbf{1} + X_K d_L$ are equal, i.e. that $Q_{KL}\mathbf{1} + X_K d_L = \mathbf{1} \cdot \alpha_{KL}$ for some $\alpha_{KL} \geqslant 0$. We obtain

$$LQ_s\bar{R} = \begin{pmatrix} \mu_1 \cdot (Q_{11} \cdot \mathbf{1} + X_1 d_1) & \dots & \mu_1 \cdot (Q_{1N} \cdot \mathbf{1} + X_1 d_N) \\ \vdots & \ddots & \vdots \\ \mu_N \cdot (Q_{N1} \cdot \mathbf{1} + X_N d_N) & \dots & \mu_N \cdot (Q_{NN} \cdot \mathbf{1} + X_N d_N) \end{pmatrix}.$$

Now, since $Q_{KL}\mathbf{1} + X_K d_L = \alpha_{KL} \cdot \mathbf{1}$ we have

$$\mu_K \cdot (Q_{KL}\mathbf{1} + X_K d_L) = \mu_K \cdot \alpha_{KL} \cdot \mathbf{1} =$$
$$= \alpha_{KL} \begin{pmatrix} \mu_{K1} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mu_{Kc_K} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{1} \\ \vdots \\ \mathbf{1} \end{pmatrix} = \alpha_{KL} \begin{pmatrix} \mu_{K1}\mathbf{1} \\ \vdots \\ \mu_{Kc_K}\mathbf{1} \end{pmatrix} = \alpha_{KL} \cdot \mathbf{1}.$$

From Condition 3 we obtain that $\rho_K = \alpha_K \cdot \mathbf{1}$ for some constant α_K. We also have

$$L\rho = \begin{pmatrix} \mu_1\rho_1 \\ \vdots \\ \mu_N\rho_N \\ \mathbf{0} \end{pmatrix}.$$

Now, since $\rho_K = \mathbf{1} \cdot \alpha_K$, with the same calculation as before, we obtain $\mu_K\rho_K = \alpha_K \cdot \mathbf{1}$. We conclude that the lumping condition holds.

Now suppose that $\bar{R}'$ is defined in a similar way as $\bar{R}'$. From $\sigma_T\delta_K = \sum_{i:d_K[i]=1}\sigma[i] = \sum_{i:d'_K[i]=1}\sigma[i]$ it easily follows that $\sigma\bar{R}' = \sigma\bar{R}$. That $L'Q_s\bar{R}' = LQ_s\bar{R}$ follows from $X_K d_L = X_K d'_L$ and $\mu_K \cdot (Q_{KL}\mathbf{1} + X_K d_L) = \alpha_{KL} \cdot \mathbf{1}$, both implied by Condition 2. Finally, that $L'\rho = L\rho$ follows from $\mu_K\rho_K = \alpha_K \cdot \mathbf{1} = \mu'_K\rho_K$.

($\Leftarrow$) Because of the lumping condition we can assume that $u_K > 0$ for all $1 \leqslant K \leqslant N$. Observe that the forms of $\bar{L}$ and $\bar{R}$ are very similar to the forms of L and R. Let $K \in \{1, \ldots, N\}$. Since $u_K > 0$, we have $m_K > 0$. Since the elements of μ_K range over all positive numbers, also the elements of m_K range over all positive numbers. Clearly, $0 \leqslant d_K \leqslant \mathbf{1}$ and since the elements of δ_K that are not in $\{0,1\}$ can take any value in $(0,1)$, the same holds for the elements of d_K. This allows us to proceed just as we did in the proof of Theorem 14.3.4 but with the matrices $\bar{L}$ and $\bar{R}$ instead of L and R.

First, we have that for all $1 \leqslant i \leqslant n$, either $\sigma_T[i] = 0$ or there is a $K \in \{1, \ldots, N\}$ such that $d_K[i] = 1$. Now, note that $d_K[i] = \sum_{L=1}^{c_K}\delta_{KL}$ is equal to 1 only if $\mathrm{erg}(i) \subseteq (E_{K1} \cup \cdots \cup E_{Kc_K}) = \mathrm{flat}(C_K)$. This gives us Condition 1.

Second, we have that a) for all $1 \leqslant j \leqslant n$, either $X_K[i,j] = 0$ for all $i \in C_K$, or $d_L[j] = 1$ for some $1 \leqslant L \leqslant N$, and b) the rows of $Q_{KL}\mathbf{1} + X_K d_L$ are all the same, i.e. $(Q_{KL}\mathbf{1} + X_K d_L)[i] = (Q_{KL}\mathbf{1} + X_K d_L)[j]$ for all i, j. Then

$$(Q_{KL}\mathbf{1})[i] + (X_K d_L)[i] = \sum_{\ell \in C_L} Q_{KL}[i,\ell] + \sum_{\substack{\ell\,:\,\ell \in T \\ \mathrm{erg}(\ell) = \mathrm{flat}(C_L)}} Q_s[i,\ell] = \sum_{\ell:\mathrm{erg}(\ell)=\mathrm{flat}(C_L)} Q_s[i,\ell].$$

Finally, for the reward vector, we have $\rho_K = \alpha_K \cdot \mathbf{1}$ for some constant α_K. Note that this exactly means that $\rho[i] = \rho[j]$ for all $i, j \in \mathrm{flat}(C_K)$. ■

If a Markov reward chain with silent transitions $(\sigma, Q_s, \mathcal{Q}_f, \rho)$ is totally $\tau_\sim$-reducible with respect to a partitioning $\mathcal{P}$, we say that it *totally $\tau_\sim$-reduces* to $(\sigma RV, I, ULQ_sRV, UL\rho)$, where RL is the canonical product decomposition of the ergodic projection of Q_f, V is the collector for $\mathcal{P}$, and U is a distributor for V.

We give an example.

Example 14.3.10 Consider the Markov reward chain with silent transitions from Figure 14.5a. Its ergodic partitioning is $\mathcal{E} = \{E_1, E_2, E_3, T\}$ where $E_1 = \{2\}$, $E_2 = \{3\}$ and $E_3 = \{4\}$. Define $\mathcal{P} = \{C_1, C_2\}$ where $C_1 = \{2, 3\}$ and $C_2 = \{4\}$. It is not hard to see that the conditions for total $\tau_\sim$-reducibility hold. The process totally $\tau_\sim$-reduces to the Markov reward chain depicted in Figure 14.5d. □

Chapter 15

Comparative Analysis

In this chapter we compare the lumping method with the reduction method. As both methods are shown to preserve performance (e.g. the total reward), we are interested in which of the two can aggregate more states with instantaneous behavior. We show that the methods are in general incomparable but that reduction combined with standard lumping (on the resulting Markov reward chain) gives in general better results. The main result of the chapter is that the notion of $\tau_{\sim}$-lumping coincides with the notion of total τ-reduction (in a particular non-degenerate case). At the end, we also show how $\tau_{\sim}$-lumping (and, hence, total τ-reduction too) compares with weak bisimulation for Interactive Markov chains from [61].

15.1 Reduction vs. ordinary lumping

In general, the reduction of a discontinuous Markov reward chain to a Markov reward chain and the ordinary lumping are incomparable. However, when reduction is combined with the standard ordinary lumping for Markov reward chains it becomes a superior method. We give an example.

Example 15.1.1 Recall, from Example 14.1.3b, that the discontinuous Markov reward chain (σ, Π, Q, ρ) defined by

$$\sigma = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}, \quad \Pi = \begin{pmatrix} 0 & p & 1-p & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$Q = \begin{pmatrix} 0 & -p\lambda & -(1-p)\mu & p\lambda + (1-p)\mu \\ 0 & -\lambda & 0 & \lambda \\ 0 & 0 & -\mu & \mu \\ \nu & 0 & 0 & -\nu \end{pmatrix}, \quad \text{and} \quad \rho = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{pmatrix}$$

was reduced to the Markov reward chain $(\hat{\sigma}, I, \hat{Q}, \hat{\rho})$ defined by

$$\hat{\sigma} = \begin{pmatrix} p & 1-p & 0 \end{pmatrix}, \ \hat{Q} = \begin{pmatrix} -\lambda & 0 & \lambda \\ 0 & -\mu & \mu \\ p\nu & (1-p)\nu & -\nu \end{pmatrix} \text{ and } \hat{\rho} = \begin{pmatrix} r_2 \\ r_3 \\ r_4 \end{pmatrix}.$$

Note however that, if $\lambda \neq \mu$, the process (σ, Π, Q, ρ) only has the trivial lumping (cf. Example 13.1.9c) and so, in this case, reduction performs better.

Ordinary lumping sometimes aggregates more than reduction. This is because lumping classes can contain states from different ergodic classes while reduction only aggregates whole ergodic classes and transient states. Lumping also gives more flexibility in the sense that one can obtain the (intermediate) lumped processes that are not necessarily Markov reward chains. The intermediate lumping steps can e.g. be used in the construction of algorithms. Consider again the same discontinuous Markov reward chain (σ, Π, Q, ρ) but with $\lambda = \mu$ and $r_2 = r_3 \stackrel{\text{def}}{=} r$. In Example 13.1.9c we showed that this process could be lumped to the discontinuous Markov reward chain

$$\check{\sigma} = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}, \ \check{\Pi} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \ \check{Q} = \begin{pmatrix} 0 & -\lambda & \lambda \\ 0 & -\lambda & \lambda \\ \nu & 0 & -\nu \end{pmatrix} \quad \text{and} \quad \check{\rho} = \begin{pmatrix} r_1 \\ r \\ r_4 \end{pmatrix},$$

or all the way to the Markov reward chain

$$\check{\sigma} = \begin{pmatrix} 1 & 0 \end{pmatrix}, \ \check{\Pi} = I, \ \check{Q} = \begin{pmatrix} -\lambda & \lambda \\ \nu & -\nu \end{pmatrix}, \quad \text{and} \quad \check{\rho} = \begin{pmatrix} r \\ r_4 \end{pmatrix}$$

These two processes cannot be obtained by reduction. □

Note that, although the last process in the previous example cannot be directly obtained by reduction, it can be obtained from the reduced process $(\hat{\sigma}, \hat{\Pi}, \hat{Q}, \hat{\rho})$ by the lumping $\{\{1, 2\}, \{3\}\}$. It is therefore interesting to compare the ordinary lumping method for discontinuous Markov reward chains with the combination of the reduction method and the standard lumping for Markov reward chains. The following theorem shows that reducing a discontinuous Markov reward chain to a Markov reward chain first, and then lumping it, produces, in general, better results then only doing the lumping from the start.

Theorem 15.1.2 Suppose $(\sigma, \Pi, Q, \rho) \overset{\mathcal{P}}{\rightsquigarrow} (\hat{\sigma}, \hat{\Pi}, \hat{Q}, \hat{\rho})$. If $\hat{\Pi} = I$, then there exists a collector matrix V_E such that

$$V_E U_E L Q R V_E = L Q R V_E, \quad V_E U_E L \rho = L \rho,$$

$$\hat{\sigma} = \sigma R V_E, \ \hat{Q} = U_E L Q R V_E \text{ and } \hat{\rho} = U_E L \rho,$$

where $RL = \Pi$ is the canonical product decomposition of Π, and U_E is a distributor associated to V_E. □

Proof Let V be the collector associated to $\mathcal{P} = \{C_1, \ldots, C_N\}$ and let U be its associated distributor. Let $\mathcal{E} = \{E_1, \ldots, E_M, T\}$ be the ergodic partitioning of (σ, Π, Q, ρ).

From $U\Pi V = I$, multiplying by V from the left and using that $VU\Pi V = \Pi V$, we obtain $\Pi V = V$. Define $V_E = LV$ and $U_E = UR$.

We fist show that V_E is a collector matrix. Suppose not. Then there exist $1 \leqslant K \leqslant M$ and $1 \leqslant L \leqslant N$ such that $(LV)[K, L] \notin \{0, 1\}$. From $(LV)[K, L] \neq 0$ we have that there is a $1 \leqslant i \leqslant n$ such that $L[K, i] \notin \{0, 1\}$ and $V[i, L] = 1$. This implies that $i \in E_K \cap C_L$. From $(LV)[K, L] \neq 1$ we obtain that $E_K \not\subseteq C_L$. Now, $(\Pi V)[i, L] = \sum_{j=1}^{n} \Pi[i, j] V[j, L] = \sum_{j \in E_K \cap C_L} \Pi[i, j] \notin \{0, 1\}$. Contradiction, because $\Pi V = V$. We conclude that V_E is a collector.

That U_E is a distributor associated to V_E follows from $U \geqslant 0$, $R \geqslant 0$ and $U_E V_E = URLV = U\Pi V = I$. Now, using that $\Pi Q = Q \Pi = Q$ and that $VUQV = QV$, we have

$$\begin{aligned} V_E U_E L Q R V_E &= LVURLQRLV = LVU\Pi Q \Pi V = \\ &= LVUQV = LQV = L\Pi Q \Pi V = LQRLV = LQRV_E. \end{aligned}$$

Similarly, using that $VU\rho = \rho$, we have

$$V_E U_E L \rho = LVURL\rho = LVU\Pi\rho = LVU\Pi VU\rho = L\Pi VU\rho = L\Pi\rho = L\rho.$$

In addition, $\sigma R V_E = \sigma RLV = \sigma \Pi V = \sigma V = \hat{\sigma}$,

$$U_E L Q R V_E = URLQRLV = U\Pi Q \Pi V = UQV = \hat{Q}$$

and

$$U_E L \rho = URL\rho = U\Pi\rho U\Pi VU\rho = UVU\rho = U\rho = \hat{\rho}.$$ ■

From the proof we can also see when a reduction and a lumping coincide. Clearly, this is only when $LV = I$ and $UR = I$. The first equality implies that lumping is performed such that each ergodic class is one partitioning class. The second equality implies that there are no transient states that are trapped to more than one ergodic class in the original process. This was the case for the discontinuous Markov reward chain from Example 12.1.7b, that was lumped (Example 13.1.9b) and reduced (Example 14.1.3b) to the same Markov reward chain.

15.2 τ-reduction vs. τ-lumping

As τ-reduction and τ-lumping are based on reduction and ordinary lumping respectively, it is not surprising that the two methods are again incomparable. Moreover, as expected, τ-reduction combined with ordinary lumping aggregates more than just τ-lumping.

We give an example that corresponds to Example 15.1.1.

Example 15.2.1 Consider the Markov reward chain with fast transitions depicted in Figure 15.1a. Example 14.1.3c shows that this Markov reward chain with fast transitions τ-reduces to the Markov reward chain from Figure 15.1b. This aggregation cannot be obtained by lumping. On the other hand, if $\lambda = \mu$, the process from Figure 15.1a τ-lumps to the Markov reward chain in Figure 15.1c by the lumping $\{\{1\},\{2,3\},\{4\}\}$, and to the one Figure 15.1d by the lumping $\{\{1,2,3\},\{4\}\}$. These aggregations cannot be obtained by reduction. However, when $\lambda = \mu$, the Markov reward chain from Figure 15.1b lumps by the standard lumping to the Markov reward chain in Figure 15.1d. Therefore, like in the case for reduction, although the aggregation methods are incomparable, τ-reduction combined with standard lumping is superior than just τ-lumping. □

Theorem 15.2.2 Suppose $(\sigma, Q_s, Q_f, \rho) \overset{\mathcal{P}}{\rightsquigarrow}_\tau (\hat{\sigma}, \hat{Q}_s, \hat{Q}_f, \hat{\rho})$. If $\hat{Q}_f = \mathbf{0}$, then there exists a collector matrix V_E such that

$$V_E U_E L Q_s R V_E = L Q_s R V_E, \quad V_E U_E L\rho = L\rho,$$

$$\hat{\sigma} = \sigma R V_E, \ \hat{Q}_s = U_E L Q_s R V_E \text{ and } \hat{\rho} = U_E L \rho,$$

where $RL = \Pi$ is the canonical product decomposition of Π, the ergodic projection of Q_f, and U_E is a distributor associated to V_E. □

Proof Since $\hat{Q}_f = \mathbf{0}$, we obtain $\hat{\Pi} = U\Pi V = I$. As in the proof of Theorem 13.2.13, this implies $\Pi V = V$ and $W\Pi = W$ where W is the τ-distributor used to define the τ-lumped process $(\hat{\sigma}, \hat{Q}_s, \hat{Q}_f, \hat{\rho})$. Let $V_E = LV$ and $U_E = WR$. That V_E is a collector matrix and that U_E is a distributor associated to it is shown in the proof of Theorem 15.1.2.

Now, using that $VW\Pi Q_s \Pi V = \Pi Q_s \Pi V$, we have

$$V_E U_E L Q_s R V_E = LVWRLQ_sRLV = \\ = LVW\Pi Q_s \Pi V = L\Pi Q_s \Pi V = LQ_sRLV = LQ_sRV_E.$$

Similarly, using that $VW\Pi\rho = \Pi\rho$, we have

$$V_E U_E L\rho = LVWRL\rho = LVW\Pi\rho = L\Pi\rho = L\rho.$$

In addition, $\sigma RV_E = \sigma RLV = \sigma\Pi V = \sigma V = \hat{\sigma}$,

$$U_E L Q_s R V_E = WRLQ_sRLV = W\Pi Q_s \Pi V = WQV = \hat{Q}_s$$

and

$$U_E L\rho = WRL\rho = W\Pi\rho = W\rho = \hat{\rho}. \qquad \blacksquare$$

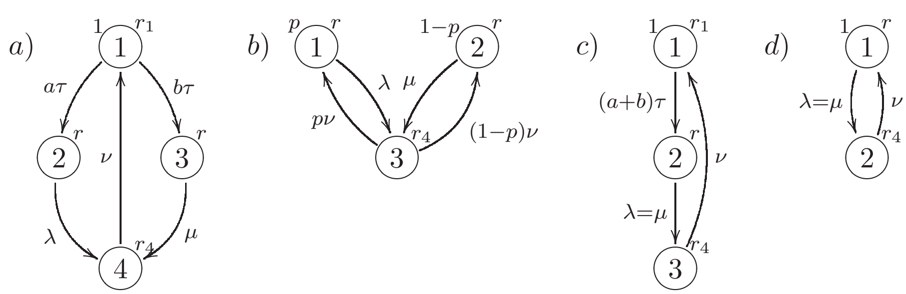


Figure 15.1: τ-reduction vs. τ-lumping –Example 15.2.1

Both techniques can produce the same simplified process only in the case when no transient states are trapped to more than one ergodic class. In this case τ-lumping must be performed such that the lumping classes contain complete ergodic classes together with all the transient states that lead to them. The Markov reward chain with fast transitions from Figure 15.2a reduces (Example 14.1.3b) and lumps (Example 13.2.4b) to the same Markov reward chain in Figure 15.2b.

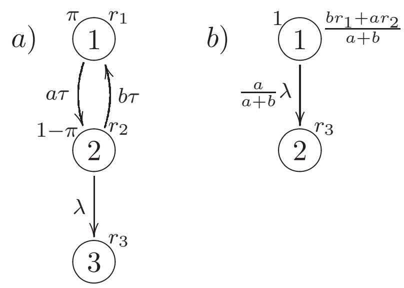


Figure 15.2: τ-reduction sometimes coincides with τ lumping

15.3 $\tau_\sim$-reduction vs. $\tau_\sim$-lumping

In this section we compare $\tau_\sim$-lumping with $\tau_\sim$- and total $\tau_\sim$-reduction. We show that in non-degenerate cases $\tau_\sim$-reduction is just a special instance of $\tau_\sim$-lumping, and that $\tau_\sim$-lumping and total $\tau_\sim$-reduction coincide when lumping eliminates all silent transitions.

The following example shows that in some cases $\tau_\sim$-lumping aggregates more than $\tau_\sim$-reduction.

Example 15.3.1 Consider the Markov reward chain with silent transitions depicted in Figure 15.3a. This process $\tau_\sim$-lumps to the Markov reward chain in Figure 15.3b by the lumping $\{\{1,2,3\},\{4\}\}$. However, the process in Figure 15.3a cannot be τ-reduced because the state 1 violates the condition that a transient state must lead to exactly one ergodic class. □

We now prove that $\tau_\sim$-reduction is a special case of $\tau_\sim$-lumping in case the process does not have unreachable states.

Definition 15.3.2 A state i is a *reachable* state if there exists $j_0,\ldots,j_m$ such that $\sigma[j_0]\neq 0$, $j_m = i$, and, for all $0\leqslant k\leqslant m$, either $Q_s[j_k,j_{k+1}]>0$ or $Q_f[j_k,j_{k+1}]>0$. □

Theorem 15.3.3 Suppose $(\sigma, Q_s, \mathcal{Q}_f, \rho)$ $\tau_\sim$-reduces to $(\sigma R, I, LQ_sR, L\rho)$. If $(\sigma, Q_s, \mathcal{Q}_f, \rho)$ does not have unreachable states, then there exists a partitioning $\mathcal{P}$ such that $(\sigma, Q_s, \mathcal{Q}_f, \rho) \overset{\mathcal{P}}{\rightsquigarrow}_{\tau_\sim} (\sigma V, WQ_sV, \{\mathbf{0}\}, W\rho)$, where V is the collector associated to $\mathcal{P}$ and W is a τ-distributor associated to Q_f. Moreover, $V = R$ and $W = L$. □

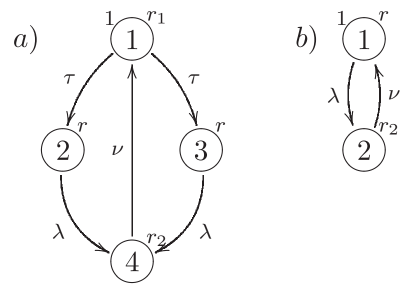


Figure 15.3: The process in a) $\tau_\sim$-lumps to the one in b) but cannot be $\tau_\sim$-reduced – Example 15.3.1

Proof Let $\mathcal{E} = \{E_1, \ldots, E_S, T\}$ be the ergodic partitioning of the Markov reward chain with silent transitions $(\sigma, Q_s, [Q_f]_\sim, \rho)$. We first show that for all $t \in T$ there is a $L \in \{1, \ldots, S\}$ such that $\text{erg}(t) = E_L$.

Since $(\sigma, Q_s, [Q_f]_\sim, \rho)$ does not have unreachable states, we have that there exist $i_0, \ldots, i_m$ such that $\sigma[i_0] \neq 0$, $i_m = t$, and, for all $0 \leqslant k \leqslant m$, either $Q_s[i_k, i_{k+1}] > 0$ or $Q_f[i_k, i_{k+1}] > 0$. We first prove, by induction on m, that $\text{erg}(t) = E_L$ for some $L \in \{1, \ldots, S\}$.

If $m = 0$, then $\sigma[i_0] \neq 0$ and the statement follows from the first condition in Definition 14.3.1. Suppose the statement holds for all $k \leqslant m$. Now, if $Q_f[i_k, i_{k+1}] > 0$ then, because $t = i_{m+1} \in T$ also $i_m \in T$. By the inductive hypothesis $\text{erg}(i_m) = E_L$ for some $1 \leqslant L \leqslant S$. Since $\text{erg}(i_{m+1}) \subseteq \text{erg}(i_m)$, we have $\text{erg}(i_{m+1}) \subseteq E_L$ and so $\text{erg}(i_{m+1}) = E_L$. If $Q_s[i_k, i_{k+1}] > 0$, then the statement follows from Condition 2a of Definition 14.3.1.

We now construct the lumping partitioning. Define $F_I = E_I \cup \{t \mid t \in T, \text{erg}(t) = E_I\}$, for $1 \leqslant I \leqslant S$ and let $\mathcal{P} = \{F_1, \ldots, F_S\}$. Since $\text{erg}(F_I) = E_I$ and $F_I = \{i \mid \text{erg}(i) = E_I\}$, if follows that $\mathcal{P}$ satisfies the conditions of Definition 13.3.2.

To show that the $\tau_\sim$-lumped and the $\tau_\sim$-reduced process coincide, we show that R is always a collector matrix. Because $\mathcal{P}$ has S elements this gives us $V = R$. Suppose that R is not a collector. Then there are $1 \leqslant K \leqslant S$ and i such that $\delta_K[i] \notin \{0, 1\}$. From this it follows that there is no $1 \leqslant L \leqslant S$ such that $\text{erg}(i) \subseteq E_L$ which is a contradiction.

The matrix L is a τ-distributor because $LR = I$ and because, for $\Pi = RL$, it satisfies $\Pi RL\Pi = \Pi\Pi\Pi = \Pi\Pi = \Pi RL$. ■

Note that, in the degenerate case when there are unreachable states it can happen that $\tau_\sim$-reduction can be applied but $\tau_\sim$-lumping cannot. This is because lumping must work for any initial vector. An example is given in Figure 15.4 (states 1 and 3 are unreachable).

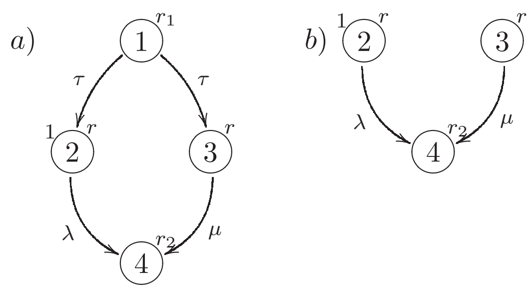


Figure 15.4: The process in a) $\tau_\sim$-reduces to the one in b) but cannot be (properly) $\tau_\sim$-lumped

We now compare $\tau_\sim$-lumping with total $\tau_\sim$-reduction. The following two theorems show that the notions coincide.

Theorem 15.3.4 Let $(\sigma, Q_s, \mathcal{Q}_f, \rho)$ be a Markov reward chain with silent transitions and let be $\mathcal{E} = \{E_1, \ldots, E_M, T\}$ be its ergodic partitioning. Suppose $(\sigma, Q_s, \mathcal{Q}_f, \rho) \overset{\mathcal{P}}{\rightsquigarrow} (\hat{\sigma}, \hat{Q}_s, \{\mathbf{0}\}, \hat{\rho})$. Then there exists a partitioning $\mathcal{P}_\mathcal{E}$ of $\{E_1, \ldots, E_M\}$ such that $(\sigma, Q_s, \mathcal{Q}_f, \rho)$ totally $\tau_\sim$-reduces to $(\hat{\sigma}, I, \hat{Q}_s, \hat{\rho})$ with respect to $\mathcal{P}_\mathcal{E}$. □

Proof Since $\hat{\mathcal{Q}}_f = \{\mathbf{0}\}$, we have that for every $C \in \mathcal{P}$ and every $i \in C$, $\mathrm{erg}(i) \subseteq C$. This implies that if $i \in C \cap E_K$, for some $1 \leqslant K \leqslant M$, then $E_K \subseteq C$. Intuitively, every lumping class must contain whole ergodic classes. Define, for each $C \in \mathcal{P}$, $e(C) = \{E_K \mid E_K \subseteq C\}$ and define $\mathcal{P}_\mathcal{E} = \{e(C) \mid C \in \mathcal{P}\}$. Clearly, $\mathcal{P}_\mathcal{E}$ is a partitioning of $\{E_1, \ldots, E_M\}$. Observe that $\mathrm{flat}(e(C)) = \bigcup_{E_K \subseteq C} E_K = C \cap \bigcup_{L=1}^{M} E_L$. With this, the conditions of Definition 13.3.2 directly imply the conditions of Definition 14.3.9.

To show that the results of the lumping and the reduction are the same let V and $V_\mathcal{E}$ be the collectors associated to $\mathcal{P}$ and $\mathcal{P}_\mathcal{E}$ respectively. We choose a Q_f and obtain Π, L, R and W. It is not hard to show that $V_\mathcal{E} = LV$. From $\hat{Q}_f = WQ_fV = \mathbf{0}$ it follows, as before, that $\Pi V = V$ and that $W\Pi = W$. Define $U_\mathcal{E} = WR \geqslant 0$. Now $U_\mathcal{E} V_\mathcal{E} = WRLV = W\Pi V = WV = I$ and so $U_\mathcal{E}$ is an distributor for $V_\mathcal{E}$. Finally, $U_\mathcal{E} L = WRL = W\Pi = W$ and $RV_\mathcal{E} = RLV = \Pi V = V$. ■

Theorem 15.3.5 Let $(\sigma, Q_s, \mathcal{Q}_f, \rho)$ be a Markov reward chain with silent transitions that does not have unreachable states. Let $\mathcal{E} = \{E_1, \ldots, E_M, T\}$ be its ergodic partitioning and let $\mathcal{P}_\mathcal{E}$ be some partitioning of $\{E_1, \ldots, E_M\}$. If $(\sigma, Q_s, \mathcal{Q}_f, \rho)$ totally $\tau_\sim$-reduces with respect to $\mathcal{P}_\mathcal{E}$ to $(\hat{\sigma}, I, \hat{Q}_s, \hat{\rho})$, then there is a partitioning $\mathcal{P}$ such that $(\sigma, Q_s, \mathcal{Q}_f, \rho)$ $\tau_\sim$-lumps to $(\hat{\sigma}, \hat{Q}_s, \{\mathbf{0}\}, \hat{\rho})$ with respect to $\mathcal{P}$. □

Proof In the same way as we did in the proof of Theorem 15.3.3, we can show that for all $t \in T$ there is a $C \in \mathcal{P}$ such that $\mathrm{erg}(t) = \mathrm{flat}(C)$.

We first define, for each $C \in \mathcal{P}$, $s(C) = \{i \mid \mathrm{erg}(i) \subseteq \mathrm{flat}(C)\}$. Next, we define $\mathcal{P} = \{s(C) \mid C \in \mathcal{P}_\mathcal{E}\}$. We show that $\mathcal{P}$ is a $\tau_\sim$-lumping.

Let $i \in s(C)$. Then $\mathrm{erg}(i) \subseteq \mathrm{flat}(C)$ and so $\mathrm{erg}(s(C)) \subseteq \mathrm{flat}(C) \subseteq s(C)$. This proves Condition 1a of Definition 13.3.2. The other two conditions follow directly from $s(C) \cap \mathrm{flat}(\{E_1, \ldots, E_M\}) = \mathrm{flat}(C)$.

We now show that the aggregated chains are the same.

We fix Q_f and obtain Π, L and R. Let $V_\mathcal{E}$ be the collector associated to $\mathcal{P}_\mathcal{E}$. Define $V = RV_\mathcal{E}$. From the definition of $\mathcal{P}$ it follows directly that V is the collector for $\mathcal{P}$. Let $U_\mathcal{E}$ be a distributor for V such that $V_\mathcal{E}[i,k] = 1$ implies $U_\mathcal{E}[k,i] > 0$. Define $W = U_\mathcal{E} L$. That W is a τ-distributor follows from $W\Pi = U_\mathcal{E} LRL = U_\mathcal{E} L = W$ and Lemma 13.2.9. ■

15.4 $\tau_\sim$-lumping vs. weak bisimulation for Interactive Markov chains

We have already mentioned that the aggregation method for the elimination of vanishing markings in generalized stochastic Petri nets is a special instance of τ-reduction. In this section we compare the $\tau_\sim$-lumping method with the weak bisimulation method for the elimination of τ transitions in Interactive Markov chains [61].

Recall that Interactive Markov chains are extensions of Markov chains with separate transitions that are labeled by actions. Weak bisimulation is an equivalence relation on Interactive Markov chains that abstracts away from transitions labeled by the internal τ action. For comparison, we assume that there are no other actions but τ actions (note that weak bisimulation works in the general case as well). We also assume that there are no rewards associated to states. In addition, we do not allow silent transitions to lead from a state to itself. As we treat them as exponential rates, they are redundant. We now list the cases where $\tau_\sim$-lumping (or $\tau_\sim$-reduction) is different from the reduction modulo weak bisimulation.

We give priority to silent transitions over exponential delays only in transient states (see Example 13.3.4a) and not in ergodic states (see Example 13.3.1a). This leads to a different treatment of τ-divergence. For us, an infinite avoidance of an exponential delay is not possible. The transition must eventually be taken after an exponential delay (see Example 13.3.4b). This can be considered as some kind of fairness incorporated in the model. Due to the strong requirement that the lumping of Markov reward chains with silent transitions is good if it is good for all possible speeds assigned to silent transitions, $\tau_\sim$-lumping does not always allow the aggregation of states that lead to different ergodic classes (see Example 13.3.1b) unless these ergodic classes are also inside some lumping class. This means that we only disallow certain intermediate lumping steps while weak bisimulation does not. In all other cases, the weak bisimilarity of Interactive Markov chains and $\tau_\sim$-lumping coincide.

Interactive Markov chain model is the underlying model of the process algebra Interactive Markov Chains. Weak bisimulation is shown to be a congruence, i.e. compatible with all the operators of this algebra. This is a very important property because it allows for aggregation of the (usually much smaller) components first, and then composing them into the aggregated system. In our case, compositionality is not crucial. The purpose of $\tau_\sim$-lumping and $\tau_\sim$-reduction is only to minimize final models, i.e. models that no longer interact with the environment. However, it is not hard to show that all the aggregation techniques that we introduced are compatible with the parallel operator (or, in matrix terms, with Kronecker product and sum [55]). This is very useful even if the parallel structure of the model is not know. We can, for example, first decompose a very large Markov reward chain with silent transitions into a set of independent parallel components and then $\tau_\sim$-reduce each component. The additional benefit is that the solution techniques for Markov chains can also effectively exploit the decomposition further [24].

Conclusion to Part III

We formalized the notion of fast and silent transitions in extensions of continuous-time Markov reward chains arising from high-level specifications. We treated fast transitions and silent steps as exponentially distributed delays of which the rates tend to infinity with determined and undetermined speeds, respectively. We introduced and compared two different aggregation techniques for the elimination of these transitions, one based on reduction and the other based on lumping.

In the case of fast transitions we showed that the techniques, in general, produce incomparable Markov reward chains, and we identified when the resulting processes coincide. The combination of reduction and ordinary lumping proves to be superior by its ability to reduce a given Markov reward chain with fast transitions. The analysis suggests that this combination can be successfully used to handle probabilistic choices in Markov reward chain-based extensions.

For the setting with silent steps the reduction method happens to be weaker than the lumping method. However, when reduction is combined with ordinary lumping, both aggregation techniques produce the same simplified processes (provided that all silent steps are eliminated by lumping).

The reduction method always removes all fast transitions, whereas the approach based on lumping does not. The advantage of reduction is its ability to split transient states. The lumping method provides more flexibility in the sense that it is not mandatory to eliminate all fast/silent transitions at once, so all intermediate processes can be obtained.

The reduction method in the setting with fast transitions coincides with the method of elimination of vanishing markings in generalized stochastic Petri nets. Our results can also be used to extend those methods by dropping the requirement that the probabilities of the immediate transitions must be stated explicitly. We also compared our techniques with the weak bisimulation reduction method for Interactive Markov chains. We pointed out some important differences and explained that in most cases the two approaches coincide.

We did not provide any algorithms nor real world examples. Algorithms will be considered in future work. Since our main contribution is the theory of elimination of instantaneous states coming from very popular Markovian specification formalisms, examples where our results can be applied are found elsewhere. However, still in the absence of tooling, we cannot apply them in big case studies. This is not a serious drawback. One of our results is that the lumping method in the non-deterministic setting only differs from the weak bisimulation reduction method from [61] in cases that we think will not appear in real world examples (presence of divergence). This implies that the tooling for Interactive Markov chains is applicable in our setting as well.

At last, let us provide a link between the theory presented in Part III and the theory of labeled transition systems. In Part II we presented the notions of strong and weak bisimulation for transition systems (with successful termination) in terms of matrix equations. By comparing these equations with the conditions for ordinary and for τ-lumping, we can conclude that ordinary lumping coincides with strong bisimulation, and that τ-lumping can be interpreted as weak bisimulation for Markov (reward) chains.

Bibliography

[1] R.P. Agaev and P.Y. Chebotarev. On determining the eigenprojection and components of a matrix. *Automated Remote Control*, 63:1537–1545, 2002.

[2] R. Alur, C. Courcoubetis, and D.L. Dill. Model-checking in dense real-time. *Information and Computation*, 104(1):2–34, 1993.

[3] H.H. Ammar, Y.F. Huang, and R.W. Liu. Hierarchical models for systems reliability, maintainability, and availability. *IEEE Transactions on Circuits and Systems*, 34(6):629–638, 1987.

[4] F. Baccelli, G. Cohen, G. Olsder, and J.P. Quadrat. *Synchronization and Linearity*. Wiley, 1992.

[5] J.C.M. Baeten and J.A. Bergstra. Global renaming operators in concrete process algebra. *Information and Computation*, 78(3):205–245, 1988.

[6] J.C.M. Baeten, J.A. Bergstra, and M.A. Reniers. Discrete time process algebra with silent step. In *Proof, language, and interaction: essays in honour of Robin Milner*, pages 535–569. MIT Press, Cambridge, MA, USA, 2000.

[7] J.C.M. Baeten and C.A. Middelburg. *Process Algebra with Timing*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.

[8] J.C.M. Baeten and M.A. Reniers. Duplication of constants in process algebra. *Journal of Logic and Algebraic Programming*, 70(2):151–171, 2007.

[9] J.C.M. Baeten and W.P. Weijland. *Process Algebra*. Number 18 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1990.

[10] C. Baier, J.P. Katoen, H. Hermanns, and V. Wolf. Comparative branching-time semantics for Markov chains. *Information and Computation*, 200(2):149–214, 2005.

[11] T. Basten. Branching bisimilarity is an equivalence indeed! *Information Processing Letters*, 58(3):141–147, 1996.

[12] D.A. van Beek, A. van der Ham, and J.E. Rooda. Modelling and control of process industry batch production systems. In *15th Triennial World Congress of the International Federation of Automatic Control*, Barcelona, Spain, 2002.

[13] H.M.A. van Beek. *Specification and analysis of Internet applications.* PhD thesis, Eindhoven University of Technology, 2005.

[14] R. Berghammer, B. von Karger, and C. Ulke. Relation-algebraic analysis of Petri nets with RELVIEW. In *TACAS '96: Proceedings of the Second International Workshop on Tools and Algorithms for Construction and Analysis of Systems*, volume 1055 of *LNCS*, pages 49–69. Springer, 1996.

[15] J.A. Bergstra and J.W. Klop. Algebra of communicating processes with abstraction. *Theoretical Computer Science*, 37:77–121, 1985.

[16] J.A. Bergstra, A. Ponse, and M.B. van der Zwaag. Branching time and orthogonal bisimulation equivalence. *Theoretical Computer Science*, 309(1):313–355, 2003.

[17] S. Blom, W. Fokkink, J.F. Groote, I. van Langevelde, B. Lisser, and J.C. van de Pol. μCRL: A toolset for analysing algebraic specifications. In *Computer Aided Verification, 13th International Conference (CAV 2001), Paris, France, July 18-22, 2001*, volume 2102 of *LNCS*, pages 250–254. Springer, 2001.

[18] E. Bortnik, N. Trčka, A.J. Wijs, S.P. Luttik, J.M. van de Mortel-Fronczak, J.C.M. Baeten, W.J. Fokkink, and J.E. Rooda. Analyzing a χ model of a turntable system using Spin, CADP and UPPAAL. *Journal of Logic and Algebraic Programming*, 65:51–104, 2005.

[19] V. Bos and J.J.T. Kleijn. Automatic verification of a manufacturing system. *Robotics and Computer Integrated Manufacturing*, 17:185–198, 2001.

[20] V. Bos and J.J.T. Kleijn. *Formal Specification and Analysis of Industrial Systems*. PhD thesis, Eindhoven University of Technology, 2002.

[21] J. Bradfield and C. Stirling. Modal logics and mu-calculi: an introduction. In *Handbook of Process Algebra*, chapter 1, pages 3–99. Elsevier, 2001.

[22] M.C. Browne, E.M. Clarke, and O. Grumberg. Characterizing finite Kripke structures in propositional temporal logic. *Theoretical Computer Science*, 59:115–131, 1988.

[23] P. Buchholz. Exact and ordinary lumpability in finite Markov chains. *Journal of Applied Probability*, 31:59–75, 1994.

[24] P. Buchholz and P. Kemper. Kronecker based matrix representations for large Markov chains. In *Validation of Stochastic Systems*, volume 2925 of *LNCS*, pages 256–295, 2004.

[25] S.L. Campbell. *Singular Systems of Differential Equations I*. Pitman, 1980.

[26] E.J.J. van Campen. *Design of a Multi-Process Multi-Product Wafer Fab*. PhD thesis, Eindhoven University of Technology, 2000.

[27] D. Cerotti, S. Donatelli, A. Horvath, and J. Sproston. CSL model checking for generalized stochastic Petri nets. In *Third International Conference on the Quantitative Evaluation of Systems (QEST 2006), September 2006, Riverside, California, USA*, pages 199–210. IEEE Computer Society, 2006.

[28] K.L. Chung. *Markov Chains with Stationary Probabilities*. Springer, 1967.

[29] G. Ciardo, J. Muppala, and K.S. Trivedi. On the solution of GSPN reward models. *Performance Evaluation*, 12:237–253, 1991.

[30] E.M. Clarke and E.A. Emerson. Design and synthesis of synchronization skeletons using branching-time temporal logic. In *Logic of Programs, Workshop*, pages 52–71, London, UK, 1982. Springer-Verlag.

[31] E.M. Clarke, O. Grumberg, and D.A. Peled. *Model Checking*. The MIT Press, 2000.

[32] M. Coderch, A.S. Willsky, S.S. Sastry, and D.A. Castanon. Hierarchical aggregation of singularly perturbed finite state Markov processes. *Stochastics*, 8:259–289, 1983.

[33] G. Cohen, S. Gaubert, and J.P. Quadrat. Projection and aggregation in maxplus algebra. In *Current Trends in Nonlinear Systems and Control, in Honor of Petar Kokotovic and Turi Nicosia*. Birkhauser, 2006.

[34] J.H. Conway. *Regular Algebra and Finite Machines*. Chapman and Hall; London, 1971.

[35] D. Dams. *Abstract Interpretation and Partition Refinement for Model Checking*. PhD thesis, Eindhoven University of Technology, 1996.

[36] R. De Nicola and F.W. Vaandrager. Action versus state based logics for transition systems. In *LITP spring school on theoretical computer science on Semantics of systems of concurrent processes*, volume 469 of *LNCS*, pages 407–419. Springer-Verlag, 1990.

[37] R. De Nicola and F.W. Vaandrager. Three logics for branching bisimulation. *JACM*, 42(2):458–487, 1995.

[38] F. Delebecque. A reduction process for perturbed Markov chains. *SIAM Journal of Applied Mathematics*, 2:325–330, 1983.

[39] F. Delebecque and J.P. Quadrat. Optimal control of Markov chains admitting strong and weak interactions. *Automatica*, 17:281–296, 1981.

[40] E.W. Dijkstra. *A Discipline of Programming*. Prentice-Hall, 1976.

[41] W. Doeblin. Sur l'équation matricielle a(t+s) = a(t) a(s) et ses applications aux probabilités en chaine. *Bull. Sci. Math.*, 62:21–32, 1938.

[42] J.L. Doob. *Stochastic Processes*. Wiley, 1953.

[43] E.A. Emerson. Temporal and modal logic. In *Handbook of theoretical computer science (vol. B): formal models and semantics*, pages 995–1072. MIT Press, Cambridge, MA, USA, 1990.

[44] E.A. Emerson and J.Y. Halpern. "Sometimes" and "not never" revisited: on branching versus linear time temporal logic. *JACM*, 33(1):151–178, 1986.

[45] F.O. Farid. Criteria for invertibility of diagonally dominant matrices. *Linear Algebra and Applications*, 215:63–93, 1995.

[46] J.J.H. Fey. *Design of a Fruit Juice Blending and Packaging Plant*. PhD thesis, Eindhoven University of Technology, 2000.

[47] M. Fitting. Bisimulations and boolean vectors. In *Advances in Modal Logic*, pages 97–126. King's College Publications, 2002.

[48] W. Fokkink, J. Pang, and A. Wijs. Is timed branching bisimilarity an equivalence indeed?. In *Formal Modeling and Analysis of Timed Systems, Third International Conference, FORMATS 2005, Uppsala, Sweden, September 26-28*, volume 3829 of *LNCS*, pages 258–272, 2005.

[49] A. Fronk. Using relation algebra for the analysis of Petri nets in a CASE tool based approach. In *SEFM '04: Proceedings of the Software Engineering and Formal Methods, Second International Conference on (SEFM'04)*, pages 396–405, Washington, DC, USA, 2004. IEEE Computer Society.

[50] R. Gerth, R. Kuiper, D. Peled, and W. Penczek. A partial order approach to branching time logic model checking. *Information and Computation*, 150(2):132–152, May 1999.

[51] R.J. van Glabbeek. The linear time – branching time spectrum II; the semantics of sequential systems with silent moves. In *Proceedings of the 4th International Conference on Concurrency Theory (CONCUR'93)*, volume 715 of *LNCS*, pages 66–81. Springer, 1993.

[52] R.J. van Glabbeek. The linear time – branching time spectrum I; the semantics of concrete, sequential processes. In *Handbook of Process Algebra*, chapter 1, pages 3–99. Elsevier, 2001.

[53] R.J. van Glabbeek and W.P. Weijland. Branching time and abstraction in bisimulation semantics. *JACM*, 43(3):555–600, 1996.

[54] J.A. Govaarts. Efficiency in a lean assembly line: a case study at NedCar Born. Master's thesis, Stan Ackermans Institute, Eindhoven, 1997.

[55] A. Graham. *Kronecker Products and Matrix Calculus With Applications*. Ellis Horwood Lim., Chichester, 1981.

[56] W.K. Grassmann and Y. Wang. Immediate events in Markov chains. In *Computations with Markov chains*, pages 163–176. Kluwer, 1995.

[57] J.F. Groote and A. Ponse. Process algebra with guards: Combining Hoare logic with process algebra. *Formal Aspects of Computing*, 6(2):115–164, 1994.

[58] D.J. Hartfiel. *Markov Set-Chains*. Springer-Verlag, 1998.

[59] M.C.B. Hennessy and R. Milner. On observing nondeterminism and concurrency. In *Automata, Languages and Programming, 7th Colloquium*, volume 85 of *LNCS*, pages 299–309, Noordweijkerhout, The Netherland, 1980. Springer-Verlag.

[60] M.C.B. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *JACM*, 32(1):137–161, 1985.

[61] H. Hermanns. *Interactive Markov Chains: The Quest for Quantified Quality*, volume 2428 of *LNCS*. Springer, 2002.

[62] E. Hille and R.S. Phillips. *Functional Analysis and Semi-Groups*. AMS, 1957.

[63] J. Hillston. *A Compositional Approach to Performance Modelling*. Cambridge University Press, 1996.

[64] C.A.R. Hoare. *Communicating Sequential Processes*. Prentice Hall, Englewood Cliffs, 1985.

[65] G.J. Holzmann. *The SPIN Model Checker*. Addison-Wesley, 2003.

[66] R.A. Howard. *Semi-Markov and Decision Processes*. London: Wiley, 1971.

[67] J.G. Kemeny and J.L. Snell. *Finite Markov chains*. Springer, 1976.

[68] K.H. Kim. *Boolean Matrix Theory and Applications*. Marcel Dekker, 1982.

[69] J.J. Koliha and T.D. Tran. Semistable operators and singularly perturbed differential equations. *Journal of Mathematical Analysis and Applications*, 231:446–458, 1999.

[70] C.P.J. Koymans and J.L.M. Vrancken. Extending process algebra with the empty process. Logic Group Preprint Series 1, State University of Utrecht, 1985.

[71] D. Kozen. Results on the propositional mu-calculus. *Theoretical Computer Science*, 27:333–354, 1983.

[72] D. Kozen. A completeness theorem for Kleene algebras and the algebra of regular events. *Information and Computation*, 110(2):366–390, 1994.

[73] B. Luttik. *Choice Quantification in Process Algebras*. PhD thesis, University of Amsterdam, 2002.

[74] B. Luttik and N. Trčka. Stuttering congruence for χ. In *SPIN'05*, San Francisco, California, USA, 2005.

[75] N. Lynch and F. Vaandrager. Forward and backward simulations i.: untimed systems. *Information and Computation*, 121(2):214–233, 1995.

[76] M.A. Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets*. Wiley, 1995.

[77] M.A. Marsan, G. Conte, and G. Balbo. A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems. *ACM Transactions on Computer Systems*, 2(2):93–122, 1984.

[78] R. Milner. *A Calculus of Communicating Systems*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1982.

[79] R. Milner. Operational and algebraic semantics of concurrent processes. In *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B)*, pages 1201–1242. Elsevier and MIT Press, 1990.

[80] M. Mousavi. *Structuring Structural Operational Semantics*. PhD thesis, Eindhoven University of Technology, 2005.

[81] S. Nejati. Refinement relations on partial specifications. Master's thesis, University of Toronto, 2003.

[82] V. Nicola. Lumping in Markov reward processes. IBM Research Report RC 14719, IBM, 1989.

[83] H.R. Nielson and F. Nielson. *Semantics with Applications: A Formal Introduction*. Wiley, 1992.

[84] D. Park. Concurrency and automata on infinite sequences. In *Proceedings of the 5th GI-Conference on Theoretical Computer Science*, volume 104 of *LNCS*, pages 167–183, London, UK, 1981. Springer-Verlag.

[85] D. Peled. Partial order reduction: linear and branching temporal logics and process algebras. In *Proceedings of the DIMACS workshop on Partial order methods in verification POMIV '96*, pages 233–257, New York, NY, USA, 1997. AMS Press, Inc.

[86] B. Plateau and K. Atif. Stochastic automata network of modeling parallel systems. *IEEE Transactions on Software Engineering*, 17(10):1093–1108, 1991.

[87] G.D. Plotkin. A Structural Approach to Operational Semantics. Technical Report DAIMI FN-19, University of Aarhus, 1981.

[88] A. Pnueli. The temporal logic of programs. In *19th Annual Symposium on Foundations of Computer Science (Providence, R.I.)*, pages 46–57, Ney York, 1977. IEEE.

[89] D. Sangiorgi. On the bisimulation proof method. *Mathematical Structures in Computer Science*, 8(5):447–479, 1998.

[90] R.R.H. Schiffelers and K.L. Man. *Formal Specification and Analysis of Hybrid Systems*. PhD thesis, Eindhoven University of Technology, 2006.

[91] G. Schmidt and T. Ströhlein. *Relations and Graphs - Discrete Mathematics for Computer Scientists*. EATCS Monographs on Theoretical Computer Science. Springer, 1993.

[92] R. Segala and N. Lynch. Probabilistic simulations for probabilistic processes. *Nordic Journal of Computing*, 2(2):250–273, 1995.

[93] J. Sproston and S. Donatelli. Backward stochastic bisimulation in CSL model checking. In *QEST '04: Proceedings of the The Quantitative Evaluation of Systems, First International Conference on (QEST'04)*, pages 220–229, Washington, DC, USA, 2004. IEEE Computer Society.

[94] C. Stirling. *Modal and Temporal Properties of Processes*. Springer, 2001.

[95] N. Trčka. Verifying χ models of industrial systems in Spin. In *8th International Conference on Formal Engineering Methods, ICFEM 2006*, volume 4260 of *LNCS*, pages 132–148. Springer, 2006.

[96] Y.S. Usenko. *Linearization in μCRL*. PhD thesis, Eindhoven University of Technology, 2002.

[97] J.J. Vereijken. *Discrete-time process algebra*. PhD thesis, Eindhoven University of Technology, 1997.

[98] J.L.M. Vrancken. The algebra of communicating processes with empty process. *Theoretical Computer Science*, 177(2):287–328, 1997.

[99] A. Wijs. From χ_t to μCRL: Combining performance and functional analysis. In *ICECCS '05: Proceedings of the 10th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'05)*, pages 184–193, Washington, DC, USA, 2005. IEEE Computer Society.

[100] V. Wolf, M. Majster-Cederbaum, and C. Baier. Trace machines for observing continous time Markov chains. In *ENTCS - QAPL*, 2005(2006).

[101] S.-H. Wu, S.A. Smolka, and E. Stark. Composition and behaviors of probabilistic I/O automata. *Theoretical Computer Science*, 176(1–2):1–38, 1997.

Index

Titles in the IPA Dissertation Series since 2002

M.C. van Wezel. *Neural Networks for Intelligent Data Analysis: theoretical and experimental aspects*. Faculty of Mathematics and Natural Sciences, UL. 2002-01

V. Bos and J.J.T. Kleijn. *Formal Specification and Analysis of Industrial Systems*. Faculty of Mathematics and Computer Science and Faculty of Mechanical Engineering, TU/e. 2002-02

T. Kuipers. *Techniques for Understanding Legacy Software Systems*. Faculty of Natural Sciences, Mathematics and Computer Science, UvA. 2002-03

S.P. Luttik. *Choice Quantification in Process Algebra*. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2002-04

R.J. Willemen. *School Timetable Construction: Algorithms and Complexity*. Faculty of Mathematics and Computer Science, TU/e. 2002-05

M.I.A. Stoelinga. *Alea Jacta Est: Verification of Probabilistic, Real-time and Parametric Systems*. Faculty of Science, Mathematics and Computer Science, KUN. 2002-06

N. van Vugt. *Models of Molecular Computing*. Faculty of Mathematics and Natural Sciences, UL. 2002-07

A. Fehnker. *Citius, Vilius, Melius: Guiding and Cost-Optimality in Model Checking of Timed and Hybrid Systems*. Faculty of Science, Mathematics and Computer Science, KUN. 2002-08

R. van Stee. *On-line Scheduling and Bin Packing*. Faculty of Mathematics and Natural Sciences, UL. 2002-09

D. Tauritz. *Adaptive Information Filtering: Concepts and Algorithms*. Faculty of Mathematics and Natural Sciences, UL. 2002-10

M.B. van der Zwaag. *Models and Logics for Process Algebra*. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2002-11

J.I. den Hartog. *Probabilistic Extensions of Semantical Models*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2002-12

L. Moonen. *Exploring Software Systems*. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2002-13

J.I. van Hemert. *Applying Evolutionary Computation to Constraint Satisfaction and Data Mining*. Faculty of Mathematics and Natural Sciences, UL. 2002-14

S. Andova. *Probabilistic Process Algebra*. Faculty of Mathematics and Computer Science, TU/e. 2002-15

Y.S. Usenko. *Linearization in μCRL*. Faculty of Mathematics and Computer Science, TU/e. 2002-16

J.J.D. Aerts. *Random Redundant Storage for Video on Demand*. Faculty of Mathematics and Computer Science, TU/e. 2003-01

M. de Jonge. *To Reuse or To Be Reused: Techniques for component composition and construction*. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2003-02

J.M.W. Visser. *Generic Traversal over Typed Source Code Representations*. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2003-03

S.M. Bohte. *Spiking Neural Networks*. Faculty of Mathematics and Natural Sciences, UL. 2003-04

T.A.C. Willemse. *Semantics and Verification in Process Algebras with Data and Timing*. Faculty of Mathematics and Computer Science, TU/e. 2003-05

S.V. Nedea. *Analysis and Simulations of Catalytic Reactions*. Faculty of Mathematics and Computer Science, TU/e. 2003-06

M.E.M. Lijding. *Real-time Scheduling of Tertiary Storage*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2003-07

H.P. Benz. *Casual Multimedia Process Annotation – CoMPAs*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2003-08

D. Distefano. *On Modelchecking the Dynamics of Object-based Software: a Foundational Approach*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2003-09

M.H. ter Beek. *Team Automata – A Formal Approach to the Modeling of Collaboration Between System Components*. Faculty of Mathematics and Natural Sciences, UL. 2003-10

D.J.P. Leijen. *The λ Abroad – A Functional Approach to Software Components*. Faculty of Mathematics and Computer Science, UU. 2003-11

W.P.A.J. Michiels. *Performance Ratios for the Differencing Method*. Faculty of Mathematics and Computer Science, TU/e. 2004-01

G.I. Jojgov. *Incomplete Proofs and Terms and Their Use in Interactive Theorem Proving*. Faculty of Mathematics and Computer Science, TU/e. 2004-02

P. Frisco. *Theory of Molecular Computing – Splicing and Membrane systems*. Faculty of Mathematics and Natural Sciences, UL. 2004-03

S. Maneth. *Models of Tree Translation*. Faculty of Mathematics and Natural Sciences, UL. 2004-04

Y. Qian. *Data Synchronization and Browsing for Home Environments*. Faculty of Mathematics and Computer Science and Faculty of Industrial Design, TU/e. 2004-05

F. Bartels. *On Generalised Coinduction and Probabilistic Specification Formats*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2004-06

L. Cruz-Filipe. *Constructive Real Analysis: a Type-Theoretical Formalization and Applications*. Faculty of Science, Mathematics and Computer Science, KUN. 2004-07

E.H. Gerding. *Autonomous Agents in Bargaining Games: An Evolutionary Investigation of Fundamentals, Strategies, and Business Applications*. Faculty of Technology Management, TU/e. 2004-08

N. Goga. *Control and Selection Techniques for the Automated Testing of Reactive Systems*. Faculty of Mathematics and Computer Science, TU/e. 2004-09

M. Niqui. *Formalising Exact Arithmetic: Representations, Algorithms and Proofs*. Faculty of Science, Mathematics and Computer Science, RU. 2004-10

A. Löh. *Exploring Generic Haskell*. Faculty of Mathematics and Computer Science, UU. 2004-11

I.C.M. Flinsenberg. *Route Planning Algorithms for Car Navigation*. Faculty of Mathematics and Computer Science, TU/e. 2004-12

R.J. Bril. *Real-time Scheduling for Media Processing Using Conditionally Guaranteed Budgets*. Faculty of Mathematics and Computer Science, TU/e. 2004-13

J. Pang. *Formal Verification of Distributed Systems*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2004-14

F. Alkemade. *Evolutionary Agent-Based Economics*. Faculty of Technology Management, TU/e. 2004-15

E.O. Dijk. *Indoor Ultrasonic Position Estimation Using a Single Base Station*.

Faculty of Mathematics and Computer Science, TU/e. 2004-16

S.M. Orzan. *On Distributed Verification and Verified Distribution*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2004-17

M.M. Schrage. *Proxima - A Presentation-oriented Editor for Structured Documents*. Faculty of Mathematics and Computer Science, UU. 2004-18

E. Eskenazi and A. Fyukov. *Quantitative Prediction of Quality Attributes for Component-Based Software Architectures*. Faculty of Mathematics and Computer Science, TU/e. 2004-19

P.J.L. Cuijpers. *Hybrid Process Algebra*. Faculty of Mathematics and Computer Science, TU/e. 2004-20

N.J.M. van den Nieuwelaar. *Supervisory Machine Control by Predictive-Reactive Scheduling*. Faculty of Mechanical Engineering, TU/e. 2004-21

E. Ábrahám. *An Assertional Proof System for Multithreaded Java -Theory and Tool Support-* . Faculty of Mathematics and Natural Sciences, UL. 2005-01

R. Ruimerman. *Modeling and Remodeling in Bone Tissue*. Faculty of Biomedical Engineering, TU/e. 2005-02

C.N. Chong. *Experiments in Rights Control - Expression and Enforcement*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2005-03

H. Gao. *Design and Verification of Lock-free Parallel Algorithms*. Faculty of Mathematics and Computing Sciences, RUG. 2005-04

H.M.A. van Beek. *Specification and Analysis of Internet Applications*. Faculty of Mathematics and Computer Science, TU/e. 2005-05

M.T. Ionita. *Scenario-Based System Architecting - A Systematic Approach to Developing Future-Proof System Architectures*. Faculty of Mathematics and Computing Sciences, TU/e. 2005-06

G. Lenzini. *Integration of Analysis Techniques in Security and Fault-Tolerance*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2005-07

I. Kurtev. *Adaptability of Model Transformations*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2005-08

T. Wolle. *Computational Aspects of Treewidth - Lower Bounds and Network Reliability*. Faculty of Science, UU. 2005-09

O. Tveretina. *Decision Procedures for Equality Logic with Uninterpreted Functions*. Faculty of Mathematics and Computer Science, TU/e. 2005-10

A.M.L. Liekens. *Evolution of Finite Populations in Dynamic Environments*. Faculty of Biomedical Engineering, TU/e. 2005-11

J. Eggermont. *Data Mining using Genetic Programming: Classification and Symbolic Regression*. Faculty of Mathematics and Natural Sciences, UL. 2005-12

B.J. Heeren. *Top Quality Type Error Messages*. Faculty of Science, UU. 2005-13

G.F. Frehse. *Compositional Verification of Hybrid Systems using Simulation Relations*. Faculty of Science, Mathematics and Computer Science, RU. 2005-14

M.R. Mousavi. *Structuring Structural Operational Semantics*. Faculty of Mathematics and Computer Science, TU/e. 2005-15

A. Sokolova. *Coalgebraic Analysis of Probabilistic Systems*. Faculty of Mathematics and Computer Science, TU/e. 2005-16

T. Gelsema. *Effective Models for the Structure of pi-Calculus Processes with*

Replication. Faculty of Mathematics and Natural Sciences, UL. 2005-17

P. Zoeteweij. *Composing Constraint Solvers*. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2005-18

J.J. Vinju. *Analysis and Transformation of Source Code by Parsing and Rewriting*. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2005-19

M.Valero Espada. *Modal Abstraction and Replication of Processes with Data*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2005-20

A. Dijkstra. *Stepping through Haskell*. Faculty of Science, UU. 2005-21

Y.W. Law. *Key management and link-layer security of wireless sensor networks: energy-efficient attack and defense*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2005-22

E. Dolstra. *The Purely Functional Software Deployment Model*. Faculty of Science, UU. 2006-01

R.J. Corin. *Analysis Models for Security Protocols*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2006-02

P.R.A. Verbaan. *The Computational Complexity of Evolving Systems*. Faculty of Science, UU. 2006-03

K.L. Man and R.R.H. Schiffelers. *Formal Specification and Analysis of Hybrid Systems*. Faculty of Mathematics and Computer Science and Faculty of Mechanical Engineering, TU/e. 2006-04

M. Kyas. *Verifying OCL Specifications of UML Models: Tool Support and Compositionality*. Faculty of Mathematics and Natural Sciences, UL. 2006-05

M. Hendriks. *Model Checking Timed Automata - Techniques and Applications*. Faculty of Science, Mathematics and Computer Science, RU. 2006-06

J. Ketema. *Böhm-Like Trees for Rewriting*. Faculty of Sciences, VUA. 2006-07

C.-B. Breunesse. *On JML: topics in tool-assisted verification of JML programs*. Faculty of Science, Mathematics and Computer Science, RU. 2006-08

B. Markvoort. *Towards Hybrid Molecular Simulations*. Faculty of Biomedical Engineering, TU/e. 2006-09

S.G.R. Nijssen. *Mining Structured Data*. Faculty of Mathematics and Natural Sciences, UL. 2006-10

G. Russello. *Separation and Adaptation of Concerns in a Shared Data Space*. Faculty of Mathematics and Computer Science, TU/e. 2006-11

L. Cheung. *Reconciling Nondeterministic and Probabilistic Choices*. Faculty of Science, Mathematics and Computer Science, RU. 2006-12

B. Badban. *Verification techniques for Extensions of Equality Logic*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2006-13

A.J. Mooij. *Constructive formal methods and protocol standardization*. Faculty of Mathematics and Computer Science, TU/e. 2006-14

T. Krilavicius. *Hybrid Techniques for Hybrid Systems*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2006-15

M.E. Warnier. *Language Based Security for Java and JML*. Faculty of Science, Mathematics and Computer Science, RU. 2006-16

V. Sundramoorthy. *At Home In Service Discovery*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2006-17

B. Gebremichael. *Expressivity of Timed Automata Models*. Faculty of Science, Mathematics and Computer Science, RU. 2006-18

L.C.M. van Gool. *Formalising Interface Specifications*. Faculty of Mathematics and Computer Science, TU/e. 2006-19

C.J.F. Cremers. *Scyther - Semantics and Verification of Security Protocols*. Faculty of Mathematics and Computer Science, TU/e. 2006-20

J.V. Guillen Scholten. *Mobile Channels for Exogenous Coordination of Distributed Systems: Semantics, Implementation and Composition*. Faculty of Mathematics and Natural Sciences, UL. 2006-21

H.A. de Jong. *Flexible Heterogeneous Software Systems*. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2007-01

N.K. Kavaldjiev. *A run-time reconfigurable Network-on-Chip for streaming DSP applications*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2007-02

M. van Veelen. *Considerations on Modeling for Early Detection of Abnormalities in Locally Autonomous Distributed Systems*. Faculty of Mathematics and Computing Sciences, RUG. 2007-03

T.D. Vu. *Semantics and Applications of Process and Program Algebra*. Faculty of Natural Sciences, Mathematics, and Computer Science, UvA. 2007-04

L. Brandán Briones. *Theories for Model-based Testing: Real-time and Coverage*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2007-05

I. Loeb. *Natural Deduction: Sharing by Presentation*. Faculty of Science, Mathematics and Computer Science, RU. 2007-06

M.W.A. Streppel. *Multifunctional Geometric Data Structures*. Faculty of Mathematics and Computer Science, TU/e. 2007-07

N. Trčka. *Silent Steps in Transition Systems and Markov Chains*. Faculty of Mathematics and Computer Science, TU/e. 2007-08

Curriculum Vitae

Nikola Trčka was born on the 5th of October 1977 in Belgrade, Serbia (then Yugoslavia). He studied computer science at the Faculty of Mathematics, University of Belgrade, and obtained the degree *Graduated Mathematician for Computer Science* (equivalent to M.Sc.) in 2003. In July 2003 he became a Ph.D student at the Formal Methods Group, Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands.