Control Structures

Alex Mifsud

Doctor of Philosophy

University of Edinburgh

1996

Abstract

Action structures have been proposed as an algebraic framework for models of concurrent behaviour. In this thesis, refinements of action structures are developed, providing an abstract treatment of the structural aspect of processes, as well as a setting in which to study their dynamics.

Concrete models of concurrent computation such as Petri nets and the π-calculus have been cast as action structures in a uniform manner, giving rise to a concrete class of action structures, called *action calculi*. As a result, action calculi are here adopted as the point of departure towards an abstract algebraic treatment of process construction and concurrent computation. The refinement of action structures to *control structures* gives a semantic space for action calculi; and includes a semantic account of names, based around a semantic counterpart to the syntactic notion of free names called *surface*.

Two variants of action calculi are explored in analogous fashion. Present in these variants are some intuitively appealing aspects, such as greater expressivity of dataflow; a semantic treatment of name hiding or restriction; and, in one of the variants, garbage collection of restricted but unused names and a characterisation of surface in terms of restriction.

While the treatment of process constructors reveals rich structural issues, the algebraic framework given by control structures provides considerable support for studying the dynamical aspects of processes. In particular, it allows a comparison of diverse action calculi upon their dynamic properties; illustrated here is a method of achieving this. The method involves an examination of action calculi dynamics through the images of the calculi on a common *static* model called a *classifier*.

Finally, as a step towards establishing formal connections with mainstream process algebra, an operational semantics for PIC^r, the π-calculus cast in the framework, is developed. Labelled transition relations on the terms of PIC^r are defined,

leading to the formulation of operational models through the familiar technique of bisimulation.

Acknowledgements

I would like to thank my supervisors: Stuart Anderson, for nurturing within me a spirit of scientific inquisitiveness; Robin Milner, for providing both inspiration and a sense of direction that stayed my meandering spirit from losing focus; and finally, John Power, for his unrelenting insistence that inquisitiveness be accompanied with a commensurate degree of rigour. All three have greatly contributed to my understanding of computation and, in no small way, to my general intellectual development throughout the time I spent in their company.

My studies have been funded by the University of Malta to which I am indebted especially for its support during the year I spent in Malta writing this thesis. In particular, I acknowledge the help and encouragement extended to me by Juanito Camilleri, my head at the Department of Computer Science and Artificial Intelligence, who by unburdening me from teaching duties, enabled me to dedicate that year almost entirely to the composition of this thesis.

I am grateful for the support of my wife Stephanie and our parents for shielding me, throughout the past three years, from the responsibilities of parenting a lively child who started life just when I embarked on this work.

Declaration

I declare that this thesis was composed by myself, and that the work it presents is my own with the exception of chapter 2, which contains joint work with Robin Milner and John Power; chapter 3, which presents work done with Masahito Hasegawa; and where explicitly stated.

Alex Mifsud

Table of Contents

Chapter 1

Introduction

In the study of concurrency one encounters two distinct but complementary notions: *independence* and *interaction.* Independent entities, which we shall refer to as *processes* may interact and moreover, interaction may result in dependencies—or links—being established between such processes. Processes which are independent and yet cannot influence one anothers' behaviour are hardly interesting: it is debatable whether such silent or non-interacting processes are even observable.

One of the aims of developing a theory of concurrency is to support engineering reasoning for the construction and analysis of systems composed of concurrent parts. This imposes two broad concerns on our enquiry: one to do with structure, specifically relating to how entities may be put together; and the other to do with behaviour, which allows us to tell when two such entities may be considered equivalent or interchangeable without effect on the system they might form part of. In this setting, the abovementioned concerns with linkage (dependency) and interaction are manifest as the interplay between structure (statics) and behaviour (dynamics).

Many existing models for concurrency address both these concerns in either of two ways. Process-algebraic models start by identifying process constructors (structure) and then go on to assign behaviours to the processes built from them. An alternative, sometimes called *denotational*, approach starts by proposing structures for modelling behaviour and then provides constructions on these structures which correspond to the process-algebraic constructs.

Behavioural models are often based on an abstract notion of interaction or its observation: in such models these notions are assumed as given *a priori*. Moreover, most models of this kind capture a specific type of relationship between distinct *events*, as we shall refer to both interaction and its observation; the relationship is usually expressed in terms of mathematical structure imposed on the events. Examples of such structures are *traces* and *synchronisation trees* which respectively reflect the linear and branching ordering of (potential) events. Some structures such as *asynchronous transition systems* and *event structures* account also for the causal relationships between events. The notion of independence, concurrency or parallelism is typically presented as a property of the structures employed to describe such causal links between events.

Taking the behavioural approach, Nielsen, Winskel and others [32,33] have classified some of the existing models by casting them in a category theoretic setting where the relationships between the models are expressed in terms of reflections and coreflections: adjunctions which represent the embeddings between models. Their classification is motivated by three independent parameters: abstraction from the causal independence of events by a nondeterministic interleaving; abstraction from the looping structure by unfolding; and abstraction from the branching structure by regarding a process as a collection of event sequences corresponding to paths in the computation tree (traces). Their work also addresses the issue of process constructs through categorical constructions on the behavioural structures. Indeed, an important part of their work is in establishing connection with process algebra, not only by recovering the constructions, but also in giving an account of the ubiquitous operational-semantic device of bisimulation [13].

Even within a narrow behavioural view, the degree of choice (of process model) is large. In [7] van Glabbeek provides an extensive comparison between the various equivalences which abound in what he calls the *linear time-branching time spectrum* bounded by bisimulation on transition systems at one end and trace equivalence at the other. The models considered differ in their choice of what should be taken as an observable interaction, in the structures built from the observations and finally in the equivalences on the structures used to obtain more abstract models.

The traditional approach employed in process-algebraic models has been to describe (a fixed set of) process constructors as a term algebra. The behaviour of a process would then be obtained by defining labelled transition relations between terms which reflect the ability for interaction of the process described by the term. The labelled transition relations are then employed to generate, depending on the behavioural structure favoured, transition systems; transition trees; or transition paths for each term. These structures would then be factored by equivalences based on the labelled transitions which constitute them, giving a behavioural justification to the semantics. Some of the equivalences will be congruences and the identifications made induce equations on the terms giving a term algebra. One possible advantage of this approach over the behavioural one is that no commitment to a particular notion of behaviour (or its observation) is made a priori. Indeed, by considering different equivalences, the interpretation of process terms can be effectively varied; even the notion of interaction can be modified by changing the labelled transition relations for a given set of process constructors. Examples of the algebraic approach includes process algebras such as CCS [31], CSP [9], the box calculus [4] for Petri Nets [34] and the π-calculus [30,22].

Process calculi take process constructions as their starting point and include explicit accounts of the dynamic interactions of processes. However, the variety of process algebras indicate the absence of a canonical algebraic structure for concurrency. One interesting approach to dealing with this diversity is provided by Berry and Boudol's Chemical Abstract Machine (CHAM) for "implementing" process calculi [3]. Based on multisets—following the ideas of Banâtre and Metayer [2]—the CHAM suggests common underpinnings for the various process calculi representable as CHAMs and also provides a basis for comparison. Indeed, the CHAM was to prove an important source of inspiration for the concrete structures in which existing process calculi are cast in this thesis [21,24].

Action structures have been proposed by Milner [21] as a general framework in which concrete models of concurrency and interaction may be studied. These structures are essentially strict monoidal categories[1] [16] with added structure including *reaction*, a local preorder on the arrows. The arrows of an action structure are called *actions* and represent processes, while reaction represents their dynamics. This algebraic framework does not make any commitment to a specific level of abstraction and simply provides a setting in which to cast and study combinators which express process constructions: the axioms of an action structure constrain but do not determine the interpretation of the operations. Furthermore, a class of syntactic action structures has been developed [24], called *action calculi*; providing machinery for dealing—syntactically—with name binding and substitution. In addition to these operations, an action calculus is obtained by the inclusion of a set of combinators, called *controls*, and an associated set of rules describing their dynamic behaviour. These combinators are sufficiently powerful to enable processes to be represented as complex actions. Milner has shown that existing models such as Petri Nets and the π-calculus fit readily in this framework [24], indicating that the expressiveness provided by existing models is not limited by this reduction of entities.

To return to our initial remark, we shall now cast the notions of independence and interaction in terms of action structures. Processes (here called *actions*) are represented by the arrows of an action structure: tensor product embodies the operation of parallel composition or, in behavioural terms, independence. Composition signifies a form of data dependency: $a \cdot b$ indicates that the information produced by a, say, as a result of computation, is fed into b. The idea of dataflow may be hard to intuit in the context of process algebra. In most process algebras, processes exchange data through synchronisation and not through static links of input and output as in functional paradigms. Such processes can be thought to be special cases where such input and output "dataflow channels" are absent. The presence of dataflow channels provides an interesting form of dependency; in $a \cdot b$

[1] The use of monoidal categories to model concurrency has at least one precursor in Meseguer and Montanari's modelling of Petri Nets as a monoidal category [19].

the process b "depends" on a in the sense that information passed by a to b may influence the behaviour of b.

Interaction, or computation (we shall not distinguish the two), is represented by the reaction preorder $\searrow$ with $a \searrow b$ meaning that a can get to b as a result of computation. The correspondence between independence and interaction may now be phrased as follows: computation may produce changes in the dataflow topology of a process; and, in turn, the presence of dataflow channels between processes may, by the information flowing through them, affect the computational behaviour.

Other approaches towards establishing a general framework for concurrency include Meseguer's conditional rewriting logic [18], whose models he calls *$\mathcal{R}$-systems.* In an $\mathcal{R}$-system algebra, the carrier consists of the computations of an individual process, whereas in action structures, the processes (actions) themselves constitute the carrier. An alternative approach with similar motivation as for action structures is Abramsky's *interaction categories* [1] which provide an expressive type structure that controls the construction or linking-together of processes. One difference between interaction categories and action structures is that in the former the use of names to express such linkage (as employed in action calculi) is eschewed. Another is that the treatment of dynamics in action calculi is more explicit through the employment of controls and reaction rules. The differences apparent among the various models are indeed striking; yet, if a canonical abstract semantic model for concurrency is to be found, the common elements underlying the structure of processes and their dynamic behaviour must be identified. It is their aspect, not the elements, that is distinct in each of the models mentioned.

1.1 Objectives and Outline

The task of eliciting common *abstract* structure in process-algebraic models for concurrency is assisted by the ability to cast various existing models within a common framework. This is just what the notion of an action structure provides;

indeed, as mentioned above, a concrete kind of action structure—action calculi—exists which allows such models to be represented. The availability of a common, albeit concrete, structural basis leads to a natural strategy for extracting the underlying abstract structure; the strategy is to look for the additional *abstract* structure present in (all instances of) action calculi which is not provided for by action structures. Technically, this is achieved in this thesis by a refinement of action structures, which we call *control structures*, amongst which action calculi occupy a special place as the *initial* such. Two remarks are in order at this point; the first concerns the qualification of action calculi as the right kind of structure in which to cast concrete models of concurrency: can one do with less structure, or, indeed, does one need even more? That commonly used models fit the mould is evidence only of being on the right track. The second remark concerns the refinement of action structures which will provide an abstract semantic space of interpretation for action calculi: there may be many such refinements which give the required result, namely the initiality of action calculi. The choice must therefore be justified by additional factors.

The starting point of this thesis—that which is justified solely by example and intuition—is a syntactic form for representing concrete models of concurrent computation: the molecular form presentation of action calculi. Milner claims that Berry and Boudol's CHAM provided an inspiration for the molecular forms; and that the resulting action calculi provide a kind of algebraic version of it. A contemplation of their similarity highlights also their differences, and also suggests possible variations. To simplify considerably, the molecular form provides an enhanced kind of CHAM with dataflow between molecules. We shall see that, in the first kind of molecular form presented in this thesis, this dataflow is constrained in a particular fashion. This will lead us to present a variation of the molecular forms where the constraint is eased. Such consideration of alternatives is partly in response to our concern with the qualification of the molecular forms as the right concrete common basis for representing processes.

As suggested by the above, the main result obtained about action calculi consists in providing an appropriate refinement of action structures, of which action calculi are the initial instances. In the search for the right abstract structure, the problem of giving a *semantic* account of names, and their attendant syntactic notions such as freeness, binding and substitution, turned out to be one of the most challenging aspects. In many process calculi, names have a crucial role in specifying interaction and have a greater role than that of simple "place-holders" as do variables in, say, the λ-calculus. Thus, names—or as John Power insists [8], naming—have a semantic presence beyond that of simple indeterminates. While in many process calculi, the names of channels are kept distinct from the names employed as place-holders or variables, this is not universally the case: in the π-calculus, names assume both roles, giving the calculus the means to express mobility of channels. Therefore, it should not be surprising that in developing our model we were compelled to deal with the issue of naming.

Our abstract semantic treatment of action calculi focuses predominantly on their static structure. We recall that action calculi are determined by the controls and their reaction rules; the controls are responsible for providing additional static constructions. The computational behaviour of an action calculus is specified syntactically by a set of reaction rules which, by some closure conditions, determine the reaction relation. In the definition of action calculi, only lax constraints have been imposed on the forms that such rules can assume (for instance, that both sides of a reaction must have the same arities). Therefore, one way to explore the dynamics is by means of a classification of reaction relations based on some syntactic criteria on the reaction rules which induce such relations.

An alternative means for exploring dynamics is provided by a device we shall call a *classifier*. A classifier is a (concretely or abstractly specified) model of *static* action calculi—that is, one which does not necessarily preserve the dynamics—which arises uniformly from any set of controls. This allows a comparison of action calculi to be made by considering their image onto a common model. For a comparison of the dynamics, we shall consider (homomorphic) maps from action calculi to their models which preserve the reaction relation; accordingly, associated with the classifier will be a (fixed) reaction relation, which somehow embodies the property of dynamics in question. Thus, since static structure alone ensures the existence of a map to the classifier, the existence of a map *which preserves the reaction relation* will depend on the reaction relation of the action calculus. In this way the classifier distinguishes between those action calculi which have such a map to those which do not. This approach is useful when the existence of such a map can be related to some interesting property of the dynamics, such as mobility. In the second part of this thesis, we will explore two related examples of such classifiers.

One of the results obtained for control structures, and their reflexive variants, is closure under quotient by congruence. This allows us to obtain computationally meaningful models (control structures) through an operational semantics. For process algebras such models have traditionally been obtained through bisimilarity on labelled transition relations between process terms. This technique will be applied to a leading example: the π-calculus cast in our framework.

In summary, this thesis will include a general treatment of the static, or data-flow, aspect of processes and a foray, largely by way of concrete example, into the issues concerning dynamics. It is loosely organised in three parts. The first explores mainly the static structure of processes, with special emphasis on the nature of static dependencies and their expression through naming. The rest of the thesis will be concerned with providing examples and applications of the semantic framework established. In particular, we illustrate the potential of the framework for providing a means of comparing diverse action calculi upon their dynamic properties. Another example deals with an operational semantics of the π-calculus cast in our framework. The notion of a labelled transition is developed for this example with the purpose of eliciting the underlying semantic ideas embodied by labelled transition relations. The presence of labelled transitions permits comparison with the traditional presentation of the π-calculus and provide a basis for obtaining operational models through bisimulation.

Outline by chapter

Below is a brief outline of each chapter.

Chapter 2: Control structures Action structures are reviewed as an algebraic variety underlying models of concurrency. Action calculi, a syntactic class of action structures parameterised over a set of *control constructions*, are then introduced. Action calculi, each determined by sets of such controls, are presented in two ways: through syntactic constructions called molecular forms; and as a term algebra factored by a congruence arising from a set of equations (the theory AC). Central to the semantic treatment of action calculi is the notion of *surface*, which provides a semantic counterpart to the syntactic concept of free names. Inspired by the definition of surface, we formulate an elegant refinement of action structures which yields a class (actually, a category) of models for action calculi. The category of control structures is shown to be closed under quotient by congruence.

Chapter 3: Reflexive control structures The *reflexion* operation, which corresponds to a form of feedback in a dataflow interpretation, is introduced by means of a set of equations (giving, together with AC, the theory AC^{r}) constraining its interaction with the operations of a control structure. By way of illustration, we show how reflexion—in the presence of higher order controls—provides a form of recursion. The inclusion of reflexion leads to a variation of action calculi which will be called *reflexive action calculi*. Similarly *reflexive control structures* are defined as a corresponding refinement of control structures in which reflexion is manifest as a trace on a strict monoidal category. Analogously to chapter 2, the main result holds that the reflexive action calculus determined by a given set of controls is initial in the category of reflexive control structures over that set of controls. The imposition of an additional equation governing reflexion is also considered, resulting in a form of garbage collection in the resultant (reflexive) molecular forms; it also allows an alternative characterisation of surface.

Chapter 4: Skeleta Two kinds of reflexive control structures are explored in terms of both their static and dynamic properties. Skeleta are syntactic reflexive control structures in which some of the structure of the controls is forgotten. This allows them to be uniformly defined for arbitrary sets of controls and this fact makes them useful in comparing and classifying reaction rules, and thereby, action calculi. For each of the skeleta under consideration, we shall define a natural notion of dynamics. This will be used to determine certain dynamic properties of those action calculi for which a structure-preserving map (homomorphism) to the skeleta exists.

Chapter 5: The reflexive π-calculus In this chapter we establish the setting for an exploration of the dynamics of an important example of reflexive control structures: the reflexive π-calculus PIC^r. Derivation rules for labelled transitions on the terms of PIC^r are presented and shown to derive identical transitions from equal terms. This allows us to establish a meaningful correspondence between transitions on terms and computations on the molecular forms, thereby justifying our use of the labelled transition relations as a basis for an operational semantics.

Chapter 6: Bisimilarities Strong bisimilarity is defined in the expected way on the labelled transitions. This bisimilarity is shown to be too strong as it does not identify enough actions which are deemed behaviourally indistinguishable. A technique for obtaining weaker forms of bisimilarity is then presented. This technique consists essentially of specifying the set of labelled transitions upon which the bisimilarity will be based. Sufficient conditions are given for the congruentiality of the bisimilarities obtained in this way. A limitation of the technique is also identified and a rectification is proposed through the introduction of a further rule for obtaining labelled transitions.

Chapter 2

Control Structures

Concrete models of concurrency such as Petri Nets and the π-calculus, may be cast as action structures in a uniform way, as instances of a syntactic class of action structures called *action calculi*. Two presentations of action calculi exist [24]: a direct construction of the syntactic objects called *molecular forms*, and the quotient of a term algebra whose constructors include the operations of action structures. These two presentations have been shown isomorphic in [23].

Each action calculus $\mathsf{AC}(\mathcal{K})$ is determined essentially by a set $\mathcal{K}$ of *control* operations called a *signature*; for example, an action calculus for an interesting fragment of the π-calculus is obtained by the controls $\boldsymbol{\nu}$, **out** and **box** (restriction, output and input guarding respectively). $\mathsf{AC}(\mathcal{K})$ may also be equipped with a set of reaction rules $\mathcal{R}$—in which case we write $\mathsf{AC}(\mathcal{K}, \mathcal{R})$—which determine its reaction relation; these rules provide the meaning of the controls in $\mathcal{K}$.

Our aim in this chapter is to find a natural category of action structures in which $\mathsf{AC}(\mathcal{K})$ is initial. In effect, this entails selecting a space of semantic interpretations for $\mathsf{AC}(\mathcal{K})$, which we shall call *control structures over* $\mathcal{K}$. These structures together with the expected notion of homomorphism form a category $\mathbf{CS}(\mathcal{K})$ with $\mathsf{AC}(\mathcal{K})$ initial.

A significant difficulty to be overcome in defining control structures is the treatment of names. The difficulty arises as the axioms of an action calculus are not purely algebraic; they are *axiom schemata* rather than axioms since they contain side conditions which make reference to the free names of action terms. A finite set of pure algebraic axioms which are equipotent (in the term algebra) with the action calculus axioms would guarantee initiality for $\mathsf{AC}(\mathcal{K})$ in the category of structures arising from such axioms. Such a set is not uniquely determined but we have found a set which we believe is satisfying both mathematically and intuitively. This has been achieved by introducing a semantic counterpart to the notion of the free names of an action.

Each action a of $\mathsf{AC}(\mathcal{K})$, for certain cases of $\mathcal{K}$ (for instance, that which gives the π-calculus), represents a process with an external surface through which other processes may communicate with it. This surface is therefore semantically significant, since the potential for communication is expected to be at least partly determined by it; for instance, in the π-calculus, those independent processes (those not connected through dataflow channels) which do not have any free names in common in their respective surfaces will not be able to communicate.

An important property of the category $\mathbf{CS}(\mathcal{K})$ is closure under quotient by an arbitrary congruence. In particular, it will contain any model derived by factoring the action calculus $\mathsf{AC}(\mathcal{K})$ by a congruence; when the congruence has operational significance, as in the case of bisimulation congruence, this accords with the established practice of giving operational semantics to such calculi. Moreover, the surface of each action in the model (an equivalence class) is given exactly by the intersection of the surfaces of all the actions in the equivalence class: thus, those names which are semantically insignificant are discarded in the model.

Outline In Section 2.1 action structures are reviewed followed, in Section 2.2, by a presentation of action calculi in terms of syntactic constructions known as *molecular forms* as well as a quotient of a term algebra by a theory AC. The section ends with a discussion on the axioms of AC. This leads the way to the formulation of control structures via the intermediate step of symmetric action structures which are defined in Section 2.3. In this section we shall also introduce the notion of surface and derive some relevant properties in the context of symmetric action structures. Control structures are defined in Section 2.4; the main results are that

$\mathsf{AC}(\mathcal{K})$ is initial in the category of control structures and that this category is closed under quotient by arbitrary congruence.

2.1 Action Structures

An action structure is a strict monoidal category with additional structure. The arrows of the category are called *actions* and the objects are called *arities*: these objects may be interpreted as types for the input and output of each action. The additional structure is given by endofunctors called abstractors indexed by a set of names. Dynamic action structures are also equipped with a preorder on actions called *reaction* which embodies the computational behaviour or dynamics of the actions. The following definitions give an algebraic description of an action structure.

Definition 2.1 (Static action structure) *Let X be a set of* names *(ranged over by x, y, z) and $(M, \otimes, \epsilon)$ be a monoid of* arities *with an assignment of an arity $m \in M$ to each $x \in X$. Let A be a set of* actions *partitioned by pairs of arities m, n where for each partition $A_{m,n}$, if $a \in A_{m,n}$ we say that a has arity $m \to n$ and write $a : m \to n$. Let A be equipped with*

- *an* identity *operation* $\mathbf{id}_m : m \to m$ *for each arity m;*
- composition $\cdot$ *and* tensor $\otimes$ *operations subject to the rules of arity*

$$\cdot\frac{a_1 : k \to m \quad a_2 : m \to n}{a_1 \cdot a_2 : k \to n} \qquad \otimes\frac{a_1 : m_1 \to n_1 \quad a_2 : m_2 \to n_2}{a_1 \otimes a_2 : m_1 \otimes m_2 \to n_1 \otimes n_2}$$

- *and for each name x, an* abstraction *operation* $\mathbf{ab}_x$ *subject to the arity rule*

$$\mathbf{ab}\frac{a : m \to n}{\mathbf{ab}_x a : k \otimes m \to k \otimes n}\, x : k$$

Then (M, X, A) is a static action structure *over X if the following axioms hold in A:*

$$\begin{aligned}
\mathrm{C}_1 &: \ a \cdot \mathbf{id} = a = \mathbf{id} \cdot a & \mathrm{C}_2 &: \ a \cdot (b \cdot c) = (a \cdot b) \cdot c \\
\mathrm{P}_1 &: \ a \otimes \mathbf{id}_\epsilon = a = \mathbf{id}_\epsilon \otimes a & \mathrm{P}_2 &: \ a \otimes (b \otimes c) = (a \otimes b) \otimes c \\
\mathrm{PF}_1 &: \ \mathbf{id} \otimes \mathbf{id} = \mathbf{id} & \mathrm{PF}_2 &: \ (a \cdot b) \otimes (c \cdot d) = (a \otimes c) \cdot (b \otimes d) \\
\mathrm{AF}_1 &: \ \mathbf{ab}_x \mathbf{id} = \mathbf{id} & \mathrm{AF}_2 &: \ \mathbf{ab}_x(a \cdot b) = (\mathbf{ab}_x a) \cdot (\mathbf{ab}_x b)
\end{aligned}$$

where, in the above equations, arities may be assigned in any way that obeys the rules of arity. ∎

The definition of homomorphism is standard.

Definition 2.2 *Let $\mathcal{A}$ and $\mathcal{B}$ be two static action structures. Then a* homomorphism of static action structures $\Phi : \mathcal{A} \to \mathcal{B}$ *consists of*

- *a monoid homomorphism $\Phi : M_A \to M_B$;*
- *a map $\Phi : X_A \to X_B$ such that $x : m$ implies $\Phi x : \Phi m$;*
- *a map $\Phi : A \to B$ such that*
  - *$a : m \to n$ implies $\Phi a : \Phi m \to \Phi n$;*
  - *Φ preserves* **id**, $\cdot$, $\otimes$ *and* $\mathbf{ab}_x$*;*

If, in addition, Φ is injective, then A is called a static sub-actionstructure of B.

∎

We can motivate the operations by an informal interpretation in terms of dataflow. We think of an action $a : m \to n$ as a black box with input (dataflow) channels of aggregate width m and output channels of width n. Identity is just a simple dataflow channel through which information may pass unobstructed and unchanged [24]. The tensor operation may be interpreted as parallel composition: it is a construction which does not create dataflow dependencies and simply places two actions side by side, thus aggregating both input and output arities. The composition operation on the other hand connects two actions by tying the outputs of one to the inputs of the other; hence the arity rule requiring the output arity of a_1 to match the input arity of a_2 for the composite $a_1 \cdot a_2$ to be well formed.

The inclusion of names in action structures is hard to motivate since there is no interesting role for them given within the abstract structure itself. Informally we have suggested that the abstraction operation $\mathbf{ab}_x$ captures parametrisation by the name x; hence, it may be expected that every "free" occurrence of the name x in a would be "bound" in $\mathbf{ab}_x a$. In dataflow terms this allows the creation of a new dataflow channel connected to each point where x occurs "free". But freeness and binding are concrete notion which assume a concrete or syntactic structure for objects. There is, however, an indirect way to capture the semantic notion of freeness for names by analysing the effect of applying abstraction of a given name upon an action. Although some insights can be derived even at this stage, treatment of this will be deferred until further structure has been introduced, particularly that which allows more to be said about the interaction of abstraction with the tensor operation.

Note on names and arities For this thesis we shall assume that M is freely generated by a set P of *prime* arities (ranged over by $p, q, \ldots$), that the arity of every name is prime and moreover, that there are infinitely many names associated with every prime arity.

Definition 2.3 ((Dynamic) action structure) *Let* (M, X, A) *be a static action structure and let* $\searrow$ *be a preorder on each* $A_{m,n}$ *called* reaction *which is preserved by composition, tensor and abstraction. Also each* **id** *is minimal for* $\searrow$*, i.e. if* $\mathbf{id} \searrow a$ *then* $\mathbf{id} = a$*. Then* $(M, X, A, \searrow)$ *is a* (dynamic) action structure. ■

Definition 2.4 *Let* $\mathcal{A}$ *and* $\mathcal{B}$ *be two action structures. Then a* homomorphism of action structures $\Phi : \mathcal{A} \to \mathcal{B}$ *is a homomorphism of static action structures which preserves the reaction relation i.e. whenever* $a \searrow_A a'$ *then* $\Phi a \searrow_B \Phi a'$.

If, in addition, Φ *is injective, and* $\Phi a \searrow_B \Phi a'$ *implies* $a \searrow_A a'$ *then* A *is called a sub-actionstructure of* B. ■

Discussion For the definition of homomorphism of action structures we might have chosen a stronger condition than the preservation of the reaction relation. For instance we might have required that Φ exactly preserves the relation i.e $a \searrow_A a'$ iff $\Phi a \searrow_B \Phi a'$. This, of course, depends on our intended role for homomorpisms of action structures. We expect that the semantics of concrete models can be expressed as homomorphisms of action structures: it may be that some models collapse computational steps. The double implication form can accomodate such models since every step in the model will have at least a counterpart concrete computational step. However, we also intend homomorphisms to represent encodings of one concrete model into another: in this case one computational step in the source model may be "implemented" through a greater number of steps in the target model (such as in the compilation of a high level language into low level assembly). It is possible in this case, that in the target model there will be intermediate states which have no counterpart in the source and hence the translation or encoding would not fit the double implication form of homomorphism. At this point, therefore, we shall keep the condition fairly weak but we expect that certain applications will suggest stronger conditions.

2.2 Action Calculi

We shall address the problem of providing notions of free name, binding and substitution first in a rather concrete setting given by a syntactic class of action structures called action calculi. These concrete action structures will in turn lead us to a refinement of action structures that deals semantically with names in a more satisfactory manner. Before presenting the technical details, we illustrate the ideas by an example derived from the π-calculus. Consider the term

$$P = (\nu u)(\overline{u}y \mid u(z).Q)$$

where the subterm $\overline{u}y$ represents a message y to be transferred along the channel u, causing any (free) occurrence of z in Q to be replaced by y. (The restriction

$(\boldsymbol{\nu} u)$ ensures that the message can be received nowhere else.) Formally, this is represented by the reduction:

$$P \longrightarrow P' = (\boldsymbol{\nu} u)(\{y/z\}Q)$$

In the molecular form presentation of action calculi, actions are built from molecules, each of which arises from some control in $\mathcal{K}$. This form is in the spirit of the Chemical Abstract Machine (CHAM) of Berry and Boudol [3]. For P above, the molecular form of the corresponding action $\widehat{P}$ contains three molecules and is written

$$\widehat{P} = [\boldsymbol{\nu}(u), \langle uy \rangle \mathbf{out}, \langle u \rangle \mathbf{box}((z)\widehat{Q})]$$

where $\widehat{Q}$ is the molecular form for Q. The difference from the CHAM is that molecules may *bind* one another; in this case, the molecule $\boldsymbol{\nu}(u)$ binds the other two molecules through the name u. Note that the **box** control encapsulates an inner molecular form. In the dynamics of molecular forms, redexes consist of certain patterns of molecules; in this case the last two molecules form a redex, and the following reduction occurs (releasing $\widehat{Q}$):

$$\widehat{P} \searrow \widehat{P'} = [\boldsymbol{\nu}(u), \{y/z\}\widehat{Q}]$$

In the term algebra presentation of the action calculus, writing $\overline{P}$ as the term arising from P, we can recast the above example as follows:

$$\begin{array}{lcl} \overline{P} & = & \boldsymbol{\nu} \cdot (u)(\langle uy \rangle \cdot \mathbf{out} \otimes \langle u \rangle \cdot \mathbf{box}((z)\overline{Q})) \\ \overline{P} & \searrow & \overline{P'} = \boldsymbol{\nu} \cdot (u)(\langle y \rangle \cdot (z)\overline{Q}) \end{array}$$

Note that the tensor product $\otimes$ of action calculi represents parallel composition; also that composition $\cdot$ and abstraction (u)—a derived form of $\mathbf{ab}_u$—represent both kinds of binding (restriction and input) in the π-calculus.

Molecular forms can be seen as *normal forms* for the term algebra. But with molecules as binding operators we obtain a view of the structure of actions which differs strikingly from that offered by conventional term structure. This section is a review of [23] whose main objective was to prove the isomorphism of these two presentations.

Notation Throughout, we shall adopt the convention that all names appearing in a vector within round brackets are distinct. Moreover, it will also be assumed that all terms and expressions used are well formed, and when they occur in definitions or equations, those occurring on each side have identical arities.

An action calculus is determined by a set $\mathcal{K}$ of *control* operators, called a *signature*, together with a set $\mathcal{R}$ of *reaction rules* whose form we shall define later. We let K range over controls.

Definition 2.5 (Controls (statics)) *A* control K *is an operator which allows the construction of an action* $K(\vec{a})$ *from a sequence* $\vec{a}$ *of actions, subject to a* rule of arity *having the following form:*

$$\frac{a_1 : m_1 \to n_1 \quad \cdots \quad a_r : m_r \to n_r}{K(a_1, \ldots, a_r) : m \to n} \, (\chi)$$

where the side-condition χ *may constrain the value of the integer* r *and the arities* m_i, n_i, m, n. ■

An example of a signature for the fragment of the π-calculus mentioned in the introduction is given by the set of controls $\{\nu, \mathbf{out}, \mathbf{box}\}$ with rules of arity as follows:

$$\nu : \epsilon \to p \qquad\qquad \mathbf{out} : p \otimes m \to \epsilon \qquad\qquad \frac{a : m \to n}{\mathbf{box}\, a : p \to n}$$

Another example is given by the signature $\mathcal{K} = \{\ulcorner\ \urcorner, \mathbf{ap}\}$ which gives a representation of the simply typed λ-calculus as an action calculi. To obtain the arrow types in the λ-calculus, we assume that M supports exponentiation $m \Rightarrow n$ of arities (with $m \Rightarrow n$ prime). The arity rules are then:

$$\frac{a : m \to n}{\ulcorner a \urcorner : \epsilon \to (m \Rightarrow n)} \qquad\qquad \mathbf{ap} : (m \Rightarrow n) \otimes m \to n$$

By combining signatures, any action calculus can be lifted to higher order; for instance, the higher order π-calculus is obtained by the signature containing the controls $\{\ulcorner\ \urcorner, \mathbf{ap}, \boldsymbol{\nu}, \mathbf{out}, \mathbf{box}\}$ together with their rules of arity. To indicate the union of the signature $\{\ulcorner\ \urcorner, \mathbf{ap}\}$ with some signature $\mathcal{K}$ we shall write $\mathcal{K}^{\Rightarrow}$.

2.2.1 Molecular Forms

We shall now define the following syntactic forms which will turn out to be normal forms for the actions of an action calculus.

Definition 2.6 (Molecular forms) *Let $\mathcal{K}$ be a set of controls. The molecular forms over $\mathcal{K}$, denoted $\mathcal{M}(\mathcal{K})$, are syntactic objects; they consist of the* actions a *and the* molecules μ *defined as follows:*

$$\begin{array}{lll} a & ::= & (\vec{x})\ \mu_1 \cdots \mu_r\ \langle\vec{u}\rangle \qquad\qquad (\vec{x} : m, \vec{u} : n, a : m \to n) \\ \mu & ::= & \langle\vec{v}\rangle\ K\vec{b}\,(\vec{y}) \qquad\qquad (\vec{v} : k, \vec{y} : l, K\vec{b} : k \to l) \end{array}$$

We let λ, μ range over molecules. In both actions and molecules, whenever a vector $\vec{x}$ occurs in round brackets, its names (which by our convention must be distinct) are binding occurrences with scope extending to the right to the end of the smallest enclosing action, capturing occurrences of the names $\vec{x}$ even within molecule constructions. Names which are not thus bound are free and alpha conversion of bound names is allowed. We assume that no name has more than one binding occurrence in any molecule or action.

In the action a of the above definition, $\vec{x}$ are called the imported names and $\vec{u}$, the exported ones. The construct $\mu_1 \cdots \mu_r$, called the body of a, is a possibly empty partial *sequence of molecules, where the commutation of any two molecules is allowed provided neither binds a name occurring free in the other.* ■

We shall now define the operations of an action structure, the control operations as well as two additional ones, *datum* $\langle x\rangle$ and *discard* $\boldsymbol{\omega}$, which represent provision of (exported) and discarding of (imported) names respectively.

Definition 2.7 *Assume* $a = (\vec{u})\,\vec{\lambda}\,\langle\vec{v}\rangle$ *and* $b = (\vec{x})\,\vec{\mu}\,\langle\vec{y}\rangle$ *where no name which is bound in one occurs in the other.*

$$\begin{array}{rcll}
\mathbf{id}_m & \stackrel{\text{def}}{=} & (\vec{x})\langle\vec{x}\rangle & (|\vec{x}| = m) \\
a \cdot b & \stackrel{\text{def}}{=} & (\vec{u})\,\vec{\lambda}\,\sigma\vec{\mu}\,\langle\sigma\vec{y}\rangle & (\sigma = \{\vec{v}/\vec{x}\}) \\
a \otimes b & \stackrel{\text{def}}{=} & (\vec{u}\vec{x})\vec{\lambda}\,\vec{\mu}\,\langle\vec{v}\vec{y}\rangle & \\
\mathbf{ab}_x a & \stackrel{\text{def}}{=} & (x\vec{u})\,\vec{\lambda}\,\langle x\vec{v}\rangle & \\
\langle x\rangle & \stackrel{\text{def}}{=} & (\,)\langle x\rangle & \\
\boldsymbol{\omega} & \stackrel{\text{def}}{=} & (x)\langle\,\rangle & \\
K(\vec{a}) & \stackrel{\text{def}}{=} & (x)\,\langle\vec{x}\rangle K\vec{a}(\vec{y})\,\langle\vec{y}\rangle & (\vec{x},\vec{y} \text{ not free in } \vec{a})
\end{array}$$

where $\{\vec{v}/\vec{x}\}$ *is simultaneous substitution of* $\vec{v}$ *for* $\vec{x}$*.* ■

Fact 2.8 x *is free in* a *if and only if* $\mathbf{ab}_x a \neq \mathbf{id} \otimes a$*.* ■

Proposition 2.9 $(\mathcal{M}(\mathcal{K}), \mathbf{id}, \cdot, \otimes, \mathbf{ab})$ *is a static action structure.* ■

2.2.2 The theory AC

We are now ready to define an action calculus as a quotient of a term algebra. An action calculus AC($\mathcal{K}$) possesses a set $\mathcal{K}$ of controls, each equipped with an arity rule. Each AC($\mathcal{K}$) is determined by its controls $\mathcal{K}$ together with a set of reaction rules which defines its dynamics.

Definition 2.10 (Terms) *The* terms *over* $\mathcal{K}$*, denoted by* $\mathcal{T}(\mathcal{K})$*, are generated as follows (we let* t *range over terms):*

$$t ::= \mathbf{id} \;\mid\; \langle x\rangle \;\mid\; \boldsymbol{\omega} \;\mid\; K\vec{t} \;\mid\; t_1 \cdot t_2 \;\mid\; t_1 \otimes t_2 \;\mid\; \mathbf{ab}_x t$$

where $\langle x\rangle : \epsilon \rightarrow p$ $(x{:}p)$ *and* $\boldsymbol{\omega} : p \rightarrow \epsilon$ *(for each* p*), and the other constructions have arities dictated by the arity rules of the constructors. The notions of* free name *and* bound name *are standard;* $\mathbf{ab}_x$ *binds* x *and* $\langle x\rangle$ *represents a free occurrence of* x*. The set of names free in* t *is denoted by* $\mathbf{fn}\,(t)$*.* ■

Definition 2.11 (Derived operations) *We define an alternative form* $(x)t$ *of* abstraction, *and the* permutors $\mathbf{p}_{m,n}$, *as follows (together with some abbreviations):*

$$
\begin{array}{rcll}
(x)t & \stackrel{\text{def}}{=} & \mathbf{ab}_x t \cdot (\boldsymbol{\omega} \otimes \mathbf{id}) & \\
(\vec{x})t & \stackrel{\text{def}}{=} & (x_1)\cdots(x_r)t & (\vec{x} = x_1 \cdots x_r,\ \textit{all distinct},\ r \geq 0) \\
\langle\vec{x}\rangle & \stackrel{\text{def}}{=} & \langle x_1\rangle \otimes \cdots \otimes \langle x_r\rangle & (\vec{x} = x_1 \cdots x_r,\ r \geq 0) \\
\mathbf{p}_{m,n} & \stackrel{\text{def}}{=} & (\vec{x}\vec{y})\langle\vec{y}\vec{x}\rangle & (\vec{x} : m,\ \vec{y} : n)
\end{array}
$$

■

Note that $\mathbf{p}_{m,n}$ is defined using a *particular* vector $\vec{x}\vec{y}$ of distinct names; with α-conversion, we shall be justified in choosing these names at will.

Although unsurprising, we define substitution upon terms in detail as we shall need a careful analysis of it later.

Definition 2.12 (Substitution) Substitution $\{y/x\}$ *upon terms is defined as follows:*

$$
\begin{array}{ll}
\{y/x\}\mathbf{id} \stackrel{\text{def}}{=} \mathbf{id} & \{y/x\}(t_1 \otimes t_2) \stackrel{\text{def}}{=} \{y/x\}t_1 \otimes \{y/x\}t_2 \\
\{y/x\}\boldsymbol{\omega} \stackrel{\text{def}}{=} \boldsymbol{\omega} & \{y/x\}(t_1 \cdot t_2) \stackrel{\text{def}}{=} \{y/x\}t_1 \cdot \{y/x\}t_2 \\
\{y/x\}\langle z\rangle \stackrel{\text{def}}{=} \langle z\rangle \quad (z \neq x) & \{y/x\}K(t,\ldots) \stackrel{\text{def}}{=} K(\{y/x\}t,\ldots) \\
\{y/x\}\langle x\rangle \stackrel{\text{def}}{=} \langle y\rangle &
\end{array}
$$

$$
\begin{array}{ll}
\{y/x\}\mathbf{ab}_z t \stackrel{\text{def}}{=} \mathbf{ab}_z\{y/x\}t & (z \notin \{x, y\}) \\
\{z/x\}\mathbf{ab}_z t \stackrel{\text{def}}{=} \mathbf{ab}_w\{z/x\}\{w/z\}t & (z \neq x,\ w \notin \mathbf{fn}(t) \cup \{x, z\}) \\
\{y/z\}\mathbf{ab}_z t \stackrel{\text{def}}{=} \mathbf{ab}_z t &
\end{array}
$$

■

Note that, in the penultimate equation, some particular w is chosen. We are not assuming α-convertibility at this stage, but it is a consequence of the axioms of action calculi given below.

Lemma 2.13 $\{x/x\}t = t$.

Proof Induction on the structure of terms. ■

Definition 2.14 (The theory AC) *The* equational theory AC *is the set of equations upon terms generated by the action structure axioms together with the following:*

$$\begin{array}{lll}
\gamma: & (x)t = \omega \otimes t & (x \notin \mathbf{fn}\,(t)) \\
\delta: & (x)(\langle x\rangle \otimes \mathbf{id}_m) = \mathbf{id}_{p\otimes m} & (x : p) \\
\zeta: & \mathbf{p}_{k,m} \cdot (t_2 \otimes t_1) = (t_1 \otimes t_2) \cdot \mathbf{p}_{\ell,n} & (t_1 : k \to \ell,\ t_2 : m \to n) \\
\sigma: & (\langle y\rangle \otimes \mathbf{id}_m) \cdot (x)t = \{y/x\}t & (t : m \to n)
\end{array}$$

■

With some abuse of terminology we shall consider AC to stand for either the above set of four axioms, or the set of equations inferred from them (a congruence relation). It will be clear from the context which we mean.

It is natural to ask why the axioms AC have been expressed using the derived form $(x)t$ of abstraction rather than directly using $\mathbf{ab}_x t$. This is mostly for convenience; note especially that the permutations are more directly definable using the derived form. However, there is an equivalent formulation of γ using $\mathbf{ab}_x$:

Proposition 2.15 *The theory* AC *is unchanged when the axiom γ is replaced by the following axiom:*

$$\gamma' : \quad \mathbf{ab}_x t = \mathbf{id} \otimes t \qquad (x \notin \mathbf{fn}\, t)$$

Proof Let AC′ be the theory given by replacing the axiom γ by γ'. Then it may be shown that γ is derivable in AC′.

$$\begin{array}{rcll}
(x)t & = & \mathbf{ab}_x t \cdot (\omega \otimes \mathbf{id}) & \\
& = & (\mathbf{id} \otimes t) \cdot (\omega \otimes \mathbf{id}) & \gamma' \\
& = & \omega \otimes t &
\end{array}$$

It may also be shown that γ' is derivable in the theory AC.

$$\begin{aligned}
\mathbf{ab}_x t &= \mathbf{ab}_x t \cdot (x)(\langle x\rangle \otimes \mathbf{id}) && \delta\\
&= \mathbf{ab}_x t \cdot \mathbf{ab}_x(\langle x\rangle \otimes \mathbf{id}) \cdot (\omega \otimes \mathbf{id})\\
&= \mathbf{ab}_x(t \cdot (\langle x\rangle \otimes \mathbf{id})) \cdot (\omega \otimes \mathbf{id})\\
&= \mathbf{ab}_x(\langle x\rangle \otimes \mathbf{id}) \cdot \mathbf{ab}_x(\mathbf{id} \otimes t) \cdot (\omega \otimes \mathbf{id})\\
&= \mathbf{ab}_x(\langle x\rangle \otimes \mathbf{id}) \cdot (x)(\mathbf{id} \otimes t)\\
&= \mathbf{ab}_x(\langle x\rangle \otimes \mathbf{id}) \cdot (\omega \otimes \mathbf{id} \otimes t) && \gamma\\
&= \mathbf{ab}_x(\langle x\rangle \otimes \mathbf{id}) \cdot (\omega \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes t) && \delta\\
&= \mathbf{id} \otimes t
\end{aligned}$$

■

We shall now derive several equations in the theory AC. These demonstrate the consequence of the theory and will also serve us in later proofs. In particular note that α-conversion is obtained.

Lemma 2.16 *The following are provable in* AC *whenever* $x \notin \mathbf{fn}\,(t_2)$*:*

1. $(x)(t_1 \cdot t_2) = (x)t_1 \cdot t_2$;

2. $(x)(t_1 \otimes t_2) = (x)t_1 \otimes t_2$;

3. $(x)(t_2 \otimes t_1) = t_2 \otimes (x)t_1$, *if* $t_2 : \epsilon \rightarrow \ell$.

α*:* $(y)t = (x)\{x/y\}t$, *if* $x \notin \mathbf{fn}\,(t)$;

4. $\mathbf{ab}_x t = (x)(\langle x\rangle \otimes t)$;

5. $(\vec{x})(\langle\vec{x}\rangle \otimes \mathbf{id}) = \mathbf{id}$;

6. $\mathbf{ab}_y t = \mathbf{ab}_x\{x/y\}t$.

Proof

$$\begin{aligned}
(1) \quad (x)(t_1 \cdot t_2) &= \mathbf{ab}_x t_1 \cdot (x)t_2\\
&= \mathbf{ab}_x t_1 \cdot (\omega \otimes t_2) && \gamma\\
&= \mathbf{ab}_x t_1 \cdot (\omega \otimes \mathbf{id}) \cdot t_2\\
&= (x)t_1 \cdot t_2
\end{aligned}$$

$$
\begin{aligned}
(*)\quad (x)(t_1 \otimes \mathbf{id}) &= (x)(((\langle x\rangle \otimes \mathbf{id}) \cdot (x)t_1 \otimes \mathbf{id}) && 2.13,\ \boldsymbol{\sigma} \\
&= (x)((\langle x\rangle \otimes \mathbf{id} \otimes \mathbf{id}) \cdot ((x)t_1 \otimes \mathbf{id}) && (1) \\
&= (x)t_1 \otimes \mathbf{id} && \boldsymbol{\delta}
\end{aligned}
$$

$$
\begin{aligned}
(2)\quad (x)(t_1 \otimes t_2) &= (x)((t_1 \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes t_2)) \\
&= (x)(t_1 \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes t_2) && (1) \\
&= ((x)t_1 \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes t_2) && (*) \\
&= (x)t_1 \otimes t_2
\end{aligned}
$$

$$
\begin{aligned}
(3)\quad (x)(t_2 \otimes t_1) &= (x)(t_1 \cdot (t_2 \otimes \mathbf{id})) \\
&= (x)t_1 \cdot (t_2 \otimes \mathbf{id}) && (1) \\
&= t_2 \otimes (x)t_1
\end{aligned}
$$

$$
\begin{aligned}
(\boldsymbol{\alpha})\quad (x)\{x/y\}t &= (x)(((\langle x\rangle \otimes \mathbf{id}) \cdot (y)t) && \boldsymbol{\sigma} \\
&= (x)((\langle x\rangle \otimes \mathbf{id}) \cdot (y)t && (1) \\
&= (y)t && \boldsymbol{\delta}
\end{aligned}
$$

$$
\begin{aligned}
(4)\quad \mathbf{ab}_x t &= \mathbf{ab}_x t \cdot (x)((\langle x\rangle \otimes \mathbf{id}) && \boldsymbol{\delta} \\
&= (x)(t \cdot ((\langle x\rangle \otimes \mathbf{id})) \\
&= (x)((\langle x\rangle \otimes t)
\end{aligned}
$$

(5) Induction on length of $\vec{x}$. Basis true by definition. For the inductive step:

$$
\begin{aligned}
(x\vec{y})((\langle x\vec{y}\rangle \otimes \mathbf{id}) &= (x)(\vec{y})(((\langle \vec{y}\rangle \otimes \mathbf{id}) \cdot ((\langle x\rangle \otimes \mathbf{id})) \\
&= (x)(((\vec{y})((\langle \vec{y}\rangle \otimes \mathbf{id}) \cdot ((\langle x\rangle \otimes \mathbf{id})) && (1)^* \\
&= (x)((\langle x\rangle \otimes \mathbf{id}) && \text{induction} \\
&= \mathbf{id} && \boldsymbol{\delta}
\end{aligned}
$$

(6) If $x = y$ then result follows by lemma 2.13. Assume $x \neq y$.

$$
\begin{aligned}
\mathbf{ab}_x\{x/y\}t &= (x)((\langle x\rangle \otimes \{x/y\}t) && (4) \\
&= (x)\{x/y\}((\langle y\rangle \otimes t) \\
&= (y)((\langle y\rangle \otimes t) && \boldsymbol{\alpha} \\
&= \mathbf{ab}_y t && (4)
\end{aligned}
$$

■

We are now ready to define action calculus.

Definition 2.17 (Action calculus: statics) *The* static action calculus $\mathsf{AC}^s(\mathcal{K})$ *is defined to be the quotient* $\mathcal{T}(\mathcal{K})/\mathsf{AC}$. ∎

Fact 2.18 $\mathsf{AC}^s(\mathcal{K})$ *is a static action structure.* ∎

The following theorem [23] shows that the molecular forms $\mathcal{M}(\mathcal{K})$ provide an explicit representation of $\mathsf{AC}^s(\mathcal{K})$.

Theorem 2.19 *For any signature* $\mathcal{K}$, *the static action structure* $\mathcal{M}(\mathcal{K})$ *of molecular forms is isomorphic to* $\mathsf{AC}^s(\mathcal{K})$. ∎

We shall now introduce the reaction rules which assign computational significance to the control operations.

Definition 2.20 (Controls (dynamics)) *A* reaction rule *over a signature* $\mathcal{K}$ *takes the form:*

$$t[\vec{a}] \searrow t'[\vec{a}]$$

where t, t' *are terms of* $\mathcal{T}(\mathcal{K})$ *which may contain metavariables* $\vec{a}$ *over actions.* ∎

An example of a reaction rule over the signature $\{\boldsymbol{\nu}, \mathbf{out}, \mathbf{box}\}$ presented earlier is:

$$((\langle x\rangle \otimes \mathbf{id})\cdot\mathbf{out}) \otimes (\langle x\rangle\cdot\mathbf{box}a) \searrow a$$

The reaction rules for the controls $\{\ulcorner\ \urcorner, \mathbf{ap}\}$ are

$$\searrow_\sigma : (\ulcorner t \urcorner \otimes \mathbf{id}) \cdot (x)t' \searrow \{{}^t\!/\!_x\}t' \qquad\qquad \searrow_\beta : (\ulcorner a \urcorner \otimes \mathbf{id}) \cdot \mathbf{ap} \searrow a$$

where $\{{}^t\!/\!_x\}t'$ signifies the substitution of $\ulcorner t \urcorner$ for each free occurrence $\langle x\rangle$ of x in t'. $\searrow_\sigma$ is actually a rule schema; giving a rule for each pair of terms t, t'. The second rule corresponds to β-reduction. The dynamics of $\mathsf{AC}(\ulcorner\ \urcorner, \mathbf{ap})$ is studied in more detail in [28].

It is important to note that a reaction relation need not be preserved by controls; thus from $a \searrow a'$ it does not follow that $\mathbf{box}a \searrow \mathbf{box}a'$. Indeed, the role of **box** on the π-calculus is to prevent such reaction from occurring thereby providing a form of sequential control over reactions.

Definition 2.21 (Action calculus: dynamics) *Let $\mathcal{R}$ be a set of reaction rules over a signature $\mathcal{K}$. Then the* (dynamic) action calculus $\mathsf{AC}(\mathcal{K}, \mathcal{R})$ *is the static action structure $\mathsf{AC}^s(\mathcal{K})$ equipped with the smallest reaction relation $\searrow$ which satisfies the rules $\mathcal{R}$ (for all replacements of metavariables $\vec{a}$ by actions).* ∎

We shall henceforth use $\mathsf{AC}^s(\mathcal{K})$ and $\mathsf{AC}(\mathcal{K}, \emptyset)$ interchangeably to denote the static action calculus over $\mathcal{K}$.

As an example of an action calculus we shall now bring together a signature and a set of reaction rules which together completely define the calculus PIC . In the light of the informal explanation given in Section 2.2, we note the correspondence between PIC and a fragment of the π-calculus. A similar correspondence with a variant of PIC is stated more formally at the end of chapter 6.

PIC is defined as the action calculus over the controls $\{\mathbf{out}, \mathbf{box}\}$ together with the following arity rules

$$\frac{}{\mathbf{out} : p \otimes m \to \epsilon} \qquad\qquad \frac{a : m \to n}{\mathbf{box}a : p \to n}$$

and the reaction rule $\mathbf{out}_x \otimes \mathbf{box}_x a \searrow a$ where

$$\begin{aligned} \mathbf{out}_x &\stackrel{\mathrm{def}}{=} (\langle x \rangle \otimes \mathbf{id}) \cdot \mathbf{out} \\ \mathbf{box}_x a &\stackrel{\mathrm{def}}{=} \langle x \rangle \cdot \mathbf{box}a \end{aligned}$$

Throughout this thesis we shall draw examples from the actions, signature and reaction rule of the above calculus.

Discussion The axiomatization of AC, though succinct, is impure in two ways; the axiom γ has a syntactic condition upon terms, and the axiom σ is expressed in terms of substitution of names into terms. Thus each of these axioms is more exactly an axiom schema: a finite presentation of an infinite set of axioms. We could define a control structure to be an enriched action structure which satisfies this infinite set of axioms, and then by an entirely standard argument we would find $\mathsf{AC}(\mathcal{K})$ to be initial in this subcategory of action structures.

One shortcoming of this approach is that it does not provide a semantic account of what it means for a name to belong to the "surface" of an action, generalising the syntactic notion of free occurrence of a name in a term. Another is that instances of $\boldsymbol{\gamma}$ or $\boldsymbol{\sigma}$, interpreted in an arbitrary action structure A, constrain only those actions which lie in the image of $\mathsf{AC}(\mathcal{K})$ under a homomorphism; they impose no constraint upon actions of A *in general*, and thus contribute no understanding of A as an algebraic structure. Finally, a finite set of axioms is more satisfactory than an infinite set.

Bearing these arguments in mind, in the spirit of universal algebra we seek to characterise control structures by a finite set of pure axioms, such that $\mathsf{AC}^s(\mathcal{K})$ is the initial control structure over $\mathcal{K}$. Apart from the greater elegance of this approach and greater mathematical insight it provides, it has the advantage that properties such as initiality then follow by standard arguments.

Initiality will be ensured if the axioms we propose generate exactly the theory AC, i.e. they are equipotent with $\boldsymbol{\gamma}$, $\boldsymbol{\delta}$, $\boldsymbol{\zeta}$ and $\boldsymbol{\sigma}$ over the term algebra. This condition does not fully determine the notion of control structure; therefore we must justify our choice. Our axiomatisation has other qualities; it is simple, it is a natural extension of a known categorical structure (symmetric monoidal categories) and it gives a convincing account of the notion of surface.

Before presenting the axioms, let us further analyse the central problem. The greatest difficulty is to replace the axiom schema

$$\boldsymbol{\gamma}' : \mathbf{ab}_x t = \mathbf{id} \otimes t \qquad (x \notin \mathbf{fn}\,(t))$$

(which by Proposition 2.15 is equipotent with $\boldsymbol{\gamma}$) by a finite set of purely algebraic axioms. A less satisfactory solution is to give up the purely algebraic approach, and to postulate that every control structure is equipped with a map **surf** which assigns to each action a a set $\mathbf{surf}(a) \subseteq X$; then one adopts the single axiom

$$\gamma'' : \mathbf{ab}_x a = \mathbf{id} \otimes a \qquad (x \notin \mathbf{surf}(a))$$

One also imposes upon **surf** the reasonable condition that, roughly, the surface of each algebraic construction is no greater than the union of the surfaces of its arguments. More precisely, one imposes the following *surface* axioms:

$$\begin{aligned}
\mathbf{surf}(\mathbf{id}) &= \emptyset \\
\mathbf{surf}(a \otimes b) &\subseteq \mathbf{surf}(a) \cup \mathbf{surf}(b) \\
\mathbf{surf}(a \cdot b) &\subseteq \mathbf{surf}(a) \cup \mathbf{surf}(b) \\
\mathbf{surf}(\mathbf{ab}_x a) &\subseteq \mathbf{surf}(a) - \{x\} \\
\mathbf{surf}(\langle x \rangle) &\subseteq \{x\} \\
\mathbf{surf}(\omega) &= \emptyset \\
\mathbf{surf}(K\vec{a}) &\subseteq \textstyle\bigcup_i \mathbf{surf}(a_i) \qquad (\vec{a} = a_1 \cdots a_r)
\end{aligned}$$

One then obtains a finite (but not purely algebraic) set of axioms for control structures which ensures that $\mathsf{AC}(\mathcal{K})$ is initial.

This was indeed our first approach. We were then surprised to find that from these axioms one can derive the double implication

$$\mathbf{ab}_x a = \mathbf{id} \otimes a \;\Leftrightarrow\; x \notin \mathbf{surf}(a)$$

To see this, note that one direction ($\Leftarrow$) is already given by γ''. For the other ($\Rightarrow$), suppose that $\mathbf{ab}_x a = \mathbf{id} \otimes a$. For any y we have

$$a = (\langle y \rangle \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes a) \cdot (\omega \otimes \mathbf{id})$$

using $\langle y \rangle \cdot \omega = \mathbf{id}_\epsilon$ which is ensured by the other control structure axioms. It follows that

$$\begin{aligned}
\mathbf{surf}(a) &\subseteq \{y\} \cup \mathbf{surf}(\mathbf{id} \otimes a) && \text{by the surface axioms} \\
&= \{y\} \cup \mathbf{surf}(\mathbf{ab}_x a) && \text{by assumption} \\
&= \{y\} \cup (\mathbf{surf}(a) - \{x\}) && \text{by surface axiom}
\end{aligned}$$

and by choosing $y \neq x$ we deduce $x \notin \mathbf{surf}(a)$. In other words, the surface axioms have constrained $\mathbf{surf}(a)$ to be exactly the set $\{x \mid \mathbf{ab}_x a \neq \mathbf{id} \otimes a\}$. We therefore have the same effect if we remove γ'' and *define* surface by

$$\mathbf{surf}(a) \stackrel{\text{def}}{=} \{x \mid \mathbf{ab}_x a \neq \mathbf{id} \otimes a\}$$

The axioms are still not purely algebraic, since the surface axioms remain; each of these has now become an implication between equations. Our second discovery was that these implications can be replaced (with equivalent power) by a small number of purely equational axioms.

It is convenient to introduce the axioms in two steps. The first step is to define symmetric action structures, an enrichment of symmetric monoidal categories.

2.3 Symmetric Action Structures

We begin by recalling the standard notion of a symmetric monoidal category.

Definition 2.22 (Symmetry) *A* symmetry *on a strict monoidal category is a family of arrows* $\mathbf{c}$ *with components* $\mathbf{c}_{m,n} : m \otimes n \rightarrow n \otimes m$ *such that*

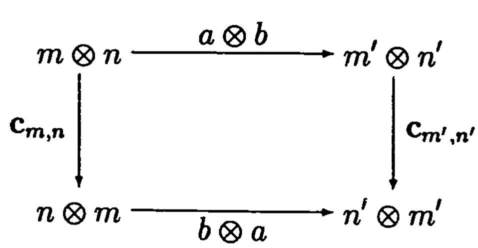
$$\mathrm{S}_1 : \quad \mathbf{c}_{m,n} \cdot (b \otimes a) = (a \otimes b) \cdot \mathbf{c}_{m',n'}$$

$$\mathrm{S}_2 : \quad \mathbf{c}_{m,n} \cdot \mathbf{c}_{n,m} = \mathbf{id}$$

$$\mathrm{S}_3 : \quad (\mathbf{c}_{m,n} \otimes \mathbf{id}_k) \cdot (\mathbf{id}_n \otimes \mathbf{c}_{m,k}) = \mathbf{c}_{m,n\otimes k}$$

where $a : m \rightarrow m'$, $b : n \rightarrow n'$. ∎

Remark The axiom S_1 states that the symmetry is a natural transformation as can be seen by expressing S_1 by the following commutative diagram:

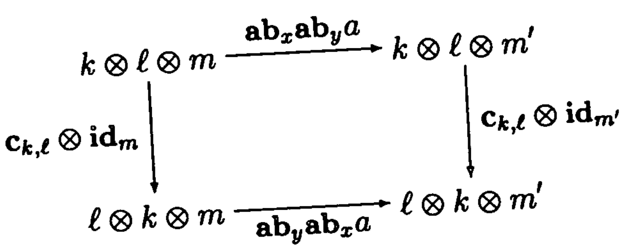


Definition 2.23 (Symmetric action structure) *A symmetric action structure is an action structure with a symmetry* c *on it for which*

$$\mathrm{S}_4: \quad \mathbf{ab}_x c = \mathbf{id} \otimes c$$
$$\mathrm{S}_5: \quad \mathbf{ab}_x(\mathbf{ab}_x a) = \mathbf{id} \otimes \mathbf{ab}_x a$$
$$\mathrm{S}_6: \quad \mathbf{ab}_x(a \otimes \mathbf{id}) = \mathbf{ab}_x a \otimes \mathbf{id}$$
$$\mathrm{S}_7: \quad (\mathbf{c}_{k,\ell} \otimes \mathbf{id}) \cdot \mathbf{ab}_y \mathbf{ab}_x a = \mathbf{ab}_x \mathbf{ab}_y a \cdot (\mathbf{c}_{k,\ell} \otimes \mathbf{id}) \quad (x : k, y : \ell, x \neq y)$$

■

Remark It may be helpful to express the axiom S_7 by means of the following commuting diagram:



Lemma 2.24 *Let* $a : m \to n$, $b : k \to \ell$, $x : k$ *and* $y : \ell$, *where* x, y *are distinct names. The following equations are valid in symmetric action structures:*

1. $a \otimes b = \mathbf{c}_{m,k} \cdot (b \otimes a) \cdot \mathbf{c}_{\ell,n}$;
2. $\mathbf{ab}_x \mathbf{ab}_y a = (\mathbf{c}_{k,l} \otimes \mathbf{id}) \cdot \mathbf{ab}_y \mathbf{ab}_x a \cdot (\mathbf{c}_{l,k} \otimes \mathbf{id})$;
3. $(\mathbf{id}_k \otimes \mathbf{c}_{m,n}) \cdot (\mathbf{c}_{k,n} \otimes \mathbf{id}_m) = \mathbf{c}_{k \otimes m, n}$;
4. $\mathbf{c}_{m,\epsilon} = \mathbf{id}_m = \mathbf{c}_{\epsilon,m}$.

Proof

(1)
$$\begin{aligned} a \otimes b &= (a \otimes b) \cdot \mathbf{c}_{n,\ell} \cdot \mathbf{c}_{\ell,n} && \mathrm{S}_2 \\ &= \mathbf{c}_{m,k} \cdot (b \otimes a) \cdot \mathbf{c}_{\ell,n} && \mathrm{S}_1 \end{aligned}$$

(2)
$$\begin{aligned} \mathbf{ab}_x \mathbf{ab}_y a &= \mathbf{ab}_x \mathbf{ab}_y a \cdot (\mathbf{c}_{k,l} \otimes \mathbf{id}) \cdot (\mathbf{c}_{l,k} \otimes \mathbf{id}) && \mathrm{S}_2 \\ &= (\mathbf{c}_{k,l} \otimes \mathbf{id}) \cdot \mathbf{ab}_y \mathbf{ab}_x a \cdot (\mathbf{c}_{l,k} \otimes \mathbf{id}) && \mathrm{S}_7 \end{aligned}$$

(3)
$$\begin{aligned}
\mathbf{c}_{k\otimes m,n} &= \mathbf{c}_{k\otimes m,n}\cdot(\mathbf{c}_{n,k}\otimes\mathbf{id}_m)\cdot(\mathbf{c}_{k,n}\otimes\mathbf{id}_m) && \mathrm{S}_2\\
&= \mathbf{c}_{k\otimes m,n}\cdot(\mathbf{c}_{n,k}\otimes\mathbf{id}_m)\cdot(\mathbf{id}_k\otimes\mathbf{c}_{n,m})\\
&\qquad\cdot(\mathbf{id}_k\otimes\mathbf{c}_{m,n})\cdot(\mathbf{c}_{k,n}\otimes\mathbf{id}_m) && \mathrm{S}_2\\
&= \mathbf{c}_{k\otimes m,n}\cdot\mathbf{c}_{n,k\otimes m}\cdot(\mathbf{id}_k\otimes\mathbf{c}_{m,n})\cdot(\mathbf{c}_{k,n}\otimes\mathbf{id}_m) && \mathrm{S}_3\\
&= (\mathbf{id}_k\otimes\mathbf{c}_{m,n})\cdot(\mathbf{c}_{k,n}\otimes\mathbf{id}_m) && \mathrm{S}_2
\end{aligned}$$

(4) We show that for every n, $\mathbf{id}_n\otimes\mathbf{c}_{m,\epsilon}=\mathbf{id}_n\otimes\mathbf{id}_m$: then $n=\epsilon$ gives result.

$$\begin{aligned}
\mathbf{id}_n\otimes\mathbf{id}_m &= \mathbf{c}_{n,m}\cdot\mathbf{c}_{m,n} && \mathrm{S}_2\\
&= \mathbf{c}_{n,m}\cdot(\mathbf{c}_{m,n}\otimes\mathbf{id}_\epsilon)\cdot(\mathbf{id}_n\otimes\mathbf{c}_{m,\epsilon}) && \mathrm{S}_3\\
&= \mathbf{id}_n\otimes\mathbf{c}_{m,\epsilon} && \mathrm{S}_2
\end{aligned}$$

■

Now, prompted by Fact 2.8, we define a semantic notion of surface. Intuitively, the surface of an action a contains just those names x for which the abstractor $\mathbf{ab}_x$ acts non-trivially upon a.

Definition 2.25 (Surface) *Call the set*

$$\{x\in X \mid \mathbf{ab}_x a\neq\mathbf{id}\otimes a\}$$

the surface *of a, written* $\mathbf{surf}(a)$. ■

Remark By Fact 2.8, when a is a molecular form its surface is exactly its set of free names.

Another way to express our semantic understanding is that an action "depends upon" a name x just in the case when x is in its surface. Whatever "depends upon" means, it should surely be the case that a compound action depends upon no more names than do its components (taken together). The notion of symmetric action structure is significant, compared to that of action structure, because it entails a proposition which expresses this property:

Proposition 2.26

1. $\mathbf{surf}(\mathbf{id})=\emptyset$*;*

2. $\mathbf{surf}\,(c) = \emptyset$;

3. $\mathbf{surf}\,(a \cdot b) \subseteq \mathbf{surf}\,a \cup \mathbf{surf}\,b$;

4. $\mathbf{surf}\,(a \otimes b) \subseteq \mathbf{surf}\,a \cup \mathbf{surf}\,b$;

5. $\mathbf{surf}\,(\mathbf{ab}_x a) \subseteq \mathbf{surf}\,a - \{x\}$.

Proof (1) and (2) follow trivially from the axioms $\mathbf{ab}_x\mathbf{id} = \mathbf{id}$ and (S_4) respectively. For (3) and (4) it suffices to show that if $\mathbf{ab}_x a = \mathbf{id} \otimes a$ and $\mathbf{ab}_x b = \mathbf{id} \otimes b$ then $\mathbf{ab}_x(a \cdot b) = \mathbf{id} \otimes (a \cdot b)$ and $\mathbf{ab}_x(a \otimes b) = \mathbf{id} \otimes a \otimes b$. For (5), by ($S_5$) we have $x \notin \mathbf{surf}\,(\mathbf{ab}_x a)$. So, let $y \notin \mathbf{surf}\,(a)$ with $y \neq x$. Assume $y : \ell$, $x : k$ and $a : m \to n$.

(3) $\mathbf{ab}_x(a \cdot b)$

$$
\begin{aligned}
&= \mathbf{ab}_x a \cdot \mathbf{ab}_x b \\
&= (\mathbf{id} \otimes a) \cdot (\mathbf{id} \otimes b) \\
&= \mathbf{id} \otimes (a \cdot b)
\end{aligned}
$$

(4) $\mathbf{ab}_x(a \otimes b)$

$$
\begin{aligned}
&= \mathbf{ab}_x((a \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes b)) \\
&= \mathbf{ab}_x(a \otimes \mathbf{id}) \cdot \mathbf{ab}_x(\mathbf{id} \otimes b) \\
&= (\mathbf{ab}_x a \otimes \mathbf{id}) \cdot \mathbf{ab}_x(\mathbf{id} \otimes b) && S_6 \\
&= (\mathbf{ab}_x a \otimes \mathbf{id}) \cdot \mathbf{ab}_x(c \cdot (b \otimes \mathbf{id}) \cdot c) && 2.24(1) \\
&= (\mathbf{ab}_x a \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes c) \cdot (\mathbf{ab}_x b \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes c) && S_4,\ S_6 \\
&= (\mathbf{id} \otimes a \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes c) \cdot (\mathbf{id} \otimes b \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes c) \\
&= (\mathbf{id} \otimes a \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes \mathbf{id} \otimes b) && 2.24(1) \\
&= \mathbf{id} \otimes a \otimes b
\end{aligned}
$$

(5) $\mathbf{ab}_y\mathbf{ab}_x a$

$=$	$(\mathbf{c}_{\ell,k} \otimes \mathbf{id}) \cdot \mathbf{ab}_x\mathbf{ab}_y a \cdot (\mathbf{c}_{k,\ell} \otimes \mathbf{id})$	2.24(2)
$=$	$(\mathbf{c}_{\ell,k} \otimes \mathbf{id}) \cdot \mathbf{ab}_x(\mathbf{id}_\ell \otimes a) \cdot (\mathbf{c}_{k,\ell} \otimes \mathbf{id})$	
$=$	$(\mathbf{c}_{\ell,k} \otimes \mathbf{id}) \cdot \mathbf{ab}_x(\mathbf{c}_{\ell,m} \cdot (a \otimes \mathbf{id}) \cdot \mathbf{c}_{n,\ell}) \cdot (\mathbf{c}_{k,\ell} \otimes \mathbf{id})$	2.24(1)
$=$	$(\mathbf{c}_{\ell,k} \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes \mathbf{c}_{\ell,m}) \cdot \mathbf{ab}_x(a \otimes \mathbf{id})$ $\cdot(\mathbf{id} \otimes \mathbf{c}_{n,\ell}) \cdot (\mathbf{c}_{k,\ell} \otimes \mathbf{id})$	S_6
$=$	$\mathbf{c}_{\ell,k\otimes m} \cdot (\mathbf{ab}_x a \otimes \mathbf{id}) \cdot \mathbf{c}_{k\otimes n,\ell}$	S_3, 2.24(3)
$=$	$\mathbf{id} \otimes \mathbf{ab}_x a$	2.24(1) ■

Remarks In proposition 2.26, clauses (3)–(5) are inclusions rather than equations. In the more refined class of models for action calculi given by control structures (see the following section) we prove a stronger version of (5) with inclusion being replaced by equality.

However, equality does not hold for (3) or (4). A counterexample for (3) given by $\langle x \rangle \cdot \omega = \mathbf{id}_\epsilon$, which holds in any action calculus. For (4), a counterexample is provided by the action structure whose typical element is of the form $(\vec{x})g\langle\vec{y}\rangle$, where g is an element of the free abelian group generated by the names X; thus g takes the form $x_1^{h_1} \cdots x_r^{h_r}$ where $h_1, \ldots, h_r$ are integers[1]. The tensor product of $a = (\vec{u})f\langle\vec{v}\rangle$ and $b = (\vec{x})g\langle\vec{y}\rangle$ is $(\vec{u}\vec{x})f \times g\langle\vec{v}\vec{y}\rangle$, where $f \times g$ is the group product. If $a = xy^{-1}$ and $b = yz^2$ are two actions of arity $\epsilon \to \epsilon$, then y lies in the surface of a and of b but not in the surface of $a \otimes b = xz^2$.

Proposition 2.27 *The action calculus* $\mathsf{AC}(\mathcal{K}, \mathcal{R})$ *is a symmetric action structure.*

Proof We take the permutations $\mathbf{p}_{m,n}$ as the symmetry on $\mathsf{AC}(\mathcal{K}, \mathcal{R})$. Naturality ($S_1$) is immediate by ζ. We shall now show that axioms S_2–S_7 are provable in AC. In the proofs that follow assume that $\vec{x} : m$, $\vec{v} : m$, $\vec{y} : n$, $\vec{u} : n$ and $\vec{w} : k$ and that names $\vec{u}, \vec{v}, \vec{w}, \vec{x}, \vec{y}$ are all distinct. Reasons for each step appear in the right-hand column; an asterisk indicates repeated use of an equation.

---

[1]This example arises as a quotient of the action structure for Synchronous CCS [21].

(S₂) $\mathbf{p}_{m,n} \cdot \mathbf{p}_{n,m}$

$$
\begin{aligned}
&= (\vec{x}\vec{y})\langle\vec{y}\vec{x}\rangle \cdot (\vec{u}\vec{v})\langle\vec{v}\vec{u}\rangle && 2.16(1)^* \\
&= (\vec{x}\vec{y})(\langle\vec{y}\vec{x}\rangle \cdot (\vec{u}\vec{v})\langle\vec{v}\vec{u}\rangle) && \sigma^* \\
&= (\vec{x}\vec{y})\langle\vec{x}\vec{y}\rangle && 2.16(5) \\
&= \mathbf{id}
\end{aligned}
$$

(S₃) $(\mathbf{p}_{m,n} \otimes \mathbf{id}_k) \cdot (\mathbf{id}_n \otimes \mathbf{p}_{m,k})$

$$
\begin{aligned}
&= ((\vec{x}\vec{y})\langle\vec{y}\vec{x}\rangle \otimes \mathbf{id}_k) \cdot (\mathbf{id}_n \otimes (\vec{v}\vec{w})\langle\vec{w}\vec{v}\rangle) && 2.16(2)^* \\
&= (\vec{x}\vec{y})(\langle\vec{y}\vec{x}\rangle \otimes \mathbf{id}_k) \cdot (\mathbf{id}_n \otimes (\vec{v}\vec{w})\langle\vec{w}\vec{v}\rangle) && 2.16(1)^* \\
&= (\vec{x}\vec{y})((\langle\vec{y}\vec{x}\rangle \otimes \mathbf{id}_k) \cdot (\mathbf{id}_n \otimes (\vec{v}\vec{w})\langle\vec{w}\vec{v}\rangle)) \\
&= (\vec{x}\vec{y})(((\langle\vec{y}\rangle \cdot \mathbf{id}_n) \otimes (((\langle\vec{x}\rangle \otimes \mathbf{id}_k) \cdot (\vec{v}\vec{w})\langle\vec{w}\vec{v}\rangle)) && \sigma^* \\
&= (\vec{x}\vec{y})(\langle\vec{y}\rangle \otimes (\vec{w})\langle\vec{w}\vec{x}\rangle) && 2.16(3)^* \\
&= (\vec{x}\vec{y}\vec{w})\langle\vec{y}\vec{w}\vec{x}\rangle) \\
&= \mathbf{p}_{m,n\otimes k}
\end{aligned}
$$

(S₄) $\mathbf{ab}_x\mathbf{p}$

$$
\begin{aligned}
&= (x)(\langle x\rangle \otimes \mathbf{p}) && 2.16(4) \\
&= (x)\langle x\rangle \otimes \mathbf{p} && 2.16(2) \\
&= \mathbf{id} \otimes \mathbf{p} && \delta
\end{aligned}
$$

(S₅) $\mathbf{ab}_x(\mathbf{ab}_x t)$

$$
\begin{aligned}
&= (x)(\langle x\rangle \otimes \mathbf{ab}_x t) && 2.16(4) \\
&= (x)\langle x\rangle \otimes \mathbf{ab}_x t && 2.16(2) \\
&= \mathbf{id} \otimes \mathbf{ab}_x t && \delta
\end{aligned}
$$

(S₆) $\mathbf{ab}_x(t \otimes \mathbf{id})$

$$
\begin{aligned}
&= (x)(\langle x\rangle \otimes t \otimes \mathbf{id}) && 2.16(4) \\
&= (x)(\langle x\rangle \otimes t) \otimes \mathbf{id} && 2.16(2) \\
&= \mathbf{ab}_x t \otimes \mathbf{id} && 2.16(4)
\end{aligned}
$$

$$
\begin{aligned}
(\mathrm{S}_7)\quad & \mathbf{ab}_x\mathbf{ab}_y t \cdot (\mathbf{p}_{p,q} \otimes \mathbf{id}) \\
= {} & \mathbf{ab}_x\mathbf{ab}_y t \cdot ((xy)\langle yx\rangle \otimes \mathbf{id}) \\
= {} & \mathbf{ab}_x\mathbf{ab}_y t \cdot (xy)(\langle yx\rangle \otimes \mathbf{id}) && 2.16(2) \\
= {} & \mathbf{ab}_x\mathbf{ab}_y t \cdot \mathbf{ab}_x\mathbf{ab}_y(\langle yx\rangle \otimes \mathbf{id}) \cdot (x)(\boldsymbol{\omega} \otimes \mathbf{id}) \\
= {} & \mathbf{ab}_x\mathbf{ab}_y(\langle yx\rangle \otimes t) \cdot (x)(\boldsymbol{\omega} \otimes \mathbf{id}) \\
= {} & (xy)(\langle yx\rangle \otimes t) \\
= {} & (xy)((\langle x\rangle \otimes \mathbf{id}) \cdot (x)(\langle yx\rangle \otimes t)) && 2.13,\sigma \\
= {} & (xy)((\langle x\rangle \otimes \mathbf{id}) \cdot (\langle y\rangle \otimes \mathbf{id}) \cdot (yx)(\langle yx\rangle \otimes t)) && 2.13,\sigma \\
= {} & (xy)((\langle yx\rangle \otimes \mathbf{id}) \cdot (yx)(\langle yx\rangle \otimes t)) \\
= {} & (xy)((\langle yx\rangle \otimes \mathbf{id}) \cdot (y)(\langle y\rangle \otimes (x)(\langle x\rangle \otimes t))) && 2.16(3) \\
= {} & (xy)((\langle yx\rangle \otimes \mathbf{id}) \cdot (y)(\langle y\rangle \otimes \mathbf{ab}_x t)) && 2.16(4) \\
= {} & (xy)((\langle yx\rangle \otimes \mathbf{id}) \cdot \mathbf{ab}_y\mathbf{ab}_x t) && 2.16(4) \\
= {} & (xy)(\langle yx\rangle \otimes \mathbf{id}) \cdot \mathbf{ab}_y\mathbf{ab}_x t && 2.16(1)^* \\
= {} & (\mathbf{p}_{p,q} \otimes \mathbf{id}) \cdot \mathbf{ab}_y\mathbf{ab}_x t && 2.16(2)^*
\end{aligned}
$$

■

2.4 Control Structures

We have prepared the way for the central definition and result of the chapter, namely the definition of control structures over $\mathcal{K}$ and the proof that $\mathsf{AC}^s(\mathcal{K})$ is the initial control structure. Our strategy has been to find a finitary axiomatization of the equational theory AC (see [24]); once this is found, the step to a suitable category of models for the molecular forms is much better defined.

Definition 2.28 (control structure) *Let A be a symmetric action structure (over X). Let $\mathcal{K}$ be a set of controls, equipped with reaction rules. Then A together with*

- datum $\langle x\rangle^A : \epsilon \to p$ *for each* $x : p \in X$*;*

- *a* discard *operation* $\boldsymbol{\omega}^A : p \to \epsilon$*, for each prime arity* p*;*

- *a* control *operation* K^A *for each* $K \in \mathcal{K}$*, obeying the arity rules for* K*;*

is a control structure over K *if*

Surface	γ_1 :	$\mathbf{ab}_x\langle y\rangle = \mathbf{id} \otimes \langle y\rangle$	*if* $y \neq x$
	γ_2 :	$\mathbf{ab}_x\boldsymbol{\omega} = \mathbf{id} \otimes \boldsymbol{\omega}$	
Datum	ϵ :	$(x)\langle x\rangle = \mathbf{id}$	
Substitution	σ_1 :	$[^x\!/\!_x]a = a$	
	σ_2 :	$[^y\!/\!_x](\langle x\rangle \otimes \langle x\rangle) = \langle y\rangle \otimes \langle y\rangle$	
	σ_3 :	$[^y\!/\!_x]K(a_1, \cdots, a_n) = K([^y\!/\!_x]a_1, \cdots, [^y\!/\!_x]a_n)$	

where

$$(x)a \stackrel{\text{def}}{=} (\mathbf{ab}_x a) \cdot (\boldsymbol{\omega} \otimes \mathbf{id})$$
$$[^y\!/\!_x]a \stackrel{\text{def}}{=} (\langle y\rangle \otimes \mathbf{id}) \cdot (x)a \qquad (\text{arity}(x) = \text{arity}(y))$$

■

Remarks The operation $[^y\!/\!_x]$ is called *semantic substitution.* Notice that the axiom $\boldsymbol{\sigma}$ simply asserts that, in $\mathsf{AC}^s(\mathcal{K})$, semantic substitution agrees with syntactic substitution.

The axioms γ_1 and γ_2 are counterparts to γ in AC; ϵ is an instance of $\boldsymbol{\delta}$ and σ_1–σ_3, in the presence of the other axioms correspond to the substitution equations together with $\boldsymbol{\sigma}$.

Note that the employment of symmetry in our formulation has allowed us to avoid the use of vectors of names and also to isolate the treatment of datum, discard and the controls. We shall discuss alternative axiomatisation after proposition 2.36.

The following proposition expresses the interaction between data and discard.

Proposition 2.29 (Absorption) $\langle x\rangle \cdot \boldsymbol{\omega} = \mathbf{id}_\epsilon$.

Proof

$$\begin{aligned} \mathbf{id}_\epsilon &= [^x\!/\!_x]\mathbf{id}_\epsilon && \sigma_1 \\ &= \langle x\rangle \cdot \mathbf{ab}_x\mathbf{id}_\epsilon \cdot \boldsymbol{\omega} \\ &= \langle x\rangle \cdot \mathbf{id} \cdot \boldsymbol{\omega} \\ &= \langle x\rangle \cdot \boldsymbol{\omega} \end{aligned}$$

■

Proposition 2.30 *For any a and x, the following are equivalent*

1. $\mathbf{ab}_x a = \mathbf{id} \otimes a$;

2. $(x)a = \boldsymbol{\omega} \otimes a$;

3. $[y/x]a = a$, *for all* y.

Proof By definition (1) implies (2) and, by proposition 2.29 and the definition of $[y/x]$, (2) implies (3). To show that (3) implies (1) choose $y \neq x$.

$$\begin{aligned}
\mathbf{ab}_x a &= \mathbf{ab}_x([y/x]a) \\
&= \mathbf{ab}_x(((\langle y\rangle \otimes \mathbf{id}) \cdot (x)a) \\
&= \mathbf{ab}_x(\langle y\rangle \otimes \mathbf{id}) \cdot \mathbf{ab}_x(x)a \\
&= (\mathbf{id} \otimes \langle y\rangle \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes (x)a) && \mathrm{S}_6,\gamma_1,\mathrm{S}_5,\gamma_2 \\
&= \mathbf{id} \otimes (((\langle y\rangle \otimes \mathbf{id}) \cdot (x)a) \\
&= \mathbf{id} \otimes [y/x]a \\
&= \mathbf{id} \otimes a
\end{aligned}$$

■

Remark The above proposition can be regarded as the semantic equivalent of γ. If $x \notin \mathbf{surf}(a)$, then by the definition of surface, $\mathbf{ab}_x a = \mathbf{id} \otimes a$. By proposition 2.30, $(x)a = \boldsymbol{\omega} \otimes a$.

Proposition 2.31 (surface)

1. $\mathbf{surf}(\langle x\rangle) \subseteq \{x\}$;

2. $\mathbf{surf}(\boldsymbol{\omega}) = \emptyset$;

3. $\mathbf{surf}(\mathbf{ab}_x a) = \mathbf{surf}(a) - \{x\}$;

4. $\mathbf{surf}(K(a_1, \cdots, a_n)) \subseteq \bigcup_{1\leq i\leq n} \mathbf{surf}\, a_i$.

Proof (1) and (2) follow trivially from γ_1 and γ_2 respectively.

For (3) we need only show that $\mathbf{surf}(\mathbf{ab}_x a) \supseteq \mathbf{surf}(a) - \{x\}$ since proposition 2.26(5) gives the other inclusion. By $\boldsymbol{\sigma}_1$, $a = (\langle x\rangle \otimes \mathbf{id}) \cdot \mathbf{ab}_x a \cdot (\boldsymbol{\omega} \otimes \mathbf{id})$. Hence, by (1), (2) and proposition 2.26 (3) and (4) we get $\mathbf{surf}(a) \subseteq \{x\} \cup \mathbf{surf}(\mathbf{ab}_x a)$ and the result follows immediately.

To show (4), assume $x \notin \mathbf{surf}(a_i)$, for all a_i in $K\vec{a}$. Now by $\boldsymbol{\sigma}_3$, $[y/x]K\vec{a} = K[y/x]\vec{a}$. By assumption, for each i, $\mathbf{ab}_x a_i = \mathbf{id} \otimes a_i$, so by proposition 2.30, $[y/x]a_i = a_i$. Hence $[y/x]K\vec{a} = K\vec{a}$ and by proposition 2.30, the result follows. ■

Remark We do not have $\mathbf{surf}(\langle x\rangle) = \{x\}$ in general, since in the trivial control structure where all terms of the same arity are identified (the terminal control structure) the surface of each term is necessarily empty. Note also that we have refined proposition 2.26(5) by equality rather than inclusion.

Proposition 2.32 **($\boldsymbol{\delta}$)** $(x)(\langle x\rangle \otimes \mathbf{id}) = \mathbf{id}$.

Proof

$$
\begin{aligned}
(x)(\langle x\rangle \otimes \mathbf{id}) &= \mathbf{ab}_x(\langle x\rangle \otimes \mathbf{id}) \cdot (\boldsymbol{\omega} \otimes \mathbf{id}) && \mathrm{S}_6 \\
&= (\mathbf{ab}_x\langle x\rangle \otimes \mathbf{id}) \cdot (\boldsymbol{\omega} \otimes \mathbf{id}) \\
&= (\mathbf{ab}_x\langle x\rangle \cdot (\boldsymbol{\omega} \otimes \mathbf{id})) \otimes \mathbf{id} \\
&= (x)\langle x\rangle \otimes \mathbf{id} \\
&= \mathbf{id} && \boldsymbol{\epsilon}
\end{aligned}
$$
■

Proposition 2.33 *The following equations hold in a control structure whenever* $x \notin \mathbf{surf}(b)$*:*

1. $(x)(a \cdot b) = (x)a \cdot b$;

2. $(x)(a \otimes b) = (x)a \otimes b$;

3. $(x)(b \otimes a) = (\mathbf{c}_{p,m} \otimes \mathbf{id}) \cdot (b \otimes (x)a)$, *if* $b : m \rightarrow n$.

α: $(y)b = (x)[x/y]b$;

4. $\mathbf{ab}_x a = (x)(\langle x\rangle \otimes a)$;

5. $(\vec{x})(\langle\vec{x}\rangle \otimes \mathbf{id}) = \mathbf{id}$;

6. $(x)(y)a = (\mathbf{c}_{p,q} \otimes \mathbf{id}) \cdot (y)(x)a$, *where* $x : p, y : q$.

Proof

(1)
$$\begin{aligned} (x)(a \cdot b) &= \mathbf{ab}_x a \cdot (x)b && \\ &= \mathbf{ab}_x a \cdot (\omega \otimes b) && 2.30 \\ &= \mathbf{ab}_x a \cdot (\omega \otimes \mathbf{id}) \cdot b && \\ &= (x)a \cdot b && \end{aligned}$$

(*)
$$\begin{aligned} (x)(a \otimes \mathbf{id}) &= (x)([x/x]a \otimes \mathbf{id}) && \sigma_1 \\ &= (x)(((\langle x\rangle \otimes \mathbf{id}) \cdot (x)a \otimes \mathbf{id}) && \\ &= (x)((\langle x\rangle \otimes \mathbf{id} \otimes \mathbf{id}) \cdot ((x)a \otimes \mathbf{id}) && (1) \\ &= (x)a \otimes \mathbf{id} && 2.32 \end{aligned}$$

(2)
$$\begin{aligned} (x)(a \otimes b) &= (x)((a \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes b)) && \\ &= (x)(a \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes b) && (1), 2.26(1, 4) \\ &= ((x)a \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes b) && (*) \\ &= (x)a \otimes b && \end{aligned}$$

(†)
$$\begin{aligned} (x)(\mathbf{id}_m \otimes a) &= (x)(\mathbf{id}_m \otimes a) \cdot (b \otimes \mathbf{id}) && (1) \\ &= (x)(\mathbf{c}_{m,k} \cdot (a \otimes \mathbf{id}_m) \cdot \mathbf{c}_{l,m}) && 2.24(1), a : k \to l \\ &= \mathbf{ab}_x \mathbf{c}_{m,k} \cdot (x)(a \otimes \mathbf{id}_m) \cdot \mathbf{c}_{l,m} && (1) \\ &= (\mathbf{id}_p \otimes \mathbf{c}_{m,k}) \cdot (x)(a \otimes \mathbf{id}_m) \cdot \mathbf{c}_{l,m} && S4 \\ &= (\mathbf{id}_p \otimes \mathbf{c}_{m,k}) \cdot ((x)a \otimes \mathbf{id}_m) \cdot \mathbf{c}_{l,m} && (2) \\ &= (\mathbf{id}_p \otimes \mathbf{c}_{m,k}) \cdot \mathbf{c}_{p\otimes k,m} \cdot (\mathbf{id}_m \otimes (x)a) \cdot \mathbf{c}_{m,l} \cdot \mathbf{c}_{l,m} && 2.24(1) \\ &= (\mathbf{id}_p \otimes \mathbf{c}_{m,k}) \cdot \mathbf{c}_{p\otimes k,m} \cdot (\mathbf{id}_m \otimes (x)a) && S2 \\ &= (\mathbf{id}_p \otimes \mathbf{c}_{m,k}) \cdot (\mathbf{id}_p \otimes \mathbf{c}_{k,m}) && \\ &\qquad \cdot (\mathbf{c}_{p,m} \otimes \mathbf{id}) \cdot (\mathbf{id}_m \otimes (x)a) && 2.24(3) \\ &= (\mathbf{c}_{p,m} \otimes \mathbf{id}) \cdot (\mathbf{id}_m \otimes (x)a) && S_2 \end{aligned}$$

$$
\begin{array}{llll}
(3) & (x)(b \otimes a) & = (x)((\mathbf{id}_m \otimes a) \cdot (b \otimes \mathbf{id})) & \\
& & = (x)(\mathbf{id}_m \otimes a) \cdot (b \otimes \mathbf{id}) & (1) \\
& & = (\mathbf{c}_{p,m} \otimes \mathbf{id}) \cdot (\mathbf{id}_m \otimes (x)a) \cdot (b \otimes \mathbf{id}) & (\dagger) \\
& & = (\mathbf{c}_{p,m} \otimes \mathbf{id}) \cdot (b \otimes (x)a) &
\end{array}
$$

$$
\begin{array}{llll}
(\boldsymbol{\alpha}) & (x)[x/y]b & = (x)((\langle x \rangle \otimes \mathbf{id}) \cdot (y)b) & \\
& & = (x)(\langle x \rangle \otimes \mathbf{id}) \cdot (y)b & (1) \\
& & = (y)b & 2.32
\end{array}
$$

$$
\begin{array}{llll}
(4) & \mathbf{ab}_x a & = \mathbf{ab}_x a \cdot (x)(\langle x \rangle \otimes \mathbf{id}) & 2.32 \\
& & = (x)(a \cdot (\langle x \rangle \otimes \mathbf{id})) & \\
& & = (x)(\langle x \rangle \otimes a) &
\end{array}
$$

(5) Induction on length of $\vec{x}$. Basis true by definition. Step:

$$
\begin{array}{lll}
(x\vec{y})(\langle x\vec{y} \rangle \otimes \mathbf{id}) & = (x)(\vec{y})((\langle \vec{y} \rangle \otimes \mathbf{id}) \cdot (\langle x \rangle \otimes \mathbf{id})) & \\
& = (x)((\vec{y})(\langle \vec{y} \rangle \otimes \mathbf{id})) \cdot (\langle x \rangle \otimes \mathbf{id}) & (1)^* \\
& = (x)(\langle x \rangle \otimes \mathbf{id}) & \text{induction} \\
& = \mathbf{id} & 2.32
\end{array}
$$

$$
\begin{array}{llll}
(6) & (x)(y)a & = \mathbf{ab}_x(\mathbf{ab}_y a \cdot (\boldsymbol{\omega} \otimes \mathbf{id})) \cdot (\boldsymbol{\omega} \otimes \mathbf{id}) & \\
& & = \mathbf{ab}_x\mathbf{ab}_y a \cdot \mathbf{ab}_x(\boldsymbol{\omega} \otimes \mathbf{id}) \cdot (\boldsymbol{\omega} \otimes \mathbf{id}) & \\
& & = \mathbf{ab}_x\mathbf{ab}_y a \cdot (\mathbf{id}_p \otimes \boldsymbol{\omega} \otimes \mathbf{id}) \cdot (\boldsymbol{\omega} \otimes \mathbf{id}) & \mathrm{S}_6,\ \gamma_2 \\
& & = (\mathbf{c}_{p,q} \otimes \mathbf{id}) \cdot \mathbf{ab}_y\mathbf{ab}_x a \cdot (\mathbf{c}_{q,p} \otimes \mathbf{id}) & \\
& & \qquad \cdot (\mathbf{id}_p \otimes \boldsymbol{\omega} \otimes \mathbf{id}) \cdot (\boldsymbol{\omega} \otimes \mathbf{id}) & 2.24(2) \\
& & = (\mathbf{c}_{p,q} \otimes \mathbf{id}) \cdot \mathbf{ab}_y\mathbf{ab}_x a \cdot (\boldsymbol{\omega} \otimes \mathbf{id}) \cdot (\boldsymbol{\omega} \otimes \mathbf{id}) & \mathrm{S}_1, 2.24(4) \\
& & = (\mathbf{c}_{p,q} \otimes \mathbf{id}) \cdot (y)\mathbf{ab}_x a \cdot (\boldsymbol{\omega} \otimes \mathbf{id}) & \\
& & = (\mathbf{c}_{p,q} \otimes \mathbf{id}) \cdot (y)(\mathbf{ab}_x a \cdot (\boldsymbol{\omega} \otimes \mathbf{id})) & 2.33(1) \\
& & = (\mathbf{c}_{p,q} \otimes \mathbf{id}) \cdot (y)(x)a &
\end{array}
$$

■

Remark When $x \notin \mathbf{surf}(b)$, $\mathbf{ab}_y b = \mathbf{ab}_x[x/y]b$ follows from $\boldsymbol{\alpha}$, (4) and proposition 2.36(5).

Proposition 2.34 *Define* $\mathbf{p}_{m,n}$ *as* $(\vec{x}\vec{y})\langle\vec{y}\vec{x}\rangle$, *where* $\vec{x} : m$ *and* $\vec{y} : n$. *Then* $\mathbf{p}_{m,n} = \mathbf{c}_{m,n}$.

Proof

$$\begin{array}{rcll}
(\vec{x}\vec{y})\langle\vec{y}\vec{x}\rangle & = & (\vec{x})((\vec{y})\langle\vec{y}\rangle \otimes \langle\vec{x}\rangle) & 2.33(2)^* \\
& = & (\vec{x})(\mathbf{id} \otimes \langle\vec{x}\rangle) & 2.33(5) \\
& = & (\vec{x})((\langle\vec{x}\rangle \otimes \mathbf{id}) \cdot \mathbf{c}_{m,n}) & \mathrm{S}_1 \\
& = & (\vec{x})(\langle\vec{x}\rangle \otimes \mathbf{id}) \cdot \mathbf{c}_{m,n} & 2.26(2), 2.33(1)^* \\
& = & \mathbf{c}_{m,n} & 2.33(5)
\end{array}$$

■

Corollary 2.35 (ζ) *Assume* $a : m \to n$ *and* $b : k \to \ell$. *Then* $\mathbf{p}_{k,m} \cdot (a \otimes b) = (b \otimes a) \cdot \mathbf{p}_{\ell,n}$.

Proof Immediate by proposition 2.34 and naturality of symmetries (S_1). ■

The following proposition asserts that the semantic substitution $[y/x]$ behaves equationally like the syntactic substitution $\{y/x\}$ as given in definition 2.12.

Proposition 2.36 *The following properties of semantic substitution hold in any control structure:*

1. $[y/x]\mathbf{id} = \mathbf{id}$
2. $[y/x]\boldsymbol{\omega} = \boldsymbol{\omega}$
3. $[y/x]\langle z\rangle = \langle z\rangle \quad (z \neq x)$
4. $[y/x]\langle x\rangle = \langle y\rangle$
5. $[y/x](a \otimes b) = [y/x]a \otimes [y/x]b$
6. $[y/x](a \cdot b) = [y/x]a \cdot [y/x]b$
7. $[y/x]K(a, \ldots) = K([y/x]a, \ldots)$
8. $[y/x]\mathbf{ab}_z a = \mathbf{ab}_z[y/x]a \qquad (z \notin \{x, y\})$
9. $[z/x]\mathbf{ab}_z a = \mathbf{ab}_w[z/x][w/z]a \quad (z \neq x,\ w \notin \mathbf{surf}(a) \cup \{x, z\})$
10. $[y/z]\mathbf{ab}_z a = \mathbf{ab}_z a$.

Proof (1), (2), (3) and (10) follow directly from proposition 2.30(3), and (4) follows directly from ϵ. (7) is exactly σ_3. For the remaining cases, assume $x, y, z : p$:

(5) $[{}^{y}\!/\!{}_{x}](a \otimes b)$

$$\begin{aligned}
&= [{}^{y}\!/\!{}_{x}]([{}^{x}\!/\!{}_{x}]a \otimes [{}^{x}\!/\!{}_{x}]b) && \sigma_1\\
&= [{}^{y}\!/\!{}_{x}](((\langle x\rangle \otimes \mathbf{id}_k)\cdot(x)a) \otimes ((\langle x\rangle \otimes \mathbf{id}_m)\cdot(x)b)) &&\\
&= [{}^{y}\!/\!{}_{x}]((\langle xx\rangle \otimes \mathbf{id}_{k\otimes m})\cdot(\mathbf{id}_p \otimes \mathbf{c}_{p,k} \otimes \mathbf{id}_m)\cdot((x)a \otimes (x)b)) && \mathrm{S}_1\\
&= ((\langle y\rangle \otimes \mathbf{id}_{k\otimes m})\cdot((x)\langle xx\rangle \otimes \mathbf{id}_{k\otimes m})\cdot &&\\
&\qquad\qquad (\mathbf{id}_p \otimes \mathbf{c}_{p,k} \otimes \mathbf{id}_m)\cdot((x)a \otimes (x)b) && 2.33(1,2)\\
&= ((\langle yy\rangle \otimes \mathbf{id}_{k\otimes m})\cdot(\mathbf{id}_p \otimes \mathbf{c}_{p,k} \otimes \mathbf{id}_m)\cdot((x)a \otimes (x)b) && \sigma_2\\
&= ((\langle y\rangle \otimes \mathbf{id}_k)\cdot(x)a \otimes ((\langle y\rangle \otimes \mathbf{id}_m)\cdot(x)b) && \mathrm{S}_1\\
&= [{}^{y}\!/\!{}_{x}]a \otimes [{}^{y}\!/\!{}_{x}]b\ .
\end{aligned}$$

(6) $[{}^{y}\!/\!{}_{x}](a \cdot b)$

$$\begin{aligned}
&= ((\langle y\rangle \otimes \mathbf{id})\cdot(x)(a\cdot b) &&\\
&= ((\langle y\rangle \otimes \mathbf{id})\cdot(x)((\langle x\rangle \otimes a)\cdot(x)b) && 2.33(4)\\
&= [{}^{y}\!/\!{}_{x}]((\langle x\rangle \otimes a)\cdot(x)b &&\\
&= ((\langle y\rangle \otimes [{}^{y}\!/\!{}_{x}]a)\cdot(x)b && (4),(5)\\
&= [{}^{y}\!/\!{}_{x}]a\cdot((\langle y\rangle \otimes \mathbf{id})\cdot(x)b &&\\
&= [{}^{y}\!/\!{}_{x}]a\cdot[{}^{y}\!/\!{}_{x}]b\ .
\end{aligned}$$

(9) $[{}^{z}\!/\!{}_{x}]\mathbf{ab}_z a$

$$\begin{aligned}
&= ((\langle z\rangle \otimes \mathbf{id})\cdot\mathbf{ab}_x\mathbf{ab}_z a\cdot(\boldsymbol{\omega} \otimes \mathbf{id}) &&\\
&= ((\langle z\rangle \otimes \mathbf{id})\cdot(c \otimes \mathbf{id})\cdot\mathbf{ab}_z\mathbf{ab}_x a\cdot(c \otimes \mathbf{id})\cdot(\boldsymbol{\omega} \otimes \mathbf{id}) && 2.24(2)\\
&= (\mathbf{id} \otimes \langle z\rangle \otimes \mathbf{id})\cdot\mathbf{ab}_z\mathbf{ab}_x a\cdot(\mathbf{id} \otimes \boldsymbol{\omega} \otimes \mathbf{id}) && \mathrm{S}_1\\
&= (\mathbf{id} \otimes \langle z\rangle \otimes \mathbf{id})\cdot(z)((\langle z\rangle \otimes \mathbf{ab}_x a)\cdot(\mathbf{id} \otimes \boldsymbol{\omega} \otimes \mathbf{id}) && 2.33(4)\\
&= (\mathbf{id} \otimes \langle z\rangle \otimes \mathbf{id})\cdot(w)([{}^{w}\!/\!{}_{z}]((\langle z\rangle \otimes \mathbf{ab}_x a))\cdot(\mathbf{id} \otimes \boldsymbol{\omega} \otimes \mathbf{id}) && \boldsymbol{\alpha}\\
&= (\mathbf{id} \otimes \langle z\rangle \otimes \mathbf{id})\cdot(w)((\langle w\rangle \otimes \mathbf{ab}_x[{}^{w}\!/\!{}_{z}]a)\cdot(\mathbf{id} \otimes \boldsymbol{\omega} \otimes \mathbf{id}) && (5),(4),(8)\\
&= (\mathbf{id} \otimes \langle z\rangle \otimes \mathbf{id})\cdot\mathbf{ab}_w\mathbf{ab}_x[{}^{w}\!/\!{}_{z}]a\cdot(\mathbf{id} \otimes \boldsymbol{\omega} \otimes \mathbf{id}) && 2.33(4)\\
&= \mathbf{ab}_w((\langle z\rangle \otimes \mathbf{id})\cdot\mathbf{ab}_w\mathbf{ab}_x[{}^{w}\!/\!{}_{z}]a\cdot\mathbf{ab}_w(\boldsymbol{\omega} \otimes \mathbf{id}) && \gamma_1,\gamma_2\\
&= \mathbf{ab}_w[{}^{z}\!/\!{}_{x}][{}^{w}\!/\!{}_{z}]a\ .
\end{aligned}$$

(8) $[{}^{y}\!/\!{}_{x}]\mathbf{ab}_z a$

$$
\begin{aligned}
&= (\langle y\rangle \otimes \mathbf{id}) \cdot \mathbf{ab}_x\mathbf{ab}_z a \cdot (\omega \otimes \mathbf{id}) && \\
&= (\langle y\rangle \otimes \mathbf{id}) \cdot (c \otimes \mathbf{id}) \cdot \mathbf{ab}_z\mathbf{ab}_x a \cdot (c \otimes \mathbf{id}) \cdot (\omega \otimes \mathbf{id}) && 2.24(2) \\
&= (\mathbf{id} \otimes \langle y\rangle \otimes \mathbf{id}) \cdot \mathbf{ab}_z\mathbf{ab}_x a \cdot (\mathbf{id} \otimes \omega \otimes \mathbf{id}) && \mathrm{S}_1 \\
&= \mathbf{ab}_z((\langle y\rangle \otimes \mathbf{id}) \cdot \mathbf{ab}_z\mathbf{ab}_x a \cdot \mathbf{ab}_z(\omega \otimes \mathbf{id}) && \gamma_1, \gamma_2 \\
&= \mathbf{ab}_z[{}^{y}\!/\!{}_{x}]a\ .
\end{aligned}
$$

■

Discussion There are alternative sets of axioms to the ones given above and which is the most elegant or natural set is arguable. For instance, we can replace σ_2 by $\sigma_2' : \mathbf{ab}_x\langle x\rangle = \mathbf{ab}_y\langle y\rangle$. This equation is provable by proposition 2.33(4) and α-conversion, while σ_2 is provable (from the alternative set of axioms) as follows:

$$
\begin{aligned}
[{}^{y}\!/\!{}_{x}](\langle x\rangle \otimes \langle x\rangle) &= (\langle y\rangle \otimes \mathbf{id}) \cdot (x)(\langle x\rangle \otimes \langle x\rangle) && \\
&= (\langle y\rangle \otimes \mathbf{id}) \cdot (y)(\langle y\rangle \otimes \langle y\rangle) && \sigma_2', 2.33(4) \\
&= [{}^{y}\!/\!{}_{y}](\langle y\rangle \otimes \langle y\rangle) && \\
&= \langle y\rangle \otimes \langle y\rangle && \sigma_1
\end{aligned}
$$

■

Note that $\mathbf{ab}_x\langle x\rangle = \mathbf{ab}_y\langle y\rangle$ would be the only explicit instance of α-conversion in the axioms which is required to derive α-conversion for arbitrary terms. Finally, the axiom can also be replaced by $[{}^{y}\!/\!{}_{x}](\langle x\rangle \otimes \langle x\rangle) = [{}^{y}\!/\!{}_{x}]\langle x\rangle \otimes [{}^{y}\!/\!{}_{x}]\langle x\rangle$, in the presence of ϵ.

Definition 2.37 (The category of control structures) *The category* $\mathbf{CS}^s(\mathcal{K})$ *of control structures over a signature* $\mathcal{K}$ *has as* objects *control structures, and as* morphisms *action structure homomorphisms which act as identity upon* X *and* M *and also preserve the data, discard and control operations.* ■

It is immediate that every morphism in $\mathbf{CS}^s(\mathcal{K})$ reduces surface:

Proposition 2.38 (Surface reduction) *Let* $\Phi : A \to B$ *be any morphism of control structures. Then* $\mathbf{surf}(\Phi a) \subseteq \mathbf{surf}(a)$ *for all* $a \in A$.

Since $\mathbf{CS}^s(\mathcal{K})$ is characterized equationally, it is easy to see that it is closed under factoring by a congruence. Moreover we can state precisely what effect the morphism has upon surface.

Proposition 2.39 (Congruence) *Let $\equiv$ be a congruence over each action-set $A(m,n)$ in a control structure A, i.e. an equivalence which is preserved by the action structure operations, by the control constructions $\mathcal{K}$ and by reaction. Then, the quotient $A/\equiv$ is a control structure, with $\Phi : a \mapsto [a]$ as the induced morphism from A to $A/\equiv$, where $[a]$ is the congruence class of a. Moreover,*

$$\mathbf{surf}([a]) = \bigcap\{\mathbf{surf}(a') \mid a' \in [a]\}$$

Proof The proof is mostly of a kind which is standard in universal algebra. For the last part, we prove each inclusion as follows.

($\subseteq$) It is enough to show that $\mathbf{surf}([a]) \subseteq \mathbf{surf}(a)$ for each a; but this follows from Proposition 2.38.

($\supseteq$) Assuming $x \notin \mathbf{surf}([a])$, it is enough to find $a' \equiv a$ such that $\mathbf{ab}_x a' = \mathbf{id} \otimes a'$. Pick $a' = [y/x]a$, where $y \neq x$; then $a' \equiv a$ follows from $\mathbf{ab}_x a \equiv \mathbf{id} \otimes a$ and the rest follows much as in Proposition 2.30.

$$\begin{aligned}
\mathbf{ab}_x[y/x]a &= \mathbf{ab}_x((\langle y\rangle \otimes \mathbf{id}) \cdot (x)a) \\
&= \mathbf{ab}_x(\langle y\rangle \otimes \mathbf{id}) \cdot \mathbf{ab}_x(x)a \\
&= (\mathbf{id} \otimes \langle y\rangle \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes (x)a) && \mathrm{S}_6,\gamma_1,\mathrm{S}_5,\gamma_2 \\
&= \mathbf{id} \otimes ((\langle y\rangle \otimes \mathbf{id}) \cdot (x)a) \\
&= \mathbf{id} \otimes [y/x]a
\end{aligned}$$

■

We now proceed to consider initiality among control structures. The following has a standard proof, since the axioms are purely algebraic:

Proposition 2.40 *The category $\mathbf{CS}^s(\mathcal{K})$ has an initial object.*

Our next task is to establish the status of action calculi among control structures. The following result depends upon the fact that semantic and syntactic substitution coincide in the theory AC due to $\boldsymbol{\sigma}$.

Proposition 2.41 AC($\mathcal{K}$) *is a control structure over* $\mathcal{K}$ *with the permutations* $\mathbf{p}_{m,n}$ *as the symmetry.*

Proof By proposition 2.27, we already have that AC^s($\mathcal{K}$) is a symmetric action structure. Moreover $\boldsymbol{\epsilon}$ is a special case of $\boldsymbol{\delta}$. By $\boldsymbol{\sigma}$, $\{y/x\}$ agrees with the derived operation $[y/x]$ and $\boldsymbol{\sigma}_2$ and $\boldsymbol{\sigma}_3$ follow from the equations for $\{y/x\}$. By lemma 2.13 we have $\boldsymbol{\sigma}_1$. The following proofs give γ_1 and γ_2.

$$
\begin{array}{llcll}
\gamma_1 & \mathbf{ab}_x\langle y\rangle & = & (x)(\langle x\rangle \otimes \langle y\rangle) & 2.16(4) \\
 & & = & (x)(\langle x\rangle \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes \langle y\rangle) & 2.16(1) \\
 & & = & \mathbf{id} \otimes \langle y\rangle & \boldsymbol{\delta}
\end{array}
$$

$$
\begin{array}{llcll}
\gamma_2 & \mathbf{ab}_x\boldsymbol{\omega} & = & (x)(\langle x\rangle \otimes \boldsymbol{\omega}) & 2.16(4) \\
 & & = & (x)(\langle x\rangle \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes \boldsymbol{\omega}) & 2.16(1) \\
 & & = & \mathbf{id} \otimes \boldsymbol{\omega} & \boldsymbol{\delta}
\end{array}
$$

■

Finally we establish our main result. It depends upon the fact that, in any control structure, semantic substitution $[y/x]$ provably satisfies the equations which define syntactic substitution $\{y/x\}$.

Theorem 2.42 (Initiality) AC^s($\mathcal{K}$) *is initial in* **CS**^{s}($\mathcal{K}$).

Proof Since we have shown that the action calculus is a control structure, there is a unique map to it from the initial control structure. That map is obviously onto, so it remains to show that it is one to one. To do that, we must show that whenever the images of two terms are provably equal in AC, then they are equal in the initial control structure. It suffices to show that in the initial control structure, the axioms of AC are valid. By propositions 2.32 and 2.36, and corollary 2.35 we get $\boldsymbol{\delta}$, $\boldsymbol{\sigma}$ and $\boldsymbol{\zeta}$ respectively. It remains to derive $\boldsymbol{\gamma}$.

By proposition 2.30, it suffices to show that whenever $x \notin \mathbf{fn}(t)$ (where **fn** is defined as previously), $x \notin \mathbf{surf}(t)$. This involves an easy induction on the structure of terms (of the initial control structure): for instance, in the case of $t \equiv K\vec{t}$, $x \notin \mathbf{fn}(t)$ if and only if $x \notin \mathbf{fn}(t')$, for each $t' \in \vec{t}$. By proposition 2.31(4) the result follows immediately. ∎

Remark Note that proposition 2.36(9) holds for any $w \notin \mathbf{surf}(a)$. In general, there may not be any such w; however, in action calculi, there will always be such a w as the surface of any action is finite, by Fact 2.8.

We note that $\mathbf{CS}^s(\mathcal{K})$ contains any control structure over the signature $\mathcal{K}$, with any reaction relation. We often wish to confine attention to those which satisfy a set of rules, hence we define:

Definition 2.43 *If $\mathcal{R}$ is a set of reaction rules over $\mathcal{K}$, then $\mathbf{CS}(\mathcal{K},\mathcal{R})$ is the full subcategory of $\mathbf{CS}^s(\mathcal{K})$ containing just those control structures whose reaction relation satisfies $\mathcal{R}$.* ∎

The following is immediate:

Corollary 2.44 $\mathsf{AC}(\mathcal{K},\mathcal{R})$ *is initial in* $\mathbf{CS}(\mathcal{K},\mathcal{R})$. ∎

When $\mathcal{R}$ is understood, we often write $\mathbf{CS}(\mathcal{K})$ to mean $\mathbf{CS}(\mathcal{K},\mathcal{R})$.

Discussion The initiality of $\mathsf{AC}^s(\mathcal{K})$ is significant largely because it has a direct presentation (up to isomorphism) as the action structure of molecular forms $\mathcal{M}(\mathcal{K})$. The appeal of action calculi as concrete models of concurrent computation depends on the adequacy of the molecular forms as concrete representations of concurrent reactive systems. Evidence in favour is the fact that known concrete models fit readily into the framework. However, this does not necessarily justify every choice made in the formulation of the molecular forms: in other words, there may be variations on molecular forms and consequently in the formulation of action calculi and control structures which would still do the job.

It is still too early to decide which is the best notion of molecular form (indeed, we must first generate competing variations) and in the meantime we can only appeal to the elegance and simplicity of the molecular forms we have presented. In the following chapter we shall explore a natural variation which allows a tractable labelled transition semantics to be developed for a descendant of the π-calculus.

It is worth reflecting on the kind of applications our formulation of control structures can support. We have already noted that the category of control structures is closed under congruences and therefore any model of an action calculus obtained by quotient with a congruence gives a control structure. The homomorphism from the action calculus to such models will be onto; there are interesting control structures to which the initial morphism may not be onto. One such kind of morphism represents the notion of encoding or implementation. For instance, the actions of an action calculus $\mathsf{AC}(\mathcal{K})$ may be encodable as actions of another $\mathsf{AC}(\mathcal{K}')$. If the encoding is compositional, then it may be represented as a morphism in the category of control structures over $\mathcal{K}$; indeed $\mathsf{AC}(\mathcal{K}')$ itself can be shown to be an object in the category $\mathbf{CS}(\mathcal{K})$.

Another useful application of control structures concerns the classification of dynamics. Since morphisms of control structures preserve reaction, the existence of a morphism from an action calculus $\mathsf{AC}(\mathcal{K})$ to some control structure indicates some constraints on the reaction relation of $\mathsf{AC}(\mathcal{K})$. One way of classifying reaction rules is through such control structures; each such *classifier* C determines for which sets of reaction rules $\mathcal{R}$ a morphism from $\mathsf{AC}(\mathcal{K}, \mathcal{R})$ to C exists. In chapter 4 we shall see two examples of such classifiers.

Chapter 3

Reflexive Control Structures

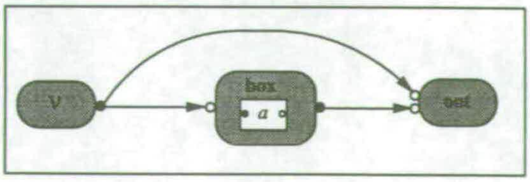
In the previous chapter, a refinement of action structures was developed to give a space of models for a concrete representation of a class of action structures given by the molecular forms. These molecular forms are essentially linear syntactic descriptions of directed acyclic graphs whose nodes consist of syntactic constructions called controls, together with a facility for handling names through binding and substitution.

An illustration of this will suffice for our purposes. The construction shown below, is a molecular form of the term $\boldsymbol{\nu} \cdot (x)\langle xx\rangle \cdot (\mathbf{id}_p \otimes \mathbf{box}t) \cdot \mathbf{out}$ in PIC, whose signature was encountered in Chapter 2:

$$[\boldsymbol{\nu}(x), \langle x\rangle \mathbf{box}a(y), \langle xy\rangle \mathbf{out}(\ )]$$

where $a : q \to q$ is the molecular form of t.

In a directed graph representation, binding occurrences stand for sources of edges, whose destinations are identified by the bound occurrences, as shown below by the diagrammatic representation of the above molecular form:



In thinking of molecular forms as graphs, it is helpful to consider the directed edges as channels through which names "flow". A little reflection reveals that the binding structure of the molecular forms of action calculi imposes certain constraints on which kind of directed graphs (and hence, dataflow configurations) are expressible: for instance, while a channel can be "split" through copying (as in the case of the channel identified by x in the above example), it is not possible to "join" or "merge" two dataflow channels into a single channel. Also, all dataflow proceeds in one direction; as illustrated by *action graphs*, a graphical representation of actions as an enhanced form of directed acyclic graphs [29]. The molecular forms which gave rise to control structures, convincing as they are by virtue of their elegant accomodation of existing concrete computational models, should not be taken as the sole form that can provide such accomodation. A natural variation, suggested by the constraint on dataflow in the molecular forms encountered hitherto, is to remove such; in other words, to move from acyclic graphs to cyclic ones.

Such cyclicity can be achieved by a suitable variation in the directionality of binding in the molecular forms. As they stand, binding in the molecular forms is to the right and hence a molecule μ which is bound by some molecule λ to its left, cannot itself bind λ. Moreover, there is no way in which the exported names of an action can be fed into an action which is precomposed to it. This form of backward dataflow is generally recognised under the term *feedback*. In this chapter we shall study such an operation, here called *reflexion*, introduced by Milner and Jensen in [25] giving a refinement of action calculi called *reflexive action calculi*.

The feedback operator that we shall study was discovered independently by several researchers working in quite dissimilar contexts. Stefănescu studied the feedback operator in the context of flow charts [39]; Bloom and Esik treat feedback in the context of iteration theories [5]; Milner first discovered reflexion (feedback) in the context of an action structure for the π-calculus [26] and then studied it in the context of action calculi in [25]; while Joyal, Street and Verity treat feedback (which they call *trace*) in the setting of (a mild generalisation of) strict symmetric monoidal categories [14].

There are several reasons which make the introduction of reflexion as a structural operation interesting in the context of action calculi. First, as argued above, the restriction on dataflow between actions to just the forward direction is effectively removed. The bodies in the molecular forms for reflexive action calculi are representable, as a result of reflexion, by multisets of molecules, rather than partial sequences. This is a manifestation of the freedom to express dataflow in any direction. Also, it makes the resulting molecular forms closer to Berry and Boudol's Chemical Abstract Machine (CHAM) [3]: the *solution* of a CHAM consists of a *multiset* of *molecules*. As an additional benefit, the restriction operation $\boldsymbol{\nu}$, present as a control operation in the action calculi for both Petri nets and the π-calculus, is derivable in terms of reflexion and copying $((x)\langle xx\rangle)$. Moreover, reflexion can also be used, in the presence of higher order controls $\{\ulcorner\ \urcorner, \mathbf{ap}\}$, to derive a form of recursion. Finally, as will be discussed in greater depth in chapter 5, the presence of reflexion will be crucial to obtaining an elegant operational semantics of (a reflexive variant of) the π-calculus based upon labelled transition relations.

Outline The presentation of reflexive action calculi in Section 3.1 is essentially a summary of [25]. In this section we review *reflexive molecular forms* and define the operations of control structures upon them. The reflexion operation is then defined, through the auxiliary notion of reflexive substitution on these molecular forms. As for action calculi, a term algebra presentation is given and shown to be isomorphic to the reflexive molecular forms. This term algebra is essentially that for action calculi with the inclusion of the reflexion operation together with equations which effectively constrain its interaction with the other operations. Further to this summary of [25], we develop an example of the use of reflexion to derive recursion in the presence of higher order controls. A further variation of the reflexive molecular forms—giving *strict* reflexive action calculi—is then briefly described.

In the following section we present a refinement of control structures which gives a category of models for reflexive action calculi. This is done through the intermediate notion of a trace on a strict monoidal category, introduced by Joyal, Street and Verity in [14]. The abstract treatment of reflexion allows us to deal semantically with the derived restriction operation $\boldsymbol{\nu}$. In particular, we explore the effect of restriction on the surface of an action. Extending the abstract treatment to the strict variant also leads to a characterisation of surface which captures the intuition that the surface of an action consists of those names that, when "hidden" by restriction, affect the behaviour—hence, the semantic interpretation—of that action.

3.1 Reflexive Action Calculi

We shall begin by presenting the reflexive variant of the molecular forms mentioned above:

Definition 3.1 (Reflexive Molecular Forms) *Let $\mathcal{K}$ be a signature and, for every prime arity p let $\boldsymbol{\nu} : \epsilon \to p$ be a control not in $\mathcal{K}$. The* reflexive molecular forms *over $\mathcal{K}$, denoted $\mathcal{M}^{\mathsf{r}}(\mathcal{K})$, consist of the actions, given by*

$$\begin{array}{lcll} a & ::= & (\vec{x})\, \mu_1 \cdots \mu_r\, \langle \vec{u} \rangle & (\vec{x} : m, \vec{u} : n, a : m \to n) \\ \mu & ::= & \langle \vec{v} \rangle\, K\vec{b}\, (\vec{y}) & (\vec{v} : k, \vec{y} : l, K\vec{b} : k \to l) \end{array}$$

where μ ranges over molecules and K ranges over $\mathcal{K} \cup \{\boldsymbol{\nu}_p \mid p \in P\}$. The body of a is a multiset of molecules where any two molecules can commute. For each molecule $\langle \vec{v} \rangle\, K\vec{b}\, (\vec{y})$ the binding occurrences $\vec{y}$ have scope throughout the action a. In the action a the binding occurrences in each molecule and the names in $\vec{x}$ must all be distinct. Actions which differ only by a change of bound names are not distinguished. ■

We shall now define reflexive substitution, which ensures that channels which loop upon themselves are detected and duely give rise to a restriction particle in the molecular form.

Definition 3.2 (Reflexive substitution) *Let x be a name not bound in a. Then* reflexive substitution *$^*\{y/x\}$ on actions is defined as follows:*

$$^*\{y/x\}a \stackrel{\text{def}}{=} \begin{cases} {}^{y}\!/\!{}_{x}a & (x \neq y) \\ (\boldsymbol{\nu} x)a & (x = y) \end{cases}$$

where ${}^{y/x}a$ *denotes the* literal replacement *of* y *for* x *in the syntactic form of* a *and* $(\nu x)a$ *denotes the introduction of the molecule* $\langle\rangle\nu(x)$ *in* a. ■

Reflexive substitution now allows us to define our feedback operator:

Definition 3.3 *Let* $a = (x\vec{u})\vec{\mu}\langle y\vec{v}\rangle$ *with* $x, y : p$. *The operation of* reflexion *on reflexive molecular forms is defined as follows*

$$\uparrow^{\mathcal{M}} a \stackrel{\text{def}}{=} {}^{*}\{{}^{y}/_{x}\}(\vec{u})\vec{\mu}\langle\vec{v}\rangle$$ ■

We shall often use a derived form of reflexion which operates on channels of arbitrary (rather than prime) arity. As reflexion on a link of prime arity is defined in terms of reflexive substitution of a single name, we will also wish to relate the derived form with an appropriate version of reflexive substitution:

Definition 3.4 *The* iterated reflexion *operator* $\uparrow^{\mathcal{M}}_{(m)}$, *for* $m = p_1 \otimes \cdots \otimes p_r$, *is given by*

$$\uparrow^{\mathcal{M}}_{(m)} a \stackrel{\text{def}}{=} \uparrow^{\mathcal{M}}_{p_r} \cdots \uparrow^{\mathcal{M}}_{p_1} a$$

Note that, if $r = 0$ *then* $m = \epsilon$ *and* $\uparrow^{\mathcal{M}}_{(m)} a = a$.

The simultaneous reflexive substitution ${}^{*}\{{}^{\vec{y}}/_{\vec{z}}\}$ *is given recursively in terms of the single form by*

$$\begin{aligned} {}^{*}\{\ \}a &\stackrel{\text{def}}{=} a \\ {}^{*}\{{}^{z}/_{w}, {}^{\vec{y}}/_{\vec{x}}\}a &\stackrel{\text{def}}{=} {}^{*}\{{}^{z/w}\vec{y}/_{\vec{x}}\}{}^{*}\{{}^{z}/_{w}\}a \end{aligned}$$

■

Proposition 3.5

1. *The reflexive substitution* ${}^{*}\{{}^{\vec{y}}/_{\vec{z}}\}a$ *is unaffected (up to alphaconversion and permutation of molecules) by permutation of the substitution elements* ${}^{y_i}/_{z_i}$;

2. *If* $\vec{y}, \vec{z} : m$ *and* $a = (\vec{z}\vec{x})\vec{\mu}\langle\vec{y}\vec{u}\rangle$ *then* $\uparrow^{\mathcal{M}}_{(m)} a = {}^{*}\{{}^{\vec{y}}/_{\vec{z}}\}(\vec{x})\vec{\mu}\langle\vec{u}\rangle$;

Proof See [25]. ■

There is also a presentation of reflexive action calculi as term algebras over a set of controls: the main result will be that the two presentations are isomorphic.

Definition 3.6 (Terms) *The* terms *over signature $\mathcal{K}$, denoted by $\mathcal{T}^{\uparrow}(\mathcal{K})$, are generated as follows (where t ranges over terms):*

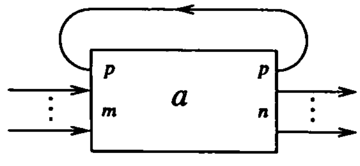
$$t ::= \mathbf{id} \mid \langle x \rangle \mid \omega \mid K\vec{t} \mid t_1 \cdot t_2 \mid t_1 \otimes t_2 \mid \mathbf{ab}_x t \mid \uparrow_p t$$

where each construction has arities dictated by the arity rules of the constructors including the following for $\uparrow_p$:

$$\uparrow\frac{t : p \otimes m \to p \otimes n}{\uparrow_p t : m \to n}$$

The notions of free name, bound name *and* substitution *are as before, with $\{y/x\}\uparrow_p t = \uparrow_p \{y/x\} t$.* ■

It is helpful to view the graphical representation of reflexion. Let t denote the action graph (or molecular form) $a : p \otimes m \to p \otimes n$. Then $\uparrow_p t$ denotes the following action graph:



Such graphic representation may greatly clarify the constructions and manipulations on reflexive terms. Note that the inclusion of action graphs here is informal and is used only to assist intuition. Nevertheless, the reader is encouraged to relate results and manipulations involving complex terms with their graphical representations.

Definition 3.7 (The theory AC^{r}) *The* equational theory AC^{r} *is the set of equations upon terms generated by the equations of* AC *together with the following:*

$$\begin{aligned}
\rho_1 &: \ \mathbf{id}_p = \uparrow_p \mathbf{p}_{p,p} \\
\rho_2 &: \ \uparrow_p t \otimes \mathbf{id} = \uparrow_p (t \otimes \mathbf{id}) \\
\rho_3 &: \ \uparrow_p t_1 \cdot t_2 = \uparrow_p (t_1 \cdot (\mathbf{id}_p \otimes t_2)) \\
\rho_4 &: \ t_1 \cdot \uparrow_p t_2 = \uparrow_p ((\mathbf{id}_p \otimes t_1) \cdot t_2) \\
\rho_6 &: \ \uparrow_q \uparrow_p t = \uparrow_p \uparrow_q ((\mathbf{p}_{q,p} \otimes \mathbf{id}) \cdot t \cdot (\mathbf{p}_{p,q} \otimes \mathbf{id}))
\end{aligned}$$

■

As for AC, we shall consider AC^{r} to be either the above set of axioms, or the set of equations inferred from them (a congruence relation). It will be clear from the context which we mean.

Remark The attentive reader will notice the absence of any axiom labelled ρ_5. In Milner's formulation of reflexive action calculi, there was such an axiom

$$\rho_5 : \ (x)\uparrow_p t = \uparrow_p ((\mathbf{p}_{p,q} \otimes \mathbf{id}) \cdot (x)t)$$

where $x : q$. This axiom was subsequently found to be redundant by Masahito Hasegawa. His proof is reproduced below.

Proposition 3.8 *In* AC^{r}, $(x)\uparrow_p t = \uparrow_p ((\mathbf{p}_{p,q} \otimes \mathbf{id}) \cdot (x)t)$ *where* $x : q$.

Proof

$$\begin{aligned}
(x)\uparrow_p t &= (x)(\uparrow_p (((\langle x \rangle \otimes \mathbf{id}) \cdot (x)t)) && \sigma \\
&= (x)(\uparrow_p ((\mathbf{id}_p \otimes \langle x \rangle \otimes \mathbf{id}) \cdot (\mathbf{p}_{p,q} \otimes \mathbf{id}) \cdot (x)t)) && \zeta \\
&= (x)(((\langle x \rangle \otimes \mathbf{id}) \cdot \uparrow_p ((\mathbf{p}_{p,q} \otimes \mathbf{id}) \cdot (x)t)) && \rho_4 \\
&= (x)((\langle x \rangle \otimes \mathbf{id}) \cdot \uparrow_p ((\mathbf{p}_{p,q} \otimes \mathbf{id}) \cdot (x)t) && 2.16(1) \\
&= \uparrow_p ((\mathbf{p}_{p,q} \otimes \mathbf{id}) \cdot (x)t) && \delta
\end{aligned}$$

■

The following equations, which are counterparts to the axioms for reflexion, are provable in $\mathsf{AC}^{\mathsf{r}}(\mathcal{K})$ for the derived form of reflexion already encountered in the molecular form setting:

Lemma 3.9 *Let* $\uparrow_{(m)} t \stackrel{\text{def}}{=} \uparrow_{p_r} \cdots \uparrow_{p_1} t$ *for* $m = p_1 \otimes \cdots \otimes p_r$, *with* $\uparrow_{(\epsilon)} t = t$*:*

1. $\uparrow_{(m)} \mathbf{p}_{m,m} = \mathbf{id}_m$;
2. $\uparrow_{(m)} t_1 \otimes t_2 = \uparrow_{(m)} (t_1 \otimes t_2)$;
3. $t_1 \otimes \uparrow_{(m)} t_2 = \uparrow_{(m)} ((\mathbf{id}_m \otimes t_1) \cdot t_2)$;
4. $\uparrow_{(m)} t_1 \cdot t_2 = \uparrow_{(m)} (t_1 \cdot (\mathbf{id}_m \otimes t_2)$;
5. $t_1 \cdot \uparrow_{(m)} t_2 = \uparrow_{(m)} ((\mathbf{id}_m \otimes t_1) \cdot t_2)$;
6. $\uparrow_{(n)} \uparrow_{(m)} t = \uparrow_{(m)} \uparrow_{(n)} ((\mathbf{p}_{n,m} \otimes \mathbf{id}) \cdot t \cdot (\mathbf{p}_{m,n} \otimes \mathbf{id}))$

Proof See [25]. ■

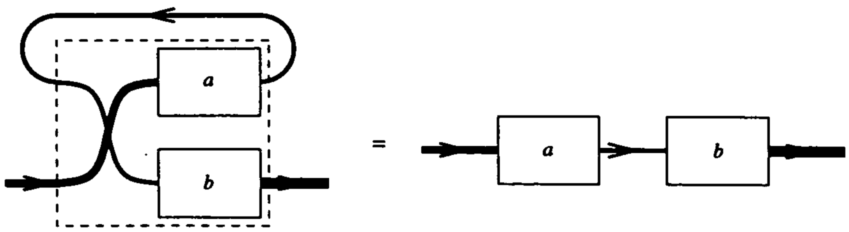
The following lemma shows how, in the presence of reflexion, the composition of two actions can be expressed in terms of their tensor product, composition by permutors and reflexion.

Lemma 3.10

1. $t_1 \cdot t_2 = \uparrow_{(m)} (\mathbf{p}_{m,k} \cdot (t_1 \otimes t_2))$, *if* $t_1 : k \rightarrow m, t_2 : m \rightarrow n$;
2. $(t_1 \otimes \mathbf{id}_k) \cdot t_2 = \uparrow_{(m)} (t_1 \otimes t_2)$, *if* $t_1 : \epsilon \rightarrow m, t_2 : m \otimes k \rightarrow n$;
3. $t_1 \cdot (t_2 \otimes \mathbf{id}_n) = \uparrow_{(m)} (t_1 \otimes t_2)$, *if* $t_1 : k \rightarrow m \otimes n, t_2 : m \rightarrow \epsilon$.

Proof See [25]. ■

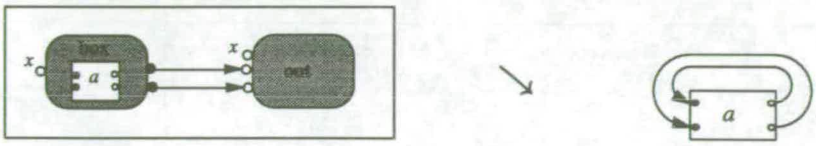
Lemma 3.10(1) states the equality of the action graphs shown below, where the terms t_1 and t_2 denote graphs a and b respectively.



Example By the above lemma, any composition of terms can be rewritten with reference to their tensor product. This has an interesting consequence in PIC^r, the reflexive counterpart of PIC, given as the reflexive action calculus over the same signature (less $\boldsymbol{\nu}$, which is derivable) and reaction rules as PIC. In PIC^r, unlike in PIC, the following reaction is derivable:

$$\begin{array}{rcll} \mathbf{box}_x a \cdot \mathbf{out}_x & = & \uparrow_{(m)}(\mathbf{box}_x a \otimes \mathbf{out}_x) & 3.10(2) \\ & \searrow & \uparrow_{(m)} a & \end{array}$$

A graphical representation of the reaction is included below:



Reflexion can express cyclic dataflow with an action a feeding b while b feeds a. As the following lemma states, this may be written with either the term representing a precomposed to b or vice versa (for an illustration see figure 3–1 on page 66):

Lemma 3.11 (Sliding) *Let* $t_1 : m \rightarrow n$. *Then*

$$\uparrow_{(m)}((t_1 \otimes \mathbf{id}) \cdot t_2) = \uparrow_{(n)}(t_2 \cdot (t_1 \otimes \mathbf{id}))$$

Proof

$$\begin{array}{lll} \uparrow_{(n)}(t_2 \cdot (t_1 \otimes \mathbf{id})) & & \\ \quad = \uparrow_{(n)}(t_2 \cdot ((\uparrow_{(m)}\mathbf{p}_{m,m} \cdot t_1) \otimes \mathbf{id})) & & \rho_1 \\ \quad = \uparrow_{(n)}(t_2 \cdot (\uparrow_{(m)}(\mathbf{p}_{m,m} \cdot (\mathbf{id}_m \otimes t_1)) \otimes \mathbf{id})) & & \rho_3 \\ \quad = \uparrow_{(n)}(t_2 \cdot \uparrow_{(m)}((\mathbf{p}_{m,m} \cdot (\mathbf{id}_m \otimes t_1)) \otimes \mathbf{id})) & & \rho_2 \\ \quad = \uparrow_{(n)}\uparrow_{(m)}((\mathbf{id}_m \otimes t_2) \cdot ((\mathbf{p}_{m,m} \cdot (\mathbf{id}_m \otimes t_1)) \otimes \mathbf{id})) & & \rho_4 \\ \quad = \uparrow_{(n)}\uparrow_{(m)}((\mathbf{id}_m \otimes t_2) \cdot ((t_1 \otimes \mathbf{id}_m) \cdot \mathbf{p}_{n,m}) \otimes \mathbf{id})) & & 2.27,\mathrm{S}_1 \end{array}$$

$$
\begin{aligned}
&= \uparrow_{(n)}\uparrow_{(m)}((\mathbf{id}_m \otimes t_2)\cdot(t_1\otimes \mathbf{id})\cdot(\mathbf{p}_{n,m}\otimes \mathbf{id})) \\
&= \uparrow_{(n)}\uparrow_{(m)}((t_1 \otimes t_2)\cdot(\mathbf{p}_{n,m}\otimes \mathbf{id})) \\
&= \uparrow_{(m)}\uparrow_{(n)}((\mathbf{p}_{n,m}\otimes \mathbf{id})\cdot(t_1 \otimes t_2)\cdot \\
&\qquad\qquad (\mathbf{p}_{n,m}\otimes \mathbf{id})\cdot(\mathbf{p}_{m,n}\otimes \mathbf{id})) && \rho_6 \\
&= \uparrow_{(m)}\uparrow_{(n)}((\mathbf{p}_{n,m}\otimes \mathbf{id})\cdot(t_1 \otimes t_2)) && 2.27,\mathrm{S}_2 \\
&= \uparrow_{(m)}\uparrow_{(n)}((\mathbf{p}_{n,m}\otimes \mathbf{id})\cdot(t_1 \otimes \mathbf{id})\cdot(\mathbf{id}_n\otimes t_2)) \\
&= \uparrow_{(m)}\uparrow_{(n)}(((\mathbf{p}_{n,m}\cdot(t_1 \otimes \mathbf{id}))\otimes \mathbf{id})\cdot(\mathbf{id}_n\otimes t_2)) \\
&= \uparrow_{(m)}\uparrow_{(n)}((((\mathbf{id}_n\otimes t_1)\cdot \mathbf{p}_{n,n})\otimes \mathbf{id})\cdot(\mathbf{id}_n\otimes t_2)) && 2.27,\mathrm{S}_1 \\
&= \uparrow_{(m)}(\uparrow_{(n)}(((\mathbf{id}_n\otimes t_1)\cdot \mathbf{p}_{n,n})\otimes \mathbf{id})\cdot t_2) && \rho_3 \\
&= \uparrow_{(m)}((\uparrow_{(n)}((\mathbf{id}_n\otimes t_1)\cdot \mathbf{p}_{n,n})\otimes \mathbf{id})\cdot t_2) && \rho_2 \\
&= \uparrow_{(m)}(((t_1\cdot \uparrow_{(n)}\mathbf{p}_{n,n})\otimes \mathbf{id})\cdot t_2) && \rho_4 \\
&= \uparrow_{(m)}((t_1\otimes \mathbf{id})\cdot t_2) && \rho_1
\end{aligned}
$$

■

We shall now define reflexive action calculi in a straightforward manner:

Definition 3.12 (Reflexive action calculus: statics) *The* static reflexive action calculus $\mathsf{AC}^{rs}(\mathcal{K})$ *is defined to be the quotient* $\mathcal{T}^{r}(\mathcal{K})/\mathsf{AC}^{r}$. ■

Theorem 3.13 *For any signature $\mathcal{K}$, the reflexive action calculus $\mathsf{AC}^{r}(\mathcal{K})$ is isomorphic to the molecular forms $\mathcal{M}^{r}(\mathcal{K})$.*

Proof See [25]. ■

The isomorphism between $\mathsf{AC}^{r}(\mathcal{K})$ and $\mathcal{M}^{r}(\mathcal{K})$ is given by the map $[\![-]\!] : \mathsf{AC}^{r}(\mathcal{K}) \to \mathcal{M}^{r}(\mathcal{K})$ with inverse $\widehat{(-)} : \mathcal{M}^{r}(\mathcal{K}) \to \mathsf{AC}^{r}(\mathcal{K})$. Both maps were shown in [25] to preserve the control structure operations together with reflexion. Thus, $[\![-]\!]$ is obtained by defining the map inductively on the structure of terms with each term constructor mapped to the corresponding operation on the molecular forms: to demonstrate that $[\![-]\!]$ is well defined it was shown that whenever $\mathsf{AC}^{r} \vdash t_1 = t_2$ then $[\![t_1]\!] = [\![t_2]\!]$. The definition of $\widehat{(-)}$ is less obvious and we reproduce it below as it gives some insight into relationship between reflexion and the scope of binding (to both left and right) in the molecular forms. Let

$$a = (\vec{x})[\langle\vec{v}_1\rangle K_1\vec{c}_1(\vec{y}_1), \dots, \langle\vec{v}_r\rangle K_r\vec{c}_r(\vec{y}_r)]\langle\vec{u}\rangle$$

where $K_i \in \mathcal{K} \cup \{\nu\}$ and $\vec{y}_i : k_i$ $(1 \leq i \leq r)$. Then

$$\widehat{a} \stackrel{\text{def}}{=} \uparrow_{k_r} \cdots \uparrow_{k_1} (\vec{y}_1 \cdots \vec{y}_r \vec{x})(\langle\vec{v}_1\rangle \cdot K_1\widehat{\vec{c}_1} \otimes \cdots \otimes \langle\vec{v}_r\rangle \cdot K_r\widehat{\vec{c}_r} \otimes \langle\vec{u}\rangle)$$

with $\widehat{\nu} \stackrel{\text{def}}{=} \uparrow_p(x)\langle xx\rangle$.

We note that $\mathsf{AC}^{rs}(\mathcal{K})$ together with an arbitrary local preorder on its actions is a control structure over $\mathcal{K}$. Choosing the appropriate local preorder for the reaction rules $\mathcal{R}$ will give us the reflexive action calculus $\mathsf{AC}^{r}(\mathcal{K}, \mathcal{R})$:

Definition 3.14 (Reflexive action calculus: dynamics) *Let $\mathcal{R}$ be a set of reaction rules over a signature $\mathcal{K}$. Then the* (dynamic) reflexive action calculus *$\mathsf{AC}^{r}(\mathcal{K}, \mathcal{R})$ is the control structure given by $\mathsf{AC}^{rs}(\mathcal{K})$ equipped with the smallest reaction relation $\searrow$ which is preserved by reflexion and satisfies the rules $\mathcal{R}$ (for all replacements of metavariables $\vec{a}$ by actions).* ■

As for action calculi, we will write $\mathsf{AC}^{r}(\mathcal{K})$ for $\mathsf{AC}^{r}(\mathcal{K}, \mathcal{R})$ when $\mathcal{R}$ is understood.

3.1.1 Example: recursion from reflexion

To provide an illustration of the use of reflexion we shall present an example of a reflexive action calculus in which two forms of recursion can be defined using reflexion. We shall consider the reflexive action calculus over the signature $\{\ulcorner\ \urcorner, \mathbf{ap}\}$ which has already been encountered (by way of example) in chapter 2.

First we shall define the operator **rec** as a form of reflexion that allows the feedback loop to be tapped.

Definition 3.15 (Recursion) *Let* $t : p \otimes m \rightarrow p \otimes n$. *Then*

$$\mathbf{rec}_p(t) \stackrel{\text{def}}{=} \uparrow_p(t \cdot (\mathbf{copy}_p \otimes \mathbf{id}))$$

where $\mathbf{copy}_p \stackrel{\text{def}}{=} (x)\langle xx \rangle$ *for* $x : p$. ∎

In the following section (Definition 3.27) we will derive restriction $\boldsymbol{\nu}$ as the reflexion of **copy**. Thus, recursing the identity also gives restriction:

Proposition 3.16 $\mathbf{rec}(\mathbf{id}_p) = \boldsymbol{\nu}$.

Proof Immediate. ∎

A more interesting application of **rec**, however, is obtained when reflexion is used to feed a code back into itself:

Proposition 3.17 $\mathbf{rec}(x)\ulcorner t \urcorner \searrow (\mathbf{rec}(x)\ulcorner t \urcorner) \cdot (x)\ulcorner t \urcorner$.

Proof

$$\begin{array}{rcll}
\mathbf{rec}(x)\ulcorner t \urcorner & = & \uparrow((x)\ulcorner t \urcorner \cdot \mathbf{copy}) & \\
& = & \uparrow(x)(\ulcorner t \urcorner \cdot \mathbf{copy}) & 2.16(1) \\
& \searrow & \uparrow(x)(\ulcorner t \urcorner \otimes \ulcorner t \urcorner) & \searrow_\sigma \\
& = & \uparrow(x)(\ulcorner t \urcorner \cdot (\mathbf{id} \otimes \ulcorner t \urcorner)) & \\
& = & \uparrow(\mathbf{ab}_x \ulcorner t \urcorner \cdot (x)(\mathbf{id} \otimes \ulcorner t \urcorner)) & \\
& = & \uparrow(\mathbf{ab}_x \ulcorner t \urcorner \cdot (x)((\ulcorner t \urcorner \otimes \mathbf{id}) \cdot \mathbf{p}_{m \Rightarrow n, m \Rightarrow n})) & \zeta \\
& = & \uparrow(\mathbf{ab}_x \ulcorner t \urcorner \cdot ((x)\ulcorner t \urcorner \otimes \mathbf{id}) \cdot \mathbf{p}_{m \Rightarrow n, m \Rightarrow n}) & 2.16(1,2) \\
& = & \uparrow(\mathbf{ab}_x \ulcorner t \urcorner \cdot \mathbf{p}_{m \Rightarrow n, m \Rightarrow n} \cdot (\mathbf{id} \otimes (x)\ulcorner t \urcorner)) & \zeta \\
& = & \uparrow(\mathbf{ab}_x \ulcorner t \urcorner \cdot \mathbf{p}_{m \Rightarrow n, m \Rightarrow n})(x)\ulcorner t \urcorner & \rho_3 \\
& = & \uparrow(x)((\langle x \rangle \otimes \ulcorner t \urcorner) \cdot \mathbf{p}_{m \Rightarrow n, m \Rightarrow n}) \cdot (x)\ulcorner t \urcorner & 2.16(4,1)
\end{array}$$

$$
\begin{array}{rll}
= & \uparrow(x)(\ulcorner t \urcorner \otimes \langle x \rangle) \cdot (x)\ulcorner t \urcorner & \zeta \\
= & \uparrow(x)((\langle x \rangle \cdot (x)\ulcorner t \urcorner) \otimes \langle x \rangle) \cdot (x)\ulcorner t \urcorner & \sigma_1 \\
= & \uparrow(x)(\langle xx \rangle \cdot ((x)\ulcorner t \urcorner \otimes \mathbf{id})) \cdot (x)\ulcorner t \urcorner & \\
= & \uparrow((x)\langle xx \rangle \cdot ((x)\ulcorner t \urcorner \otimes \mathbf{id})) \cdot (x)\ulcorner t \urcorner & 2.16(1) \\
= & \uparrow((x)\ulcorner t \urcorner \cdot (x)\langle xx \rangle) \cdot (x)\ulcorner t \urcorner & 3.11 \\
= & (\mathbf{rec}(x)\ulcorner t \urcorner) \cdot (x)\ulcorner t \urcorner &
\end{array}
$$

■

Note that the **rec** operator recurses only codes. The following construction allows recursion on arbitrary actions with identical input and output arity.

Definition 3.18 *Let $t : m \rightarrow m$ and $x : m \Rightarrow m$ such that $x \notin \mathbf{fn}(t)$; then*

$$
\begin{array}{rcl}
\mathbf{iter}_f(t) & \stackrel{\text{def}}{=} & (\mathbf{rec}((x)\ulcorner(\langle x \rangle \otimes \mathbf{id}_k) \cdot \mathbf{ap} \cdot t \urcorner) \otimes \mathbf{id}_k) \cdot \mathbf{ap} \\
\mathbf{iter}_b(t) & \stackrel{\text{def}}{=} & ((\mathbf{rec}(x)\ulcorner(\langle x \rangle \otimes t) \cdot \mathbf{ap} \urcorner) \otimes \mathbf{id}_m) \cdot \mathbf{ap}
\end{array}
$$

■

Remark In the above definition for $\mathbf{iter}_f$, the arity of $x : k \Rightarrow m$ and that of $\mathbf{ap} : (k \Rightarrow m) \otimes k \rightarrow m$; while in the definition of $\mathbf{iter}_b$, $x : m \Rightarrow k$ and $\mathbf{ap} : (m \Rightarrow k) \otimes m \rightarrow k$. The arities of the above constructions then obey the following rules:

$$
\frac{t : m \rightarrow m}{\mathbf{iter}_f(t) : k \rightarrow m} \qquad \frac{t : m \rightarrow m}{\mathbf{iter}_f(t) : m \rightarrow k}
$$

Note that k is unconstrained, and therefore, any choice of k will do in the above definitions. This means that there are a family of iteration operators indexed by arities. The semantic relationship between the iterators in each family is an interesting question.

These operators provide left and right recursion as shown below:

Proposition 3.19

1. $\mathbf{iter}_f(t) \searrow \mathbf{iter}_f(t) \cdot t$;

2. $\mathbf{iter}_b(t) \searrow t \cdot \mathbf{iter}_b(t)$.

Proof

(1) $\mathbf{iter}_f(t)$

$$
\begin{array}{lll}
= & (\mathbf{rec}((x)^{\ulcorner}((\langle x\rangle \otimes \mathbf{id}) \cdot \mathbf{ap} \cdot t^{\urcorner}) \otimes \mathbf{id}) \cdot \mathbf{ap} & \\
\searrow & (\mathbf{rec}((x)^{\ulcorner}((\langle x\rangle \otimes \mathbf{id}) \cdot \mathbf{ap} \cdot t^{\urcorner}) \otimes \mathbf{id}) & \\
 & \quad \cdot((x)^{\ulcorner}((\langle x\rangle \otimes \mathbf{id}) \cdot \mathbf{ap} \cdot t^{\urcorner} \otimes \mathbf{id}) \cdot \mathbf{ap} & 3.17 \\
= & (\mathbf{rec}((x)^{\ulcorner}((\langle x\rangle \otimes \mathbf{id}) \cdot \mathbf{ap} \cdot t^{\urcorner}) \otimes \mathbf{id}) & \\
 & \quad \cdot(x)((^{\ulcorner}((\langle x\rangle \otimes \mathbf{id}) \cdot \mathbf{ap} \cdot t^{\urcorner} \otimes \mathbf{id}) \cdot \mathbf{ap}) & 2.16(1,2) \\
\searrow & (\mathbf{rec}((x)^{\ulcorner}((\langle x\rangle \otimes \mathbf{id}) \cdot \mathbf{ap} \cdot t^{\urcorner}) \otimes \mathbf{id}) \cdot (x)(((\langle x\rangle \otimes \mathbf{id}) \cdot \mathbf{ap} \cdot t) & \searrow_{\beta} \\
= & (\mathbf{rec}((x)^{\ulcorner}((\langle x\rangle \otimes \mathbf{id}) \cdot \mathbf{ap} \cdot t^{\urcorner}) \otimes \mathbf{id}) \cdot (x)((\langle x\rangle \otimes \mathbf{id}) \cdot \mathbf{ap} \cdot t & 2.16(1) \\
= & (\mathbf{rec}((x)^{\ulcorner}((\langle x\rangle \otimes \mathbf{id}) \cdot \mathbf{ap} \cdot t^{\urcorner}) \otimes \mathbf{id}) \cdot \mathbf{ap} \cdot t & \delta \\
= & \mathbf{iter}_f(t) \cdot t &
\end{array}
$$

(2) $\mathbf{iter}_b(t)$

$$
\begin{array}{lll}
= & ((\mathbf{rec}(x)^{\ulcorner}((\langle x\rangle \otimes t) \cdot \mathbf{ap}^{\urcorner}) \otimes \mathbf{id}_m) \cdot \mathbf{ap} & \\
\searrow & ((\mathbf{rec}(x)^{\ulcorner}((\langle x\rangle \otimes t) \cdot \mathbf{ap}^{\urcorner}) \otimes \mathbf{id}_m) & \\
 & \quad \cdot((x)^{\ulcorner}((\langle x\rangle \otimes t) \cdot \mathbf{ap}^{\urcorner} \otimes \mathbf{id}_m) \cdot \mathbf{ap} & 3.17 \\
= & ((\mathbf{rec}(x)^{\ulcorner}((\langle x\rangle \otimes t) \cdot \mathbf{ap}^{\urcorner}) \otimes \mathbf{id}_m) & \\
 & \quad \cdot(x)((^{\ulcorner}((\langle x\rangle \otimes t) \cdot \mathbf{ap}^{\urcorner} \otimes \mathbf{id}_m) \cdot \mathbf{ap}) & 2.16(1,2) \\
\searrow & ((\mathbf{rec}(x)^{\ulcorner}((\langle x\rangle \otimes t) \cdot \mathbf{ap}^{\urcorner}) \otimes \mathbf{id}_m) \cdot (x)(((\langle x\rangle \otimes t) \cdot \mathbf{ap}) & \searrow_{\beta} \\
= & ((\mathbf{rec}(x)^{\ulcorner}((\langle x\rangle \otimes t) \cdot \mathbf{ap}^{\urcorner}) \otimes \mathbf{id}_m) \cdot (\mathbf{id}_{m \Rightarrow m} \otimes t) \cdot \mathbf{ap} & 2.16(1,2), \delta \\
= & t \cdot ((\mathbf{rec}(x)^{\ulcorner}((\langle x\rangle \otimes t) \cdot \mathbf{ap}^{\urcorner}) \otimes \mathbf{id}_m) \cdot \mathbf{ap} & \\
= & t \cdot \mathbf{iter}_b(t) &
\end{array}
$$

■

Remark Note that when $t : \epsilon \rightarrow \epsilon$, then $\mathbf{iter}_f t \searrow t \otimes \mathbf{iter}_f t$ and $\mathbf{iter}_b t \searrow t \otimes \mathbf{iter}_b t$. Hence, for $\mathcal{K} = \{\mathbf{out}, \mathbf{box}\}$ (our fragment of π-calculus), we can encode a form of replication in $\mathcal{K}^{\Rightarrow}$ as follows:

$$\mathbf{rep}_x t \stackrel{\text{def}}{=} \mathbf{iter}_f(\mathbf{box}_x t)$$

Then $\mathbf{out}_x \otimes \mathbf{rep}_x a \searrow a \otimes \mathbf{rep}_x a$.

Discussion The encoding of recursion from reflexion, coding and application poses some interesting problems. In an extension of the theory AC^r by the higher order axioms introduced by Milner in [28], instead of proposition 3.19 we can derive fixed point equations as follows:

$$\begin{aligned} \mathbf{iter}_f(a) &= \mathbf{iter}_f(a) \cdot a \\ \mathbf{iter}_b(a) &= a \cdot \mathbf{iter}_b(a) \end{aligned}$$

giving $\mathbf{iter}_f(a)$ and $\mathbf{iter}_b(a)$ as left and right fixed points of a with respect to composition. Are they distinguished as fixed points, for instance as the *least* such, in some suitable ordering?

3.1.2 Discarding redundant restrictions

Inspection of the axioms introduced for reflexion reveals that the only structural (non-control) operations whose interaction with reflexion is not constrained is the identity. Reflexion of the identity corresponds, in terms of dataflow, to looping a channel onto itself. This means that the channel will not be accessible any longer (at least statically or structurally). We shall express this by considering such an action to be equal to $\mathbf{id}_\epsilon$ as follows:

$$\rho_0 : \uparrow_p \mathbf{id}_p = \mathbf{id}_\epsilon$$

We shall refer to the theory resulting from adding ρ_0 to AC^r as $\mathsf{AC}^{r\epsilon}$.

Lemma 3.20 $\uparrow_{(m)} \mathbf{id}_m = \mathbf{id}_\epsilon$.

Proof Let $m = p_r \otimes \cdots \otimes p_1$. Proof follows by induction on r. The case for $r = 0$ follows by definition. Assume $r = s + 1$, letting $m' = p_s \otimes \cdots \otimes p_1$. By induction hypothesis, we have $(*) : \uparrow_{(m')} \mathbf{id}_{m'} = \mathbf{id}_\epsilon$.

$$
\begin{aligned}
\uparrow_{(m)}\mathbf{id}_m &= \uparrow_{(m')}(\uparrow_{p_{s+1}}\mathbf{id}_m) & \\
&= \uparrow_{(m')}(\uparrow_{p_{s+1}}\mathbf{id}_{p_{s+1}} \otimes \mathbf{id}_{m'}) & \rho_2 \\
&= \uparrow_{(m')}\mathbf{id}_{m'} & \rho_0 \\
&= \mathbf{id}_\epsilon & (*)
\end{aligned}
$$

■

One outcome of the axiom is to provide molecular forms with garbage collection: restriction particles which do not bind any name are discarded. Hence we define *strict reflexive molecular forms over* $\mathcal{K}$ as just those reflexive molecular forms (over $\mathcal{K}$) where, for every restriction particle $\boldsymbol{\nu}(x)$, there is at least one free occurrence of the name x bound by it.

Theorem 3.21 *For any signature* $\mathcal{K}$*, the set of terms factored by* $\mathsf{AC}^{r\epsilon}$*,* $\mathcal{T}^r(\mathcal{K})/\mathsf{AC}^{r\epsilon}$ *is isomorphic to the set of strict reflexive molecular forms* $\mathcal{M}^{r\epsilon}(\mathcal{K})$.

Proof See [25]. ■

We define strict reflexive action calculi in a manner similar to reflexive action calculi; when $\mathcal{R}$ is understood we abbreviate $\mathsf{AC}^{r\epsilon}(\mathcal{K}, \mathcal{R})$ to $\mathsf{AC}^{r\epsilon}(\mathcal{K})$.

3.2 Reflexive Control Structures

An obvious way to proceed to a formulation of reflexive control structures is through the refinement of control structures by introducing the reflexion operation constrained by the equations $\rho_1 - \rho_6$. However, this will not give us enough axioms to obtain the initiality result for reflexive action calculi. The proof of proposition 3.8 gives an indication of what is lacking. The proof makes use of the fact that x is *not free* in $\uparrow_p((\mathbf{p}_{p,q} \otimes \mathbf{id}) \cdot (x)t)$. To obtain this equation in the abstract setting, we expect to rely on the corresponding property that x is not in the surface of $(\mathbf{c}_{p,q} \otimes \mathbf{id}) \cdot (x)a$. To do this, however, we must show that reflexion does not increase surface. In other words, from the fact that x is not in the surface of $(\mathbf{c}_{p,q} \otimes \mathbf{id}) \cdot (x)a$ we must be able to deduce that x is also absent from the surface of $\uparrow_p((\mathbf{c}_{p,q} \otimes \mathbf{id}) \cdot (x)a)$. It is unlikely that this property can be deduced from the axioms mentioned so far since none of them deal with the interaction between abstraction and reflexion. As will be explained, by taking the equation $(x)\uparrow_p a = \uparrow_p((\mathbf{c}_{p,q} \otimes \mathbf{id}) \cdot (x)a)$ (where $x : q$) as an axiom, it can be shown that reflexion does not increase surface.

However, we recall that the problem of ensuring that operations do not increase surface has already been encountered in the formulation of control structures. There it was solved by introducing the axioms, one for each control operation K:

$$[{}^{y}\!/\!{}_{x}]K(\vec{a}) = K([{}^{y}\!/\!{}_{x}]\vec{a})$$

It turns out that the addition of the axiom $[{}^{y}\!/\!{}_{x}]\uparrow a = \uparrow[{}^{y}\!/\!{}_{x}]a$, suggested by Hasegawa, provides a theory which is equipotent with that given by adding instead the axiom $(x)\uparrow_p a = \uparrow_p((\mathbf{c}_{p,q} \otimes \mathbf{id}) \cdot (x)a)$ (where $x : q$).

More directly, consider a possible proof that reflexive action calculi are initial reflexive control structures: this may be done by showing that the theory AC^{r} and the theory given by the axioms of reflexive control structures are equipotent over the term algebra. To show that $\boldsymbol{\sigma}$ is provable in the latter theory, the equation $[{}^{y}\!/\!{}_{x}]\uparrow a = \uparrow[{}^{y}\!/\!{}_{x}]a$ is necessary. We recall that $\boldsymbol{\sigma}$: $[{}^{y}\!/\!{}_{x}]a = \{{}^{y}\!/\!{}_{x}\}a$ asserts the identification of syntactic and semantic substitution. In the context of reflexive action calculi, the definition of syntactic substitution was extended to include $\{{}^{y}\!/\!{}_{x}\}\uparrow t = \uparrow\{{}^{y}\!/\!{}_{x}\}t$. Therefore, in a most direct manner, the axiom $[{}^{y}\!/\!{}_{x}]\uparrow a = \uparrow[{}^{y}\!/\!{}_{x}]a$ allows us to obtain $\boldsymbol{\sigma}$ in the theory defining reflexive control structures.

As mentioned previously, the notion of reflexion or feedback has found expression in several independent research efforts. One particular formulation which is suitable for our purposes comes from Joyal, Street and Verity [14] who introduced the notion—called a *trace*—in the context of symmetric monoidal categories. We shall see that their axioms for the trace operation, together with the additional axiom presented above concerning semantic substitution, suffice to give a category of models in which reflexive action calculi are initial. The definition of a *trace* on a strict symmetric monoidal category given below essentially follows [14].

Definition 3.22 (Trace) *A* trace *on a strict symmetric monoidal category A is a family of functions $\uparrow_m^{k,l} : A(m \otimes k, m \otimes l) \to A(k,l)$ indexed by the objects m of A such that the following axioms hold (in A):*

Yanking	T_1 :	$\uparrow_m \mathbf{c}_{m,m} = \mathbf{id}_m$
Superposing	T_2 :	$a_1 \otimes \uparrow_m a_2 = \uparrow_m((\mathbf{c}_{m,k} \otimes \mathbf{id}) \cdot (a_1 \otimes a_2) \cdot (\mathbf{c}_{l,m} \otimes \mathbf{id}))$ $(a_1 : k \to l)$
Naturality	T_3 :	$\uparrow_m a_1 \cdot a_2 = \uparrow_m(a_1 \cdot (\mathbf{id}_m \otimes a_2))$
	T_4 :	$a_1 \cdot \uparrow_m a_2 = \uparrow_m((\mathbf{id}_m \otimes a_1) \cdot a_2)$
	T_5 :	$\uparrow_m((a_1 \otimes \mathbf{id}) \cdot a_2) = \uparrow_n(a_2 \cdot (a_1 \otimes \mathbf{id}))$ $(a_1 : m \to n)$
Vanishing	T_6 :	$\uparrow_\epsilon a = a$
	T_7 :	$\uparrow_{m \otimes n} a = \uparrow_n(\uparrow_m a)$

■

Remark In the setting of action graphs, the trace axioms may be illustrated by the equalities shown in figure 3–1. The axioms T_3–T_5, which assert the naturality of $\uparrow$, are labelled *Right Tightening*, *Left Tightening* and *Sliding* respectively.

Notation We shall usually drop the superscripts k, l in $\uparrow_m^{k,l}$, since in any $\uparrow_m a$ they are deducible from the arities of a. Moreover, we shall refer to the trace operation in the context of reflexive control structures as *reflexion.*

Definition 3.23 (reflexive control structure) *Let A be a control structure over a signature $\mathcal{K}$. Then A together with a trace $\uparrow$ is a* reflexive control structure over *$\mathcal{K}$ if $\uparrow$ preserves the reaction relation and the following equation holds (in A):*

$$\sigma^\uparrow : \; [y/x]\uparrow a = \uparrow[y/x]a$$

■

We shall now show that reflexion does not increase surface.

Lemma 3.24 (Surface) $\mathbf{surf}(\uparrow a) \subseteq \mathbf{surf}(a)$.

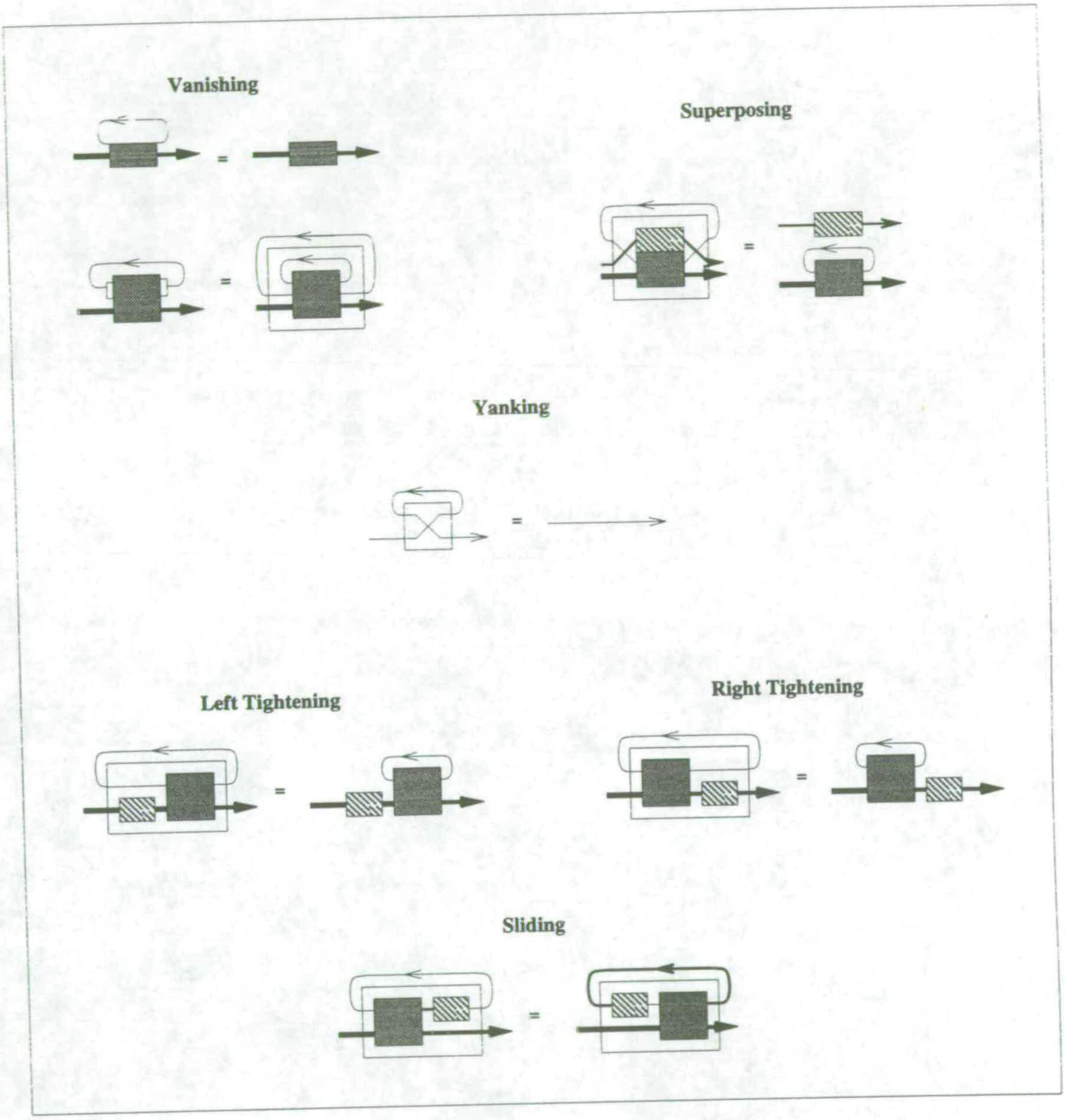


Figure 3–1: Trace axioms

Proof Assuming $x \notin \mathbf{surf}(a)$ we show that $x \notin \mathbf{surf}(\uparrow a)$, or in other words, $\mathbf{ab}_x \uparrow a = \mathbf{id} \otimes \uparrow a$. By $x \notin \mathbf{surf}(a)$ we have $ab_x a = \mathbf{id} \otimes a$. Now, consider an arbitrary y of the same arity as x. By proposition 2.30 $[y/x]a = a$. Hence $\uparrow a = \uparrow[y/x]a$ and hence, by $\sigma^\uparrow$, $\uparrow a = [y/x]\uparrow a$. Then, by proposition 2.30, $ab_x a = \mathbf{id} \otimes a$ and hence $x \notin \mathbf{surf}(\uparrow a)$. ■

Remark A comparison between the theory AC^r and that given by the axioms of a reflexive control structure is assisted by considering the (independent) replacement of axioms T_2 and T_5 by the axioms shown below:

$$\mathrm{T}_2' : \ \uparrow_m a \otimes \mathbf{id} = \uparrow_m (a \otimes \mathbf{id})$$
$$\mathrm{T}_5' : \ \uparrow_{m \otimes n} a = \uparrow_{n \otimes m}((c_{n,m} \otimes \mathbf{id}) \cdot a \cdot (c_{m,n} \otimes \mathbf{id}))$$

Proof Assume $a : m \otimes k \to m \otimes l$, $a_1 : k \to l$ and $a_2 : m \otimes k' \to m \otimes l'$.

$$
\begin{aligned}
(\mathrm{T}_2)\ & a_1 \otimes \uparrow_m a_2 \\
&= \mathbf{c}_{k,k'} \cdot (\uparrow_m a_2 \otimes a_1) \cdot \mathbf{c}_{l',l} && 2.24(1) \\
&= \mathbf{c}_{k,k'} \cdot (\uparrow_m a_2 \otimes \mathbf{id}) \cdot (\mathbf{id}_{l'} \otimes a_1) \cdot \mathbf{c}_{l',l} \\
&= \mathbf{c}_{k,k'} \cdot \uparrow_m (a_2 \otimes \mathbf{id}) \cdot (\mathbf{id}_{l'} \otimes a_1) \cdot \mathbf{c}_{l',l} \\
&= \mathbf{c}_{k,k'} \cdot \uparrow_m ((a_2 \otimes \mathbf{id}) \cdot (\mathbf{id}_{m \otimes l'} \otimes a_1)) \cdot \mathbf{c}_{l',l} && \mathrm{T}_3 \\
&= \mathbf{c}_{k,k'} \cdot \uparrow_m (a_2 \otimes a_1) \cdot \mathbf{c}_{l',l} \\
&= \uparrow_m ((\mathbf{id}_m \otimes \mathbf{c}_{k,k'}) \cdot (a_2 \otimes a_1) \cdot (\mathbf{id}_m \otimes \mathbf{c}_{l',l})) && \mathrm{T}_3,\ \mathrm{T}_4 \\
&= \uparrow_m ((\mathbf{id}_m \otimes \mathbf{c}_{k,k'}) \cdot \mathbf{c}_{m \otimes k',k} \cdot (a_1 \otimes a_2) \\
&\qquad \cdot \mathbf{c}_{l,m \otimes l'} \cdot (\mathbf{id}_m \otimes \mathbf{c}_{l',l})) && 2.24(1) \\
&= \uparrow_m ((\mathbf{id}_m \otimes \mathbf{c}_{k,k'}) \cdot (\mathbf{id}_m \otimes \mathbf{c}_{k',k}) \cdot (\mathbf{c}_{m,k} \otimes \mathbf{id}) \\
&\qquad \cdot (a_1 \otimes a_2) \cdot \mathbf{c}_{l,m \otimes l'} \cdot (\mathbf{id}_m \otimes \mathbf{c}_{l',l})) && 2.24(3) \\
&= \uparrow_m ((\mathbf{c}_{m,k} \otimes \mathbf{id}) \cdot (a_1 \otimes a_2) \cdot \mathbf{c}_{l,m \otimes l'} \cdot (\mathbf{id}_m \otimes \mathbf{c}_{l',l})) && \mathrm{S}_2 \\
&= \uparrow_m ((\mathbf{c}_{m,k} \otimes \mathbf{id}) \cdot (a_1 \otimes a_2) \cdot (\mathbf{c}_{l,m} \otimes \mathbf{id}) \\
&\qquad \cdot (\mathbf{id}_m \otimes \mathbf{c}_{l,l'}) \cdot (\mathbf{id}_m \otimes \mathbf{c}_{l',l})) && \mathrm{S}_3 \\
&= \uparrow_m ((\mathbf{c}_{m,k} \otimes \mathbf{id}) \cdot (a_1 \otimes a_2) \cdot (\mathbf{c}_{l,m} \otimes \mathbf{id})) && \mathrm{S}_2
\end{aligned}
$$

(T_2') $\uparrow_m a \otimes \mathbf{id}_n$

$$\begin{aligned}
&= \mathbf{c}_{k,n} \cdot (\mathbf{id}_n \otimes \uparrow_m a) \cdot \mathbf{c}_{n,l} && 2.24(1)\\
&= \mathbf{c}_{k,n} \cdot \uparrow_m((\mathbf{c}_{m,n} \otimes \mathbf{id}) \cdot (\mathbf{id}_n \otimes a) \cdot (\mathbf{c}_{n,m} \otimes \mathbf{id})) \cdot \mathbf{c}_{n,l} && \mathrm{T}_2\\
&= \uparrow_m((\mathbf{id}_m \otimes \mathbf{c}_{k,n}) \cdot (\mathbf{c}_{m,n} \otimes \mathbf{id}) \cdot (\mathbf{id}_n \otimes a)\\
&\qquad \cdot(\mathbf{c}_{n,m} \otimes \mathbf{id}) \cdot (id_m \otimes \mathbf{c}_{n,l})) && \mathrm{T}_3,\ \mathrm{T}_4\\
&= \uparrow_m(\mathbf{c}_{m\otimes l,n} \cdot (\mathbf{id}_n \otimes a) \cdot \mathbf{c}_{n,m\otimes l}) && \mathrm{S}_3, 2.24(3)\\
&= \uparrow_m(a \otimes \mathbf{id}_n) && 2.24(1)
\end{aligned}$$

(T_5) $\uparrow_n(a_2 \cdot (a_1 \otimes \mathbf{id}))$

$$\begin{aligned}
&= \uparrow_n(a_2 \cdot ((\uparrow_m \mathbf{c}_{m,m} \cdot a_1) \otimes \mathbf{id})) && \mathrm{T}_1\\
&= \uparrow_n(a_2 \cdot (\uparrow_m(\mathbf{c}_{m,m} \cdot (\mathbf{id}_m \otimes a_1)) \otimes \mathbf{id})) && \mathrm{T}_3\\
&= \uparrow_n(a_2 \cdot \uparrow_m((\mathbf{c}_{m,m} \cdot (\mathbf{id}_m \otimes a_1)) \otimes \mathbf{id})) && \mathrm{T}_2'\\
&= \uparrow_n\uparrow_m((\mathbf{id}_m \otimes a_2) \cdot ((\mathbf{c}_{m,m} \cdot (\mathbf{id}_m \otimes a_1)) \otimes \mathbf{id})) && \mathrm{T}_4\\
&= \uparrow_n\uparrow_m((\mathbf{id}_m \otimes a_2) \cdot ((a_1 \otimes \mathbf{id}_m) \cdot \mathbf{c}_{n,m}) \otimes \mathbf{id})) && \mathrm{S}_1\\
&= \uparrow_n\uparrow_m((\mathbf{id}_m \otimes a_2) \cdot (a_1 \otimes \mathbf{id}) \cdot (\mathbf{c}_{n,m} \otimes \mathbf{id}))\\
&= \uparrow_n\uparrow_m((a_1 \otimes a_2) \cdot (\mathbf{c}_{n,m} \otimes \mathbf{id}))\\
&= \uparrow_m\uparrow_n((\mathbf{c}_{n,m} \otimes \mathbf{id}) \cdot (a_1 \otimes a_2)\\
&\qquad \cdot(\mathbf{c}_{n,m} \otimes \mathbf{id}) \cdot (\mathbf{c}_{m,n} \otimes \mathbf{id})) && \mathrm{T}_5'\\
&= \uparrow_m\uparrow_n((\mathbf{c}_{n,m} \otimes \mathbf{id}) \cdot (a_1 \otimes a_2)) && \mathrm{S}_2\\
&= \uparrow_m\uparrow_n((\mathbf{c}_{n,m} \otimes \mathbf{id}) \cdot (a_1 \otimes \mathbf{id}) \cdot (\mathbf{id}_n \otimes a_2))\\
&= \uparrow_m\uparrow_n(((\mathbf{c}_{n,m} \cdot (a_1 \otimes \mathbf{id})) \otimes \mathbf{id}) \cdot (\mathbf{id}_n \otimes a_2))\\
&= \uparrow_m\uparrow_n((((\mathbf{id}_n \otimes a_1) \cdot \mathbf{c}_{n,n}) \otimes \mathbf{id}) \cdot (\mathbf{id}_n \otimes a_2)) && \mathrm{S}_1\\
&= \uparrow_m(\uparrow_n(((\mathbf{id}_n \otimes a_1) \cdot \mathbf{c}_{n,n}) \otimes \mathbf{id}) \cdot a_2) && \mathrm{T}_3\\
&= \uparrow_m((\uparrow_n((\mathbf{id}_n \otimes a_1) \cdot \mathbf{c}_{n,n}) \otimes \mathbf{id}) \cdot a_2) && \mathrm{T}_2'\\
&= \uparrow_m(((a_1 \cdot \uparrow_n \mathbf{c}_{n,n}) \otimes \mathbf{id}) \cdot a_2) && \mathrm{T}_4\\
&= \uparrow_m((a_1 \otimes \mathbf{id}) \cdot a_2) && \mathrm{T}_1
\end{aligned}$$

(T_5') $\uparrow_{m\otimes n} a$

$$\begin{aligned}
&= \uparrow_{m\otimes n}((\mathbf{c}_{m,n} \otimes \mathbf{id}) \cdot (\mathbf{c}_{n,m} \otimes \mathbf{id}) \cdot a) && \mathrm{S}_2\\
&= \uparrow_{m\otimes n}((\mathbf{c}_{m,n} \otimes \mathbf{id}) \cdot a \cdot (\mathbf{c}_{n,m} \otimes \mathbf{id})) && \mathrm{T}_5
\end{aligned}$$

■

While its similarity to σ_3 makes the axiom $\sigma^\uparrow$ appealing, sometimes we shall need equations in a form that expresses the interaction between reflexion and (the two kinds of) abstraction. Indeed, as we shall see below, the equations we shall take are not only provable, but actually induce identical theories when either of them replaces $\sigma^\uparrow$.

Proposition 3.25 *The theories obtained from adding any one of the equations shown below to the axioms of control structures together with T_1–T_7 are identical.*

$$\begin{array}{lll} \sigma^\uparrow\text{:} & [y/x]\uparrow_m a = \uparrow_m[y/x]a & \\ T_8\text{:} & (x)\uparrow_m a = \uparrow_m((\mathbf{c}_{m,p}\otimes\mathbf{id})\cdot(x)a) & (x:p) \\ T_8'\text{:} & \mathbf{ab}_x\uparrow_m a = \uparrow_m((\mathbf{c}_{m,p}\otimes\mathbf{id})\cdot\mathbf{ab}_x a\cdot(\mathbf{c}_{m,p}\otimes\mathbf{id})) & (x:p) \end{array}$$

Proof

$(\sigma^\uparrow \Rightarrow \mathrm{T}_8)$ $(x)\uparrow_m a$

$$\begin{array}{lll} = & (x)(\uparrow_m(((\langle x\rangle\otimes\mathbf{id})\cdot(x)a)) & \sigma_1 \\ = & (x)(\uparrow_m((\mathbf{id}\otimes\langle x\rangle\otimes\mathbf{id})\cdot(\mathbf{c}_{m,p}\otimes\mathbf{id})\cdot(x)a)) & 2.24(1,4) \\ = & (x)(((\langle x\rangle\otimes\mathbf{id})\cdot\uparrow_m((\mathbf{c}_{m,p}\otimes\mathbf{id})\cdot(x)a)) & \mathrm{T}_4 \\ = & (x)((\langle x\rangle\otimes\mathbf{id})\cdot\uparrow_m((\mathbf{c}_{m,p}\otimes\mathbf{id})\cdot(x)a) & 2.33(1),3.24 \\ = & \uparrow_m((\mathbf{c}_{m,p}\otimes\mathbf{id})\cdot(x)t) & 2.33(5) \end{array}$$

$(\mathrm{T}_8' \Rightarrow \sigma^\uparrow)$ $[y/x]\uparrow a$

$$\begin{array}{lll} = & ((\langle y\rangle\otimes\mathbf{id})\cdot(x)\uparrow_m a & \\ = & ((\langle y\rangle\otimes\mathbf{id})\cdot\mathbf{ab}_x\uparrow_m a\cdot(\omega\otimes\mathbf{id}) & \\ = & ((\langle y\rangle\otimes\mathbf{id})\cdot\uparrow_m((\mathbf{c}_{m,p}\otimes\mathbf{id})\cdot\mathbf{ab}_x a\cdot(\mathbf{c}_{m,p}\otimes\mathbf{id}))\cdot(\omega\otimes\mathbf{id}) & \mathrm{T}_8' \\ = & \uparrow_m((\mathbf{id}_m\otimes\langle y\rangle\otimes\mathbf{id})\cdot(\mathbf{c}_{m,p}\otimes\mathbf{id}) & \\ & \quad\cdot\mathbf{ab}_x a\cdot(\mathbf{c}_{m,p}\otimes\mathbf{id}))\cdot(\omega\otimes\mathbf{id}) & \mathrm{T}_4 \\ = & \uparrow_m(((\langle y\rangle\otimes\mathbf{id})\cdot\mathbf{ab}_x a\cdot(\mathbf{c}_{m,p}\otimes\mathbf{id}))\cdot(\omega\otimes\mathbf{id}) & 2.24(1,4) \\ = & \uparrow_m(((\langle y\rangle\otimes\mathbf{id})\cdot\mathbf{ab}_x a\cdot(\mathbf{c}_{m,p}\otimes\mathbf{id})\cdot(\mathbf{id}_m\otimes\omega\otimes\mathbf{id})) & \mathrm{T}_3 \\ = & \uparrow_m(((\langle y\rangle\otimes\mathbf{id})\cdot\mathbf{ab}_x a\cdot(\omega\otimes\mathbf{id})) & 2.24(1,4) \\ = & \uparrow_m(((\langle y\rangle\otimes\mathbf{id})\cdot(x)a) & \\ = & \uparrow_m[y/x]a & \end{array}$$

$(\mathrm{T}_8 \Rightarrow \mathrm{T}_8')$ $\mathbf{ab}_x \uparrow_m a$

$$\begin{array}{lll} = & (x)(\langle x\rangle \otimes \uparrow_m a) & 2.33(4) \\ = & (x)\uparrow_m((\mathbf{c}_{m,\epsilon} \otimes \mathbf{id}) \cdot (\langle x\rangle \otimes a) \cdot (\mathbf{c}_{p,m} \otimes \mathbf{id})) & \mathrm{T}_2 \\ = & (x)\uparrow_m((\langle x\rangle \otimes a) \cdot (\mathbf{c}_{p,m} \otimes \mathbf{id})) & 2.24(4) \\ = & \uparrow((\mathbf{c}_{m,p} \otimes \mathbf{id}) \cdot (x)(\langle x\rangle \otimes a) \cdot (\mathbf{c}_{p,m} \otimes \mathbf{id})) & \mathrm{T}_8 \\ = & \uparrow((\mathbf{c}_{m,p} \otimes \mathbf{id}) \cdot \mathbf{ab}_x a \cdot (\mathbf{c}_{p,m} \otimes \mathbf{id})) & 2.33(4) \end{array}$$

■

Remark We do not have $\mathbf{surf}(\uparrow a) = \mathbf{surf}(a)$ in general by the following counter-example in AC^r. Let $a = \langle x\rangle \otimes \boldsymbol{\omega}$. By lemma 3.10(1), $\uparrow a = \langle x\rangle \cdot \boldsymbol{\omega} = \mathbf{id}_\epsilon$. Hence $\mathbf{surf}(\uparrow a) = \emptyset$ while $\mathbf{surf}(a) = \{x\}$.

The equations given in proposition 3.25 express the interaction of reflexion with abstraction when the link created by abstraction is distinct from that operated upon by reflexion (the link which is fed-back). We shall now consider the case when reflexion operates on an abstraction.

Lemma 3.26 *Let $x : p$, $y : q$ and $x \neq y$. Then*

1. $(x)\uparrow_q(y)a = \uparrow_q(y)(x)a$;
2. $\mathbf{ab}_x\uparrow_q(y)a = \uparrow_q(y)(\mathbf{ab}_x a \cdot (\mathbf{c}_{q,p} \otimes \mathbf{id}))$;
3. $\uparrow_{p\otimes q}(x)(y)a = \uparrow_{q\otimes p}(y)(x)(a \cdot (\mathbf{c}_{p,q} \otimes \mathbf{id}))$;
4. $\uparrow_p(x)\uparrow_q(y)a = \uparrow_q(y)\uparrow_p(x)(a \cdot (\mathbf{c}_{q,p} \otimes \mathbf{id}))$.

Proof

(3) $\uparrow_{p\otimes q}(x)(y)a$

$$\begin{array}{lll} = & \uparrow_{p\otimes q}((\mathbf{c}_{p,q} \otimes \mathbf{id}) \cdot (y)(x)a) & 2.33(6) \\ = & \uparrow_{p\otimes q}((\mathbf{c}_{p,q} \otimes \mathbf{id}) \cdot (y)(x)a \cdot (\mathbf{c}_{p,q} \otimes \mathbf{id}) \cdot (\mathbf{c}_{q,p} \otimes \mathbf{id})) & \mathrm{S}_2 \\ = & \uparrow_{q\otimes p}(y)(x)(a \cdot (\mathbf{c}_{p,q} \otimes \mathbf{id})) & \mathrm{T}_5' \end{array}$$

(1) $\uparrow_q (yx)a$

$$
\begin{array}{lll}
= & (x)(\langle x\rangle \otimes \mathbf{id}) \cdot \uparrow_q (yx)a & 2.33(5) \\
= & (x)((\langle x\rangle \otimes \mathbf{id}) \cdot \uparrow_q (yx)a) & 2.33(1) \\
= & (x)\uparrow_q ((\mathbf{id}_q \otimes \langle x\rangle \otimes \mathbf{id}) \cdot (yx)a) & T_4 \\
= & (x)\uparrow_q (\mathbf{ab}_y((\langle x\rangle \otimes \mathbf{id}) \cdot (yx)a) & \\
= & (x)\uparrow_q (y)(((\langle x\rangle \otimes \mathbf{id}) \cdot (x)a) & 2.33(3),\ 2.24(4) \\
= & (x)\uparrow_q (y)a & \boldsymbol{\sigma}_1
\end{array}
$$

(2) $\mathbf{ab}_x \uparrow_q (y)a$

$$
\begin{array}{lll}
= & (x)(\langle x\rangle \otimes \uparrow_q (y)a) & 2.33(4) \\
= & (x)((\uparrow_q (y)a \otimes \langle x\rangle) \cdot (\mathbf{c}_{n,p} \otimes \mathbf{id})) & S_1 \\
= & (x)(\uparrow_q ((y)a \otimes \langle x\rangle) \cdot (\mathbf{c}_{n,p} \otimes \mathbf{id})) & T'_2,\ T_3 \\
= & (x)(\uparrow_q (y)(a \otimes \langle x\rangle) \cdot (\mathbf{c}_{n,p} \otimes \mathbf{id})) & 2.33(2) \\
= & (x)\uparrow_q ((y)(a \otimes \langle x\rangle) \cdot (\mathbf{id}_q \otimes \mathbf{c}_{n,p} \otimes \mathbf{id})) & T_3 \\
= & (x)\uparrow_q (y)((a \otimes \langle x\rangle) \cdot (\mathbf{id}_q \otimes \mathbf{c}_{n,p} \otimes \mathbf{id})) & 2.33(1) \\
= & (x)\uparrow_q (y)(((\langle x\rangle \otimes a) \cdot (\mathbf{c}_{p,q\otimes n} \otimes \mathbf{id}) \cdot (\mathbf{id}_q \otimes \mathbf{c}_{n,p} \otimes \mathbf{id})) & S_1 \\
= & (x)\uparrow_q (y)(((\langle x\rangle \otimes a) \cdot (\mathbf{c}_{p,q} \otimes \mathbf{id}) & \\
 & \quad \cdot (\mathbf{id}_q \otimes \mathbf{c}_{p,n} \otimes \mathbf{id}) \cdot (\mathbf{id}_q \otimes \mathbf{c}_{n,p} \otimes \mathbf{id})) & S_3 \\
= & (x)\uparrow_q (y)(((\langle x\rangle \otimes a) \cdot (\mathbf{c}_{p,q} \otimes \mathbf{id})) & S_2 \\
= & \uparrow_q (y)(x)(((\langle x\rangle \otimes a) \cdot (\mathbf{c}_{p,q} \otimes \mathbf{id})) & (1) \\
= & \uparrow_q (y)((x)((\langle x\rangle \otimes a) \cdot (\mathbf{c}_{p,q} \otimes \mathbf{id})) & 2.33(1) \\
= & \uparrow_q (y)(\mathbf{ab}_x a \cdot (\mathbf{c}_{p,q} \otimes \mathbf{id})) & 2.33(4)
\end{array}
$$

(4) $\uparrow_p (x)\uparrow_q (y)a$

$$
\begin{array}{lll}
= & \uparrow_p \uparrow_q (y)(x)a & (1) \\
= & \uparrow_{q\otimes p} (y)(x)a & T_7 \\
= & \uparrow_{p\otimes q} (x)(y)(a \cdot (\mathbf{c}_{q,p} \otimes \mathbf{id})) & (3) \\
= & \uparrow_q \uparrow_p (x)(y)(a \cdot (\mathbf{c}_{q,p} \otimes \mathbf{id})) & T_7 \\
= & \uparrow_q (y)\uparrow_p (x)(a \cdot (\mathbf{c}_{q,p} \otimes \mathbf{id})) & (1)
\end{array}
$$

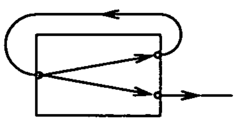
■

Restriction The reflexion operation allows us to derive restriction, an operation which occurs in the action calculi for both Petri nets and the π-calculus. Since it can be derived from algebraically defined operations, the restriction (or hiding) of names can be examined at a semantic level. Unsurprisingly, this involves the consideration of reflexion on the surface of an action.

Definition 3.27 (Restriction) *We define* restriction *on names as follows:*

$$\begin{aligned} \boldsymbol{\nu} &\stackrel{\text{def}}{=} \uparrow(x)\langle xx\rangle \\ (\boldsymbol{\nu}x)a &\stackrel{\text{def}}{=} (\boldsymbol{\nu}\otimes \mathbf{id})\cdot(x)t \end{aligned}$$

■



Notation When $\vec{x}$ consists of distinct names $x_1\cdots x_k$, we shall often write $(\boldsymbol{\nu}\vec{x})a$ to mean $(\boldsymbol{\nu}x_1)\cdots(\boldsymbol{\nu}x_k)a$.

The equations proved below give a flavour of how the restriction operation is expected to interact with the operations of a reflexive control structure.

Lemma 3.28 (Restriction)

1. $(\boldsymbol{\nu}x)a\otimes b = (\boldsymbol{\nu}x)(a\otimes b)$ *if* $x\notin \mathbf{surf}(b)$*;*

2. $a\otimes(\boldsymbol{\nu}x)b = (\boldsymbol{\nu}x)(a\otimes b)$ *if* $x\notin \mathbf{surf}(a)$*;*

3. $(\boldsymbol{\nu}x)a\cdot b = (\boldsymbol{\nu}x)(a\cdot b)$ *if* $x\notin \mathbf{surf}(b)$*;*

4. $a\cdot(\boldsymbol{\nu}x)b = (\boldsymbol{\nu}x)(a\cdot b)$ *if* $x\notin \mathbf{surf}(a)$*;*

5. $(\boldsymbol{\nu}x)\mathbf{ab}_y a = \mathbf{ab}_y(\boldsymbol{\nu}x)a$ *if* $x\neq y$*;*

6. $(\boldsymbol{\nu}x)(y)a = (y)(\boldsymbol{\nu}x)a$ *if* $x\neq y$*;*

7. $(\boldsymbol{\nu}x)\uparrow a = \uparrow(\boldsymbol{\nu}x)a$*;*

8. $(\boldsymbol{\nu} x)(\boldsymbol{\nu} y)a = (\boldsymbol{\nu} y)(\boldsymbol{\nu} x)a$.

Proof

(1) $(\boldsymbol{\nu} x)a \otimes b$

$$\begin{aligned} &= ((\boldsymbol{\nu} \otimes \mathbf{id}) \cdot (x)a) \otimes b \\ &= (\boldsymbol{\nu} \otimes \mathbf{id}) \cdot ((x)a \otimes b) \\ &= (\boldsymbol{\nu} \otimes \mathbf{id}) \cdot (x)(a \otimes b) && 2.33(2) \end{aligned}$$

(3) $(\boldsymbol{\nu} x)a \cdot b$

$$\begin{aligned} &= ((\boldsymbol{\nu} \otimes \mathbf{id}) \cdot (x)a) \cdot b \\ &= (\boldsymbol{\nu} \otimes \mathbf{id}) \cdot ((x)a \cdot b) \\ &= (\boldsymbol{\nu} \otimes \mathbf{id}) \cdot (x)(a \cdot b) && 2.33(1) \end{aligned}$$

(2) $a \otimes (\boldsymbol{\nu} x)b$

$$\begin{aligned} &= a \otimes ((\boldsymbol{\nu} \otimes \mathbf{id}) \cdot (x)b) \\ &= (a \otimes \mathbf{id}) \cdot (\mathbf{id}_n \otimes \boldsymbol{\nu} \otimes \mathbf{id}) \cdot (\mathbf{id}_n \otimes (x)b) && (a : m \to n) \\ &= (a \otimes \boldsymbol{\nu} \otimes \mathbf{id}) \cdot (\mathbf{id}_n \otimes (x)b) \\ &= (\boldsymbol{\nu} \otimes a \otimes \mathbf{id}) \cdot (\mathbf{c}_{p,n} \otimes \mathbf{id}) \cdot (\mathbf{id}_n \otimes (x)b) && 2.24(1,4) \\ &= (\boldsymbol{\nu} \otimes a \otimes \mathbf{id}) \cdot (x)(\mathbf{id}_n \otimes b) && 2.33(3),\ \mathrm{S1} \\ &= (\boldsymbol{\nu} \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes a \otimes \mathbf{id}) \cdot (x)(\mathbf{id}_n \otimes b) \\ &= (\boldsymbol{\nu} \otimes \mathbf{id}) \cdot (\mathbf{ab}_x a \otimes \mathbf{id}) \cdot (x)(\mathbf{id}_n \otimes b) \\ &= (\boldsymbol{\nu} \otimes \mathbf{id}) \cdot \mathbf{ab}_x(a \otimes \mathbf{id}) \cdot (x)(\mathbf{id}_n \otimes b) && 2.33(2,4) \\ &= (\boldsymbol{\nu} \otimes \mathbf{id}) \cdot (x)((a \otimes \mathbf{id}) \cdot (\mathbf{id}_n \otimes b)) \\ &= (\boldsymbol{\nu} \otimes \mathbf{id}) \cdot (x)(a \otimes b) \end{aligned}$$

(4) $a \cdot (\boldsymbol{\nu} x)b$

$$\begin{aligned} &= a \cdot ((\boldsymbol{\nu} \otimes \mathbf{id}) \cdot (x)b) \\ &= (a \cdot (\boldsymbol{\nu} \otimes \mathbf{id})) \cdot (x)b \\ &= (\boldsymbol{\nu} \otimes a) \cdot (x)b \\ &= (\boldsymbol{\nu} \otimes \mathbf{id}) \cdot (\mathbf{id}_p \otimes a) \cdot (x)b \\ &= (\boldsymbol{\nu} \otimes \mathbf{id}) \cdot \mathbf{ab}_x a) \cdot (x)b \\ &= (\boldsymbol{\nu} \otimes \mathbf{id}) \cdot (x)(a \cdot b) \end{aligned}$$

(5) $(\boldsymbol{\nu}x)\mathbf{ab}_y a$

$$
\begin{aligned}
&= (\boldsymbol{\nu}\otimes\mathbf{id})\cdot(x)\mathbf{ab}_y a && \\
&= (\mathbf{id}_q\otimes\boldsymbol{\nu}\otimes\mathbf{id})\cdot(\mathbf{c}_{q,p}\otimes\mathbf{id})\cdot(x)\mathbf{ab}_y a && (x:p, y:q)\\
&= (\mathbf{id}_q\otimes\boldsymbol{\nu}\otimes\mathbf{id})\cdot(\mathbf{c}_{q,p}\otimes\mathbf{id})\cdot\mathbf{ab}_x\mathbf{ab}_y a\cdot(\boldsymbol{\omega}\otimes\mathbf{id}) && \\
&= (\mathbf{id}_q\otimes\boldsymbol{\nu}\otimes\mathbf{id})\cdot\mathbf{ab}_y\mathbf{ab}_x a\cdot(\mathbf{c}_{q,p}\otimes\mathbf{id})\cdot(\boldsymbol{\omega}\otimes\mathbf{id}) && \text{S7}\\
&= (\mathbf{id}_q\otimes\boldsymbol{\nu}\otimes\mathbf{id})\cdot\mathbf{ab}_y\mathbf{ab}_x a\cdot\cdot(\mathbf{id}_q\otimes\boldsymbol{\omega}\otimes\mathbf{id}) && \text{S1,2.24(4)}\\
&= (\mathbf{id}_q\otimes\boldsymbol{\nu}\otimes\mathbf{id})\cdot\mathbf{ab}_y\mathbf{ab}_x a\cdot\cdot\mathbf{ab}_y(\boldsymbol{\omega}\otimes\mathbf{id}) && \text{S4}\\
&= (\mathbf{id}_q\otimes\boldsymbol{\nu}\otimes\mathbf{id})\cdot\mathbf{ab}_y(x)a && \\
&= \mathbf{ab}_y(\boldsymbol{\nu}\otimes\mathbf{id})\cdot\mathbf{ab}_y(x)a && \\
&= \mathbf{ab}_y((\boldsymbol{\nu}\otimes\mathbf{id})\cdot(x)a) &&
\end{aligned}
$$

(6) $(\boldsymbol{\nu}x)(y)a$

$$
\begin{aligned}
&= (\boldsymbol{\nu}x)(\mathbf{ab}_y a\cdot(\boldsymbol{\omega}\otimes\mathbf{id})) && \\
&= (\boldsymbol{\nu}x)\mathbf{ab}_y a\cdot(\boldsymbol{\omega}\otimes\mathbf{id}) && (3)\\
&= \mathbf{ab}_y(\boldsymbol{\nu}x)a\cdot(\boldsymbol{\omega}\otimes\mathbf{id}) && (5)\\
&= (y)(\boldsymbol{\nu}x)a &&
\end{aligned}
$$

(7) $(\boldsymbol{\nu}x)\uparrow_p a$

$$
\begin{aligned}
&= (\boldsymbol{\nu}\otimes\mathbf{id})\cdot(x)\uparrow_p a && \\
&= (\boldsymbol{\nu}\otimes\mathbf{id})\cdot\uparrow_p((\mathbf{c}_{p,q}\otimes\mathbf{id})\cdot(x)a) && \mathrm{T}_8, x:q\\
&= \uparrow_p((\mathbf{id}_p\otimes\boldsymbol{\nu}\otimes\mathbf{id})\cdot(\mathbf{c}_{p,q}\otimes\mathbf{id})\cdot(x)a) && \mathrm{T}_4\\
&= \uparrow_p((\boldsymbol{\nu}\otimes\mathbf{id})\cdot(x)a) && \text{2.24(1,4)}
\end{aligned}
$$

(8) $(\boldsymbol{\nu}x)(\boldsymbol{\nu}y)a$

$$
\begin{aligned}
&= (\boldsymbol{\nu}\otimes\mathbf{id})\cdot(x)((\boldsymbol{\nu}\otimes\mathbf{id})\cdot(y)a) && \\
&= (\boldsymbol{\nu}\otimes\mathbf{id})\cdot\mathbf{ab}_x(\boldsymbol{\nu}\otimes\mathbf{id})\cdot(x)(y)a && \\
&= (\boldsymbol{\nu}\otimes\mathbf{id})\cdot(\mathbf{id}_p\otimes\boldsymbol{\nu}\otimes\mathbf{id})\cdot(x)(y)a && (x:p)\\
&= (\boldsymbol{\nu}\otimes\boldsymbol{\nu}\otimes\mathbf{id})\cdot(x)(y)a && \\
&= (\boldsymbol{\nu}\otimes\boldsymbol{\nu}\otimes\mathbf{id})\cdot(\mathbf{c}_{p,q}\otimes\mathbf{id})\cdot(y)(x)a && \text{2.33(6)}, y:q\\
&= (\boldsymbol{\nu}\otimes\boldsymbol{\nu}\otimes\mathbf{id})\cdot(y)(x)a && \text{2.24(1,4)}\\
&= (\boldsymbol{\nu}\otimes\mathbf{id})\cdot(\mathbf{id}_q\otimes\boldsymbol{\nu}\otimes\mathbf{id})\cdot(y)(x)a && \\
&= (\boldsymbol{\nu}\otimes\mathbf{id})\cdot\mathbf{ab}_y(\boldsymbol{\nu}\otimes\mathbf{id})\cdot(y)(x)a && \\
&= (\boldsymbol{\nu}\otimes\mathbf{id})\cdot(y)((\boldsymbol{\nu}\otimes\mathbf{id})\cdot(x)a) &&
\end{aligned}
$$

■

Remark The above lemma holds in any control structure with control operation $\boldsymbol{\nu} : \epsilon \to p$ in its signature. Any such control has an empty surface (in the action calculus, its surface is empty, homomorphisms reduce surface, and a homomorphism exists from the action calculus to the control structure). Assuming this fact, the above lemma is provable in AC.

The following lemma illustrates the intuition of reflexion as feedback. In particular, (1) shows how a datum is fed back and (2) shows that a link looped onto itself effectively removes input access to that link, producing a restriction.

Lemma 3.29 *Let $x : p$. Then*

1. $\uparrow_p(x)(\langle y\rangle \otimes a) = [{}^{y}\!/\!_{x}]a$ *if* $x \neq y$*;*

2. $\uparrow_p(x)(\langle x\rangle \otimes a) = (\boldsymbol{\nu} x)a$*;*

Proof

(1) $\uparrow_p(x)(\langle y\rangle \otimes a)$

$=$	$\uparrow_p(\langle y\rangle \otimes (x)a)$	2.33(3)
$=$	$\uparrow_p((\langle y\rangle \otimes \mathbf{id}) \cdot (\mathbf{id}_p \otimes (x)a))$	
$=$	$\uparrow_p(\langle y\rangle \otimes \mathbf{id}) \cdot (x)a$	T_3
$=$	$\uparrow_p((\mathbf{id}_p \otimes \langle y\rangle \otimes \mathbf{id}) \cdot (\mathbf{c}_{p,p} \otimes \mathbf{id})) \cdot (x)a$	2.24(1,4)
$=$	$((\langle y\rangle \otimes \mathbf{id}) \cdot \uparrow_p(\mathbf{c}_{p,p} \otimes \mathbf{id}) \cdot (x)a$	T_4
$=$	$((\langle y\rangle \otimes \mathbf{id}) \cdot (\uparrow_p \mathbf{c}_{p,p} \otimes \mathbf{id}) \cdot (x)a$	T_2'
$=$	$((\langle y\rangle \otimes \mathbf{id}) \cdot (x)a$	T_1
$=$	$[{}^{y}\!/\!_{x}]a$	

(2) $\uparrow_p(x)(\langle x\rangle \otimes a)$

$=$	$\uparrow_p(x)(((\langle x\rangle \otimes \mathbf{id}) \cdot (\mathbf{id}_p \otimes a))$	
$=$	$\uparrow_p(x)(((\langle x\rangle \otimes \mathbf{id}) \cdot (\mathbf{id}_p \otimes (((\langle x\rangle \otimes \mathbf{id}) \cdot (x)a)))$	σ_1
$=$	$\uparrow_p(x)(((\langle x\rangle \otimes \mathbf{id}) \cdot (\mathbf{id}_p \otimes \langle x\rangle \otimes \mathbf{id}) \cdot (\mathbf{id}_p \otimes (x)a)))$	

$$= \ \uparrow_p(x)((\langle x\rangle \otimes \langle x\rangle \otimes \mathbf{id}) \cdot (\mathbf{id}_p \otimes (x)a))$$
$$= \ \uparrow_p(((x)\langle xx\rangle \otimes \mathbf{id}) \cdot (\mathbf{id}_p \otimes (x)a)) \qquad 2.33(1)$$
$$= \ \uparrow_p((x)\langle xx\rangle \otimes \mathbf{id}) \cdot (x)a \qquad \mathrm{T}_3$$
$$= \ (\boldsymbol{\nu} \otimes \mathbf{id}) \cdot (x)a \qquad \mathrm{T}_2'$$

■

Remarks

1. A generalised version of the above lemma is easily obtained as follows. Let $\vec{x}, \vec{y} : m$ and $\{\vec{x}\} \cap \{\vec{y}\} = \emptyset$. Then, if $[\vec{y}/\vec{x}]a \stackrel{\text{def}}{=} [y_1/x_1]\cdots[y_n/x_n]a$, for $\vec{x} = x_1\cdots x_n, \vec{y} = y_1\cdots y_n$:

   (a) $\uparrow_m(\vec{x})(\langle\vec{y}\rangle \otimes a) = [\vec{x}/\vec{y}]a$;

   (b) $\uparrow_m(\vec{x})(\langle\vec{x}\rangle \otimes a) = (\boldsymbol{\nu}\vec{x})a$;

   Proof Induction on $|\vec{x}|$.

   Base case $|\vec{x}| = 0$ Trivial.

   Inductive step $|\vec{x}| = i + 1$ Assume $u, v : p$. First consider an arbitrary $\vec{z}$.

$$\uparrow_{m\otimes p}(\vec{x}u)(\langle\vec{z}v\rangle \otimes a)$$
$$= \ \uparrow_p\uparrow_m(\vec{x})(u)(\langle\vec{z}\rangle \otimes \langle v\rangle \otimes a)$$
$$= \ \uparrow_p\uparrow_m(\vec{x})(\langle\vec{z}\rangle \otimes (u)(\langle v\rangle \otimes a)) \qquad 2.33(3)$$

   Case $\{\vec{z}v\} \cap \{\vec{x}u\} = \emptyset$:

$$= \ \uparrow_p(u)[\vec{z}/\vec{x}](\langle v\rangle \otimes a) \qquad \text{induction}$$
$$= \ \uparrow_p(u)(\langle v\rangle \otimes [\vec{z}/\vec{x}]a) \qquad 2.33^*, 2.24(4)^*$$
$$= \ [v/u][\vec{z}v/\vec{x}u]a \qquad 3.29(1)$$
$$= \ ((\langle v\rangle \otimes \mathbf{id}) \cdot (u)(((\langle\vec{z}\rangle \otimes \mathbf{id}) \cdot (\vec{x})a)$$
$$= \ ((\langle v\rangle \otimes \mathbf{id}) \cdot \mathbf{ab}_u((\langle\vec{z}\rangle \otimes \mathbf{id}) \cdot (u\vec{x})a$$
$$= \ ((\langle v\rangle \otimes \mathbf{id}) \cdot (\mathbf{id}_p \otimes \langle\vec{z}\rangle \otimes \mathbf{id}) \cdot (u\vec{x})a \qquad 2.30$$
$$= \ ((\langle v\vec{z}\rangle \otimes \mathbf{id}) \cdot (u\vec{x})a$$
$$= \ [\vec{z}v/\vec{x}u]a$$

Case $\vec{z}v = \vec{x}u$:

$$\begin{array}{lll} = & \uparrow_p(u)(\nu\vec{x})(\langle v\rangle \otimes a) & \text{induction} \\ = & \uparrow_p(u)(\langle v\rangle \otimes (\nu\vec{x})a) & 3.28(1)^* \\ = & (\nu\vec{x}u)a & 3.29(2), 3.28(8) \end{array}$$

2. We could, in place of (2), have derived $\uparrow\mathbf{ab}_x a = (\nu x)a$ using practically the same proof, since $\mathbf{ab}_x a = (x)(\langle x\rangle \otimes a)$. This fact is used to prove the following proposition which expresses the effect of restriction on the surface of an action.

Proposition 3.30

1. $\mathbf{surf}(\nu) = \emptyset$;

2. $\mathbf{surf}((\nu x)a) \subseteq \mathbf{surf}(a) - \{x\}$.

Proof For (1), $\mathbf{surf}(\nu) = \mathbf{surf}(\uparrow(x)\langle xx\rangle)$. Therefore, by lemma 3.24, $\mathbf{surf}(\nu) \subseteq \mathbf{surf}((x)\langle xx\rangle)$. But $\mathbf{surf}((x)\langle xx\rangle) = \emptyset$. For (2), by lemma 3.29, $(\nu x)a = \uparrow\mathbf{ab}_x a$. Hence $\mathbf{surf}((\nu x)a) = \mathbf{surf}(\uparrow\mathbf{ab}_x a) \subseteq \mathbf{surf}(\mathbf{ab}_x a) = \mathbf{surf}(a) - \{x\}$. ∎

We shall now express a sort of semantic counterpart to reflexive substitution. In particular it is worth noting how (semantic) substitution may occur across bindings without renaming, akin to the literal replacement of names employed in defining reflexive substitution over the molecular forms.

Proposition 3.31

$$\uparrow_p(x\vec{y})(\langle z\rangle \otimes a) = \begin{cases} (\vec{y})[z/x]a & (x \neq z) \\ (\vec{y})(\nu x)a & (x = z) \end{cases}$$

Proof First, by sufficiently many applications of lemma 3.26(1), $\uparrow_p(x\vec{y})(\langle z\rangle \otimes a) = (\vec{y})\uparrow_p(x)(\langle z\rangle \otimes a)$. By lemma 3.29 result follows immediately. ∎

In the expected manner, we shall now define a category of reflexive control structures in which the reflexive action calculus $\mathsf{AC}^{rs}(\mathcal{K})$ is initial.

Proposition 3.32 $\mathsf{AC}^{rs}(\mathcal{K})$ *is a reflexive control structure over* $\mathcal{K}$.

Proof By proposition 2.41, we already know that, over the term algebra, the axioms of a control structure are provable in AC, and therefore in AC^r. This means that $\mathsf{AC}^{rs}(\mathcal{K})$ is a control structure. Therefore, the result will follow if a trace is defined in terms of the operations of $\mathsf{AC}^{rs}(\mathcal{K})$ which satisfies the axioms T_1–T_7 and $\sigma^{\uparrow}$. Let the trace $\uparrow_m \stackrel{\text{def}}{=} \uparrow_{(m)}$. Then, by definition, the axioms T_6 and T_7 are provable. Also, T_1, T_3 and T_4 follow from lemma 3.9(1,3,4) respectively; T_5 follows from lemma 3.11. $\sigma^{\uparrow}$ follows immediately by σ and the definition of substitution. The proof of T_2 in AC^r is shown below:

$$
\begin{aligned}
(\mathrm{T}_2)\quad & t_1 \cdot \uparrow_m t_2 \\
&= \mathbf{p}_{k,k'} \cdot (\uparrow_m t_2 \otimes t_1) \cdot \mathbf{p}_{l',l} && 2.27, 2.24(1)\\
&= \mathbf{p}_{k,k'} \cdot (\uparrow_m t_2 \otimes \mathbf{id}) \cdot (\mathbf{id}_{l'} \otimes t_1) \cdot \mathbf{p}_{l',l} \\
&= \mathbf{p}_{k,k'} \cdot \uparrow_m (t_2 \otimes \mathbf{id}) \cdot (\mathbf{id}_{l'} \otimes t_1) \cdot \mathbf{p}_{l',l} \\
&= \mathbf{p}_{k,k'} \cdot \uparrow_m ((t_2 \otimes \mathbf{id}) \cdot (\mathbf{id}_{m\otimes l'} \otimes t_1)) \cdot \mathbf{p}_{l',l} && \rho_3\\
&= \mathbf{p}_{k,k'} \cdot \uparrow_m (t_2 \otimes t_1) \cdot \mathbf{p}_{l',l} \\
&= \uparrow_m ((\mathbf{id}_m \otimes \mathbf{p}_{k,k'}) \cdot (t_2 \otimes t_1) \cdot (\mathbf{id}_m \otimes \mathbf{p}_{l',l})) && \rho_3, \rho_4\\
&= \uparrow_m ((\mathbf{id}_m \otimes \mathbf{p}_{k,k'}) \cdot \mathbf{p}_{m\otimes k',k} \cdot (t_1 \otimes t_2) \\
&\qquad \cdot \mathbf{p}_{l,m\otimes l'} \cdot (\mathbf{id}_m \otimes \mathbf{p}_{l',l})) && 2.27, 2.24(1)\\
&= \uparrow_m ((\mathbf{id}_m \otimes \mathbf{p}_{k,k'}) \cdot (\mathbf{id}_m \otimes \mathbf{p}_{k',k}) \cdot (\mathbf{p}_{m,k} \otimes \mathbf{id}) \\
&\qquad \cdot (t_1 \otimes t_2) \cdot \mathbf{p}_{l,m\otimes l'} \cdot (\mathbf{id}_m \otimes \mathbf{p}_{l',l})) && 2.27, 2.24(3)\\
&= \uparrow_m ((\mathbf{p}_{m,k} \otimes \mathbf{id}) \cdot (t_1 \otimes t_2) \cdot \mathbf{p}_{l,m\otimes l'} \cdot (\mathbf{id}_m \otimes \mathbf{p}_{l',l})) && 2.27, \mathrm{S}_2\\
&= \uparrow_m ((\mathbf{p}_{m,k} \otimes \mathbf{id}) \cdot (t_1 \otimes t_2) \cdot (\mathbf{p}_{l,m} \otimes \mathbf{id}) \\
&\qquad \cdot (\mathbf{id}_m \otimes \mathbf{p}_{l,l'}) \cdot (\mathbf{id}_m \otimes \mathbf{p}_{l',l})) && 2.27, \mathrm{S}_3\\
&= \uparrow_m ((\mathbf{p}_{m,k} \otimes \mathbf{id}) \cdot (t_1 \otimes t_2) \cdot (\mathbf{p}_{l,m} \otimes \mathbf{id})) && 2.27, \mathrm{S}_2
\end{aligned}
$$

■

Definition 3.33 *The category of* reflexive control structures over $\mathcal{K}$, $\mathbf{CS}^{rs}(\mathcal{K})$, *is the subcategory of* $\mathbf{CS}^{s}(\mathcal{K})$ *whose objects are the reflexive control structures and whose morphisms are all those (morphisms between reflexive control structures) which preserve reflexion.* ■

Remark Since we have added only purely equational axioms to those of control structures, the category of reflexive control structures is guaranteed an initial object.

Theorem 3.34 $\mathsf{AC}^{rs}(\mathcal{K})$ *is initial in the category* $\mathbf{CS}^{rs}(\mathcal{K})$.

Proof Since $\mathsf{AC}^{rs}(\mathcal{K})$ is a reflexive control structure, there is a unique map to it from the initial reflexive control structure. That map is onto, so it remains to show that it is one to one. To do that, we must show that whenever the images of two terms are provably equal in AC^{r}, then they are equal in the initial reflexive control structure. It suffices to show that in the initial reflexive control structure, the axioms of AC^{r} are valid. We have already shown that the pure axioms of AC are valid (i.e. true in any control structure, hence in any reflexive control structure); therefore, it remains to validate the axioms ρ_1–ρ_5 together with the axiom schemas σ and γ. The validity of the axioms ρ_1–ρ_5 follows from T_1, T'_2, T_3, T_4, T'_5 respectively. By propositions 2.36 and $\sigma^{\uparrow}$ we get σ. It remains to show γ.

For γ, it suffices, by proposition 2.30, to show that whenever $x \notin \mathbf{fn}(t)$ then $x \notin \mathbf{surf}(t)$. This involves an easy induction on the structure of terms (of the initial reflexive control structure): the only new case is that for reflexion where for $t \equiv \uparrow t'$, we have $x \in \mathbf{fn}(\uparrow t')$ if and only if $x \in \mathbf{fn}(t')$. Hence, assuming $x \notin \mathbf{fn}(t)$ gives $x \notin \mathbf{fn}(t')$ and by induction hypothesis we get $x \notin \mathbf{surf}(t')$. By lemma 3.24 the result follows immediately. ■

We shall now define a subcategory of models for reflexive action calculi which takes the dynamics into account.

Definition 3.35 *If* $\mathcal{R}$ *is a set of reaction rules over* $\mathcal{K}$, *then* $\mathbf{CS}^{r}(\mathcal{K},\mathcal{R})$ *is the full subcategory of* $\mathbf{CS}^{rs}(\mathcal{K})$ *containing just those reflexive control structures whose reaction relation satisfies* $\mathcal{R}$. ■

Corollary 3.36 $\mathsf{AC}^{r}(\mathcal{K},\mathcal{R})$ *is initial in* $\mathbf{CS}^{r}(\mathcal{K},\mathcal{R})$. ■

3.2.1 Strict reflexive control structures

We shall now define a category of reflexive control structures (a subcategory of $\mathbf{CS}^{rs}(\mathcal{K})$) in which the strict reflexive action calculus $\mathsf{AC}^{r\epsilon}(\mathcal{K})$ is initial. An interesting property of the objects of this category is that their surface map can be characterised in a very appealing manner.

Definition 3.37 (strict reflexive control structure) *Let A be a reflexive control structure over a set of controls $\mathcal{K}$ (and over X). Then A is a* strict reflexive control structure *if the equation $\uparrow_m \mathbf{id}_m = \mathbf{id}_\epsilon$ holds.* ∎

Proposition 3.38 $\mathsf{AC}^{r\epsilon}(\mathcal{K})$ *is a strict reflexive control structure over K.*

Proof Again choosing $\uparrow_m$, by proposition 3.32 we have that the axioms of a reflexive control structure are provable in AC^{r}, and therefore in $\mathsf{AC}^{r\epsilon}$. By lemma 3.20, the axiom T_0 is provable, hence result follows. ∎

Theorem 3.39 *Strict reflexive control structures over $\mathcal{K}$ and homomorphisms of reflexive control structures form a category in which* $\mathsf{AC}^{r\epsilon}(\mathcal{K})$ *is initial.*

Proof We already have, by proposition 3.38 that $\mathsf{AC}^{r\epsilon}(\mathcal{K})$ is a strict reflexive action calculus. By theorem 3.34, we also know that, over the term algebra, all the axioms of AC^{r} are provable from the axioms of reflexive control structures. By a similar argument, it suffices to show that ρ_0 is derivable. This follows by the fact that ρ_0 is a special instance of T_0. ∎

We note that restrictions of names which are not in the surface of an action a should not affect the behaviour of the action. Indeed, the strict reflexive molecular forms illustrate this in a concrete manner, by discarding restriction particles which do not effectively bind any name in the action. An analogous semantic notion of such discarding of redundant restrictions is obtainable in the strict theory $\mathsf{AC}^{r\epsilon}$.

Lemma 3.40 (Garbage collection) *If $x \notin \mathbf{surf}(a)$, then $(\nu x)a = a$.*

Proof Assume $x \notin \mathbf{surf}(a)$. Then $\mathbf{ab}_x a = \mathbf{id}_p \otimes a$.

$$
\begin{aligned}
(\nu x)a &= \uparrow\mathbf{ab}_x a \\
&= \uparrow(\mathbf{id}_p \otimes a) \\
&= \uparrow((\mathbf{id}_p \otimes \mathbf{id}) \cdot (\mathbf{id}_p \otimes a)) \\
&= \uparrow(\mathbf{id}_p \otimes \mathbf{id}) \cdot a && \mathrm{T}_3 \\
&= (\uparrow\mathbf{id}_p \otimes \mathbf{id}) \cdot a && \mathrm{T}_2' \\
&= (\mathbf{id}_\epsilon \otimes \mathbf{id}) \cdot a && \mathrm{T}_0 \\
&= a
\end{aligned}
$$

■

Corollary 3.41 $\boldsymbol{\nu} \cdot \boldsymbol{\omega} = \mathbf{id}_\epsilon$

Proof $\boldsymbol{\nu} \cdot \boldsymbol{\omega} = \boldsymbol{\nu} \cdot (x)\mathbf{id}_\epsilon = \mathbf{id}_\epsilon$. ■

Example As an example of garbage collection following computation consider the following reaction in $\mathsf{PIC}^{\mathsf{r}}$, assuming $x \notin \mathbf{surf}(a)$:

$$
\begin{aligned}
(\nu x)(\mathbf{out}_x \otimes \mathbf{box}_x a) &\searrow (\nu x)a \\
&= a && 3.40
\end{aligned}
$$

We shall now show that, in strict reflexive control structures, the surface of an action a is exactly given by the set of names x, the restriction of which changes the action, i.e. $(\nu x)a \neq a$. This corresponds very satisfyingly with the notion that the surface of an action consists of the names which "matter semantically" in that action.

Proposition 3.42 (Surface) *For any name x and action a, $x \in \mathbf{surf}(a)$ if and only if $(\nu x)a \neq a$.*

Proof ($\Longleftarrow$) By lemma 3.40 we have that if $(\nu x)a \neq a$ then $x \in \mathbf{surf}(a)$.

($\Longrightarrow$) We now show that if $x \in \mathbf{surf}(a)$, $(\nu x)a \neq a$. We shall prove the contrapositive: assuming $(\nu x)a = a$ we show that $x \notin \mathbf{surf}(a)$. Now by lemma 3.24,

$\mathbf{surf}(\uparrow\mathbf{ab}_x a) \subseteq \mathbf{surf}(\mathbf{ab}_x a)$. Hence by lemma 2.31(3) $x \notin \mathbf{surf}(\mathbf{ab}_x a)$ and hence $x \notin \mathbf{surf}(\uparrow\mathbf{ab}_x a)$. But, by lemma 3.29(2), $(\boldsymbol{\nu} x)a = \uparrow(x)(\langle x\rangle \otimes a) = \uparrow\mathbf{ab}_x a$. Since $a = (\boldsymbol{\nu} x)a$, we get $a = \uparrow\mathbf{ab}_x a$ and therefore $x \notin \mathbf{surf}(a)$. ■

Discussion What are the relative merits of the two kinds of reflexive molecular forms and the abstract structures they give rise to? Consider the molecular forms as a kind of "programmer's notation", where the imported names serve as formal parameters. Then, if programmers are to be allowed to declare extra (local) names which they then do not use within the body of the program, then the strict form is not suitable. Discarding redundant restrictions is, in a sense, a semantic or behavioural notion rather than a syntactic one. This does not mean that models in which the strictness axiom holds are uninteresting; indeed, we expect that, in behaviourally motivated models of (non-strict) reflexive action calculi, the strictness axiom will hold. This point, in modified form, will again appear when we deal with the operational semantics of the reflexive π-calculus, $\mathsf{PIC}^{\mathsf{r}}$, in chapters 5 and 6.

Chapter 4

Skeleta

So far, the main examples of control structures we have encountered are action calculi and their reflexive variants. We shall now explore two instances of strict reflexive control structures which are simple, universal, in the sense that they arise from any set of controls $\mathcal{K}$, and are models of *static* action calculi. Both examples can be obtained by factoring the term algebra $\mathcal{T}^{r}(\mathcal{K})$ by the congruence induced by the theory $\mathsf{AC}^{r\epsilon}$ together with simple equations. Alternatively, a characterisation in terms of the term algebra $\mathcal{T}(\mathcal{K})$ over any signature $\mathcal{K}$ may be obtained which contains at least the restriction operation ν. We shall adopt the latter approach since it allows the results to hold in the wider context of control structures (rather than reflexive, or even strict reflexive, control structures).

We choose to call such structures *skeleta* since they do not contain any reference to the specific controls making up the bodies of the action from which each skeleton arises: only the free names and (some of) the binding structure are retained.

Of particular interest is their usefulness in classifying reaction rules for action calculi. The idea of using certain control structures to classify dynamics first appeared in [20], where a control structure IM was described together with its property as a classifier of reaction rules according to whether or not they result in a certain kind of mobility. In summary, for those action calculi (such as the lambda calculus) in which the kind of immobility characterised by IM is expressible, there exists a morphism of control structures to IM, whereas for other action calculi which exhibit a corresponding form of mobility, such as the action calculus originating from the π-calculus, no such morphism exists. We expect to find many such *classifiers*, each characterising some property of the dynamics of control structures. Both kinds of skeleta presented here may be employed as classifiers: whether the properties they embody are useful in understanding the dynamics of processes is another question.

Outline A simple kind of skeleta, called *pure skeleta*, is introduced in section 1; it results from an analysis of the exported names in the molecular forms under contexts built from the operations of reflexive control structures. They are presented as *skeletal forms*, a form which emphasises their nature as abstractions of molecular forms. An alternative presentation as a term algebra—essentially the same algebra as for action calculi (with restriction) but with additional axioms—is given. This further clarifies what structure in the actions of action calculi is being forgotten in obtaining pure skeleta. Indeed, this consideration leads to an abstract characterisation of (the control structure of) pure skeleta as a terminal object in a suitable category of control structures. Section 1 ends with an exploration of dynamical aspects of pure skeleta, in particular, of their use as a classifier of action calculi upon a property of their dynamics.

In section 2, the notion of name export which motivated pure skeleta is regarded as an instance of information flow. A slightly richer, but still concrete, notion of information than exported names is proposed, leading to a corresponding kind of skeletal form: *restriction skeleta*. As for pure skeleta, a term algebra presentation of restriction skeleta is given with the relevant theory being obtained by revoking one axiom from that which gives pure skeleta. Prior to dealing with the dynamic aspects of restriction skeleta, Milner's *effect structures* [21]—an abstract treatment of computationally-generated information—are reviewed. We show that the concrete notion of information adopted in the context of restriction skeleta gives an effect structure for just those action calculi which have a homomorphism to (the control structure of) restriction skeleta.

4.1 Pure Skeleta

Pure skeleta arise from a consideration of the free names exported by an action. Consider an action a in the reflexive action calculus over the controls $\{\mathbf{out}, \mathbf{box}\}$ in its molecular form:

$$a = (x)\,[\langle xu\rangle\mathbf{out}(\ ), \langle y\rangle\mathbf{box}b(w)]\,\langle wxz\rangle$$

The action a exports the names w, x and z, of which only z is free. Although they are both bound, there is a significant difference between the names w and x. If a datum $\langle v\rangle$—or indeed any action which exports the free name v—is precomposed to a, then, in the composite action, x would be replaced by v and the action will then be able to export the free name v. However, there is no operation in the action calculus that, when applied to a, would allow the bound name w to be replaced by a free name. Note that it doesn't matter to which molecule the binding occurs, the names thus bound cannot be replaced by free names as a result of applying any operation defined in terms of the action calculi operations. Thus, we distinguish between three kinds of exported name: those which are free; those which are bound by the names in the import vector; and finally, the *control bound* names which can never be replaced by free ones (unless freed as a result of reaction). It may be argued that, since it is only the exported *free* names that we are concerned with and since control bound names can never be replaced by free ones, any distinction between control bound names can be ignored. This is what we shall do to obtain *pure skeleta.*

4.1.1 Skeletal forms

Definition 4.1 (Pure skeleta) *The actions of* pSKEL pure skeleta, *ranged over by* s *have the following form:*

$$s \ ::= \ (\vec{x})\langle\vec{y}\rangle$$

where $\{\vec{y}\} \subseteq X \cup \{\star\}$ *with* $\star \notin X$, $\vec{x}$ *are distinct names and* $s : m \to n$ *if* $\vec{x} : m$ *and* $\vec{y} : n$. *Each name in* $\vec{x}$ *binds any occurrence of that name in* $\vec{y}$; *names (in* X*) not thus bound are free. Alphaconversion of bound names is allowed.* ∎

We shall now show that pure skeleta give strict reflexive control structures. First, we must define the operations of a reflexive control structure on pSKEL.

Definition 4.2 *We define the following operations on* pSKEL. *Assume* $s = (u\vec{x})\langle v\vec{y}\rangle$, $s_1 = (\vec{u})\langle\vec{v}\rangle$ *and* $s_2 = (\vec{x})\langle\vec{y}\rangle$ *with the names in* $\vec{u}$ *distinct from those in* $\vec{x}$.

$$
\begin{array}{rcll}
\mathbf{id}_m & \stackrel{\text{def}}{=} & (\vec{x})\langle\vec{x}\rangle & (\vec{x} : m) \\
\langle x\rangle & \stackrel{\text{def}}{=} & (\,)\langle x\rangle & \\
\boldsymbol{\omega} & \stackrel{\text{def}}{=} & (x)\langle\,\rangle & \\
s_1 \cdot s_2 & \stackrel{\text{def}}{=} & (\vec{u})\langle\sigma\vec{y}\rangle & (\sigma = \{\vec{v}/\vec{x}\}) \\
s_1 \otimes s_2 & \stackrel{\text{def}}{=} & (\vec{u}\vec{x})\langle\vec{v}\vec{y}\rangle & \\
\mathbf{ab}_x s_1 & \stackrel{\text{def}}{=} & (x\vec{u})\langle x\vec{v}\rangle & \\
\uparrow s & \stackrel{\text{def}}{=} & \begin{cases} (\vec{x})\langle\{v/u\}\vec{y}\rangle & \text{if } u \neq v \\ (\vec{x})\langle\{\star/u\}\vec{y}\rangle & \text{if } u = v \end{cases} &
\end{array}
$$

∎

Proposition 4.3 *For any set of controls* $\mathcal{K}$, pSKEL *together with the operations of definition 4.2, any reaction relation on* pSKEL *and, for each* $K \in \mathcal{K}$,

$$K(\vec{s}) \stackrel{\text{def}}{=} (\vec{x})\langle\star\cdots\star\rangle$$

is a strict reflexive control structure over $\mathcal{K}$.

Proof Consider the molecular forms over the strict reflexive action calculus $\mathsf{AC}^{\mathsf{r}\epsilon}(\mathcal{K})$. We define the map **pskel** : $\mathsf{AC}^{\mathsf{r}\epsilon}(\mathcal{K}) \to$ pSKEL as follows: for each $a \in \mathsf{AC}^{\mathsf{r}\epsilon}(\mathcal{K})$ with molecular form $(\vec{u})\vec{\mu}(\vec{v})\langle\vec{w}\rangle$, $\mathbf{pskel}(a) \stackrel{\text{def}}{=} (\vec{u})\langle\{\vec{\star}/\vec{v}\}\vec{w}\rangle$. Clearly, **pskel** is onto. It therefore suffices to show that **pskel** preserves the operations of a strict reflexive control structure. ∎

Remark It can easily be demonstrated that the mapping **pskel** : $(\vec{u})\vec{\mu}(\vec{v})\langle\vec{w}\rangle \mapsto (\vec{u})\langle\{\vec{\star}/\vec{v}\}\vec{w}\rangle$, when defined on the molecular forms for both action calculi and reflexive action calculi, preserves the operations of a control structure, and in the case of reflexive action calculi, preserves reflexion as well.

It will be noted that, for all the main results of this section bearing reference to action calculi and control structures, corresponding ones replacing those references respectively by ones to reflexive action calculi and reflexive control structures (and even their strict variants) are easily obtained with almost identical proofs. The reason behind this uniformity must come from the fact that pSKEL captures underlying structure which is common to the molecular forms of all these variants. The following proposition is a case in point; the propositions obtained by replacing $\mathsf{AC}^s(\mathcal{K})$ and $\mathbf{CS}^s(\mathcal{K})$ by $\mathsf{AC}^{rs}(\mathcal{K})$ and $\mathbf{CS}^{rs}(\mathcal{K})$ respectively, and also $\mathsf{AC}^{r\epsilon s}(\mathcal{K})$ and $\mathbf{CS}^{r\epsilon s}(\mathcal{K})$ (the static counterparts of $\mathsf{AC}^{r\epsilon}(\mathcal{K})$ and $\mathbf{CS}^{r\epsilon}(\mathcal{K})$ respectively), are demonstrable by practically identical proofs.

The pure skeleton arising from the action $(x)\,[\langle xu\rangle\mathbf{out}(\ ), \langle y\rangle\mathbf{box}b(w)]\,\langle wxz\rangle$ is $(x)\langle\star xz\rangle$. Thus, as the following proposition shows formally, the pure skeleton of an action (in an action calculus) accounts for the free and import-bound names exported in the molecular form of that action.

Proposition 4.4 *Let* **pskel** *be the unique morphism (in* $\mathbf{CS}^s(\mathcal{K})$*) from* $\mathsf{AC}^s(\mathcal{K})$ *to* pSKEL. *Then, for all* $s \in$ pSKEL *and* $z \neq \star$, $\mathbf{pskel}\,(a) = s \cdot (\mathbf{id}_m \otimes \langle z\rangle \otimes \mathbf{id})$ *implies that, for some* a', $a = a' \cdot (\mathbf{id}_m \otimes \langle z\rangle \otimes \mathbf{id})$ *with* $\mathbf{pskel}\,(a') = s$.

Proof Consider the molecular form of $a = (\vec{u})\vec{\mu}(\vec{v})\langle\vec{w}\rangle$. Then, $\mathbf{pskel}\,(a) = (\vec{u})\langle\{\vec{\star}/\vec{v}\}w\rangle$. Then if $\mathbf{pskel}\,(a) = s \cdot (\mathbf{id}_m \otimes \langle z\rangle \otimes \mathbf{id})$, we must have $s = (\vec{x})\langle\vec{z}_1\vec{z}_2\rangle$ with $\vec{w} = \vec{w}_1 z \vec{w}_2$, $\vec{z}_i = \{\vec{\star}/\vec{v}\}\vec{w}_i$ $(i = 1, 2)$, $\vec{z}_1 : m$ and $z \notin \vec{x}$. Hence, $a = (\vec{u})\vec{\mu}(\vec{v})\langle\vec{w}_1 z\vec{w}_2\rangle = (\vec{u})\vec{\mu}(\vec{v})\langle\vec{w}_1\vec{w}_2\rangle \cdot (\mathbf{id}_m \otimes \langle z\rangle \otimes \mathbf{id})$. ∎

In pSKEL we do not expect to distinguish between different control actions having the same arity. The following proposition allows us to derive this property:

Proposition 4.5 *In* pSKEL, *for any* K, $K(\vec{s}) = \omega^m \otimes \nu^n$.

Proof Trivial. ∎

Corollary 4.6 *For any two control actions $K_1(\vec{a})$ and $K_2(\vec{b})$, if their arities are identical, then $K_1(\vec{a}) = K_2(\vec{b})$.* ∎

Nor do we expect to distinguish between control bound names:

Proposition 4.7 $\boldsymbol{\nu} \cdot (x)\langle xx\rangle = \boldsymbol{\nu} \otimes \boldsymbol{\nu}$.

Proof Trivial. ∎

4.1.2 Terms

We shall now give a characterisation of pSKEL as a quotient of the terms $\mathcal{T}(\mathcal{K})$, when $\mathcal{K}$ contains the restriction controls $\boldsymbol{\nu}_p : \epsilon \rightarrow p$, for each prime p. To denote such signatures uniformly over arbitrary control sets, we shall write $\mathcal{K}_{\boldsymbol{\nu}}$ for the signature $\mathcal{K} \cup \{\boldsymbol{\nu}_p \mid p \in P\}$.

Definition 4.8 (The theory $\mathsf{AC}^{\mathsf{pS}}$) *Let $\mathsf{AC}^{\mathsf{pS}}$ be the theory resulting from the addition of the following equation to the theory* AC*:*

$$\begin{aligned}
\boldsymbol{\nu} \cdot \boldsymbol{\omega} &= \mathbf{id}_\epsilon \\
\boldsymbol{\nu} \cdot (x)\langle xx\rangle &= \boldsymbol{\nu} \otimes \boldsymbol{\nu} \\
K(\vec{t}) &= \boldsymbol{\omega}^m \otimes \boldsymbol{\nu}^n
\end{aligned}$$

∎

Since pSKEL is a strict reflexive control structure, it might be expected that the characterisation we seek would involve the reflexive terms $\mathcal{T}^{\mathsf{r}}(\mathcal{K})$. Indeed, this is possible, and it is fairly easy to show that adding the equation $K(\vec{t}) = \boldsymbol{\omega}^m \otimes \boldsymbol{\nu}^n$ (with $\boldsymbol{\nu}$ as defined previously) to AC^{r} would allow every term in $\mathcal{T}^{\mathsf{r}}(\mathcal{K})$ to be proven equal to a term in $\mathcal{T}(\mathcal{K}_{\boldsymbol{\nu}})$ in the resulting theory. Also, the equation $\boldsymbol{\nu} \cdot \boldsymbol{\omega} = \mathbf{id}_\epsilon$ is derivable in the theory $\mathsf{AC}^{\mathsf{r}\epsilon}$ and therefore adding the other two equations to this theory would also suffice. Our chosen approach is then justified by the fact that our results do not depend on the presence of reflexion but rather on that of restriction, which while derivable in arbitrary reflexive control structures, may nevertheless be present in control structures where the reflexion operation is absent. The advantage of our approach will be apparent as we shall be able to derive results concerning the classification of dynamics which are applicable to both reflexive and (ordinary) action calculi.

Let us extend the notion of substitution of names to restriction particles: we let $\{\nu/x\}t$ denote the term obtained by replacing every occurrence of $\langle x\rangle$ by ν in t, provided the name x is free in that occurrence. Note that since ν is not itself a name, name clashes with binding occurrences can never occur.

Lemma 4.9 *For any term t, $\mathsf{AC}^{\mathsf{ps}} \vdash (\nu \otimes \mathbf{id}) \cdot (x)t = \{\nu/x\}t$.*

Proof Induction on the structure of terms. ∎

The following definitions shall provide the isomorphism (and its inverse) between pSKEL and the term algebra factored by the theory $\mathsf{AC}^{\mathsf{ps}}$. First, the map from pSKEL to the terms $\mathcal{T}(\mathcal{K}_\nu)$.

Definition 4.10 (pSKEL to Terms) *Define the translation $\widehat{(-)} : \mathsf{pSKEL} \to \mathcal{T}(\mathcal{K}_\nu)$ as follows:*

$$\widehat{(\vec{x})\langle\vec{y}\rangle} \stackrel{\text{def}}{=} (\vec{x})\langle\vec{y}\rangle$$

where $\widehat{\langle\star\rangle} \stackrel{\text{def}}{=} \nu$. ∎

We would like to get a translation from pSKEL to equivalence classes of terms $\mathcal{T}(\mathcal{K}_\nu)$ induced by the theory $\mathsf{AC}^{\mathsf{ps}}$. By an abuse of notation let $\widehat{(-)}$ denote a mapping from the skeleton s to the equivalence class $[\widehat{s}]$.

Lemma 4.11 *The translation $\widehat{(-)} : \mathsf{pSKEL} \to \mathcal{T}(\mathcal{K}_\nu)/\mathsf{AC}^{\mathsf{ps}}$ is well-defined.*

Proof It suffices to show that the translation $\widehat{(-)}$ preserves alphaconvertibility. This is trivial. ∎

The following lemma shows that the map $\widehat{(-)}$ is a morphism of control structures.

Lemma 4.12 $\widehat{(-)}$ *preserves the operations of a control structure over any signature* $\mathcal{K}_{\mathcal{V}}$.

Proof Routine. ∎

Definition 4.13 (Terms to pSKEL) *Define the translation* $[\![-]\!] : \mathcal{T}(\mathcal{K}_{\mathcal{V}}) \rightarrow \mathsf{pSKEL}$ *to map each term constructor to the corresponding control structure operation in* pSKEL. ∎

The following proposition ensures the existence of a morphism of control structures $[\![-]\!] : \mathcal{T}(\mathcal{K}_{\mathcal{V}})/\mathsf{AC}^{\mathsf{pS}} \rightarrow \mathsf{pSKEL}$.

Lemma 4.14 *For any two terms* t_1, t_2*, whenever* $\mathsf{AC}^{\mathsf{pS}} \vdash t_1 = t_2$*, we have* $[\![t_1]\!] = [\![t_2]\!]$.

Proof Since the map is inductively defined on the operations of a reflexive control structure and the skeletal forms in pSKEL satisfy the axioms of a control structure, the result follows easily. ∎

Proposition 4.15 *The morphism (of control structures)* $\widehat{(-)}$ *from* pSKEL *to the quotient* $\mathcal{T}(\mathcal{K}_{\mathcal{V}})/\mathsf{AC}^{\mathsf{pS}}$ *is an isomorphism.*

Proof We must show both $[\![\widehat{s}]\!] = s$ and $\widehat{[\![t]\!]} = t$ for arbitrary pure skeleta s and terms t. To show that $[\![\widehat{s}]\!] = s$, consider $s = (\vec{x})\langle\vec{y}\rangle$. Then $\widehat{s} = (\vec{x})\langle\vec{y}\rangle$, where $\langle\star\rangle$ corresponds to $\boldsymbol{\nu}$. Since $[\![-]\!]$ preserves the operations of a control structure and $[\![\boldsymbol{\nu}]\!] = \langle\star\rangle$, result follows. For $\widehat{[\![t]\!]} = t$, result follows by the fact that the $[\![-]\!]$ is defined inductively on the operations and $\widehat{(-)}$ preserves all of them. ∎

Remark We note that in pSKEL, as in any action calculus, $x \in \mathbf{surf}(\langle x\rangle)$; in other words, the inequality $\mathbf{ab}_x\langle x\rangle \neq \mathbf{id} \otimes \langle x\rangle$ holds. It is worth remarking that should we add the equation $\mathbf{ab}_x\langle x\rangle = \mathbf{id} \otimes \langle x\rangle$ to the theory $\mathsf{AC}^{\mathsf{pS}}$ (making $x \notin \mathbf{surf}(\langle x\rangle)$ in the quotient of the terms by the resulting theory), all terms of equal arity would be provably equal in the resulting theory.

Lemma 4.16 *The equation* $\mathbf{ab}_x\langle x\rangle = \mathbf{id} \otimes \langle x\rangle$ *is* not *provable in the theory* $\mathsf{AC}^{\mathsf{pS}}$.

Proof We show that if such an equation were provable in $\mathsf{AC}^{\mathsf{pS}}$ then all terms would be provably equal. By proposition 4.15 we would get a contradiction, since there clearly exist pure skeleta of equal arity which are not identical. First we will show that, for any x, $\langle x\rangle = \boldsymbol{\nu}$.

$$\begin{array}{rcll}
\langle x\rangle & = & (\boldsymbol{\nu}\cdot\boldsymbol{\omega})\otimes\langle x\rangle & \\
& = & (\boldsymbol{\nu}\otimes\langle x\rangle)\cdot(\boldsymbol{\omega}\otimes\mathbf{id}) & \\
& = & \boldsymbol{\nu}\cdot(\mathbf{id}\otimes\langle x\rangle)\cdot(\boldsymbol{\omega}\otimes\mathbf{id}) & \\
& = & \boldsymbol{\nu}\cdot(\mathbf{ab}_x\langle x\rangle)\cdot(\boldsymbol{\omega}\otimes\mathbf{id}) & \text{assumption} \\
& = & \boldsymbol{\nu}\cdot(x)\langle xx\rangle\cdot(\boldsymbol{\omega}\otimes\mathbf{id}) & 2.16(4) \\
& = & (\boldsymbol{\nu}\otimes\boldsymbol{\nu})\cdot(\boldsymbol{\omega}\otimes\mathbf{id}) & \\
& = & (\boldsymbol{\nu}\cdot\boldsymbol{\omega})\otimes\boldsymbol{\nu} & \\
& = & \boldsymbol{\nu} &
\end{array}$$

Then any two terms consisting of a tensor product (of arbitrary, but finite length) of subterms $\boldsymbol{\nu}$ and $\langle x\rangle$ for any x are provably equal. Then so are terms of the form $(\vec{x})t$ and $(\vec{y})t'$ when t, t' are built from tensor product, restriction and datum, by alphaconversion. Now consider two arbitrary terms t_1, t_2 of equal arity. Then, by proposition 4.15 and the definition of $\widehat{(-)}$, there are terms $\widehat{[\![t_1]\!]}$ and $\widehat{[\![t_2]\!]}$ which have forms $(\vec{x})t$ and $(\vec{y})t'$ respectively, with t, t' built as above. Result follows immediately. ∎

4.1.3 Statics

We shall now characterise pSKEL as a terminal object in a suitable subcategory of $\mathbf{CS}^s(\mathcal{K})$. This characterisation hinges on the structure that **pskel** retains from the molecular forms; essentially, enough to account for the exported free names and enough to ensure that **pskel** is a homomorphism of control structures. Our result will also highlight a further application of surface as the semantic counterpart of free names.

Notation Consider any control structure A over a signature $\mathcal{K}_{\boldsymbol{\nu}}$. We define the *pure skeleton* of A, $\mathsf{pSKEL}(A)$ as the quotient of the smallest congruence on its actions induced by the equations:

$$\begin{aligned} \boldsymbol{\nu} \cdot \boldsymbol{\omega} &= \mathbf{id}_\epsilon \\ \boldsymbol{\nu} \cdot (x)\langle xx \rangle &= \boldsymbol{\nu} \otimes \boldsymbol{\nu} \\ K(\vec{a}) &= \boldsymbol{\omega}^m \otimes \boldsymbol{\nu}^n \end{aligned}$$

We shall call the unique morphism which takes any action in A to its equivalence class in $\mathsf{pSKEL}(A)$, $\mathbf{pskel}_A$.

Until otherwise stated, in what follows we shall assume that the reaction relation for pSKEL is the universal relation on its arrows.

Lemma 4.17 *For any control structure A over some $\mathcal{K}_{\boldsymbol{\nu}}$ in which $x \in \mathbf{surf}(\langle x \rangle)$ and the following equations hold:*

$$\begin{aligned} \boldsymbol{\nu} \cdot \boldsymbol{\omega} &= \mathbf{id}_\epsilon \\ K(\vec{a}) &= \boldsymbol{\omega}^m \otimes \boldsymbol{\nu}^n \\ \boldsymbol{\nu} \cdot (x)\langle xx \rangle &= \boldsymbol{\nu} \otimes \boldsymbol{\nu} \end{aligned}$$

there is a unique morphism from pSKEL *to A mapping each $\langle \star \rangle$ in* pSKEL *to $\boldsymbol{\nu}_p$ in A. This morphism is injective.*

Proof We know that AC is equipotent to a purely equational theory on the term algebra (over any signature, including $\mathcal{K}_{\boldsymbol{\nu}}$). Therefore, the theory $\mathsf{AC}^{\mathsf{pS}}$ is also equipotent to a purely equational theory, and by a standard argument we obtain that there is a unique morphism of control structures from pSKEL to any such A. It remains to show that this morphism $\Phi : \mathsf{pSKEL} \to A$ is injective. First we shall show that in A, for any x, y: (1) $\langle x \rangle \neq \boldsymbol{\nu}$; and (2) if $x \neq y$ then $\langle x \rangle \neq \langle y \rangle$. (1) follows since the surface of $\boldsymbol{\nu}$ is necessarily empty (it is empty in the action calculus, there is a morphism from the action calculus to A and morphisms do not increase surface). (2) follows immediately since the surfaces of $\langle x \rangle$ and $\langle y \rangle$ are not equal. Consider arbitrary $s_1, s_2 \in \mathsf{pSKEL}$ such that $s_1 \neq s_2$. We show that $\Phi(s_1) \neq \Phi(s_2)$. It suffices to consider s_1, s_2 of identical arity (otherwise the proof is trivial). Assume that $\Phi(s_1) = \Phi(s_2)$. By alphaconversion we know, for some $\vec{x} \subseteq X$ and $\vec{u}, \vec{v} \subseteq X \cup \{\star\}$ that $s_1 = \vec{x}\langle\vec{u}\rangle$ and $s_2 = \vec{x}\langle\vec{v}\rangle$. Assume $\vec{u}$ and $\vec{v}$ differ in some position such that $\vec{u} = \vec{y}w_1\vec{y}_1$ and $\vec{v} = \vec{y}w_2\vec{y}_2$ with $\vec{y} : m$, $w_1, w_2 : p$ and $w_1 \neq w_2$. Now choose some distinct names $\vec{z}$ such that $\{\vec{z}\} \cap \mathbf{fn}\,(s_1, s_2) = \emptyset$. Clearly, $\{\vec{z}/\vec{x}\}w_1 \neq \{\vec{z}/\vec{x}\}w_2$. Now, for $i = 1, 2$:

$$\langle\vec{z}\rangle \cdot \Phi(s_i) \cdot (\boldsymbol{\omega}^m \otimes \mathbf{id}_p \otimes \mathbf{id}) = \langle\{\vec{z}/\vec{x}\}w_i\rangle$$

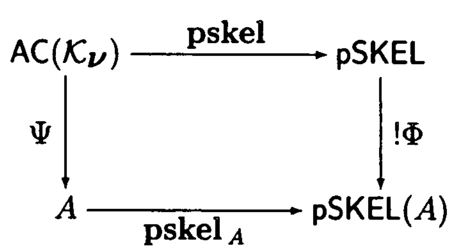
Then $\langle\{\vec{z}/\vec{x}\}w_1\rangle = \langle\{\vec{z}/\vec{x}\}w_2\rangle$. But by $\{\vec{z}/\vec{x}\}w_1 \neq \{\vec{z}/\vec{x}\}w_2$ this is a contradiction. ∎

Lemma 4.18 *For any control structure over $\mathcal{K}_{\boldsymbol{\nu}}$, A, such that $x \in$ **surf**$(\langle x\rangle)$, there exists a unique injective morphism from* pSKEL *to* pSKEL(A) *(in* $\mathbf{CS}^s(\mathcal{K}_{\boldsymbol{\nu}})$*).*

Proof By lemma 4.17, we need only show that whenever $x \in \mathbf{surf}\,(\langle x\rangle)$ in A, then $x \in \mathbf{surf}\,(\langle x\rangle)$ in pSKEL(A). By lemma 4.16 the result follows. ∎

Theorem 4.19 pSKEL *is terminal in the full subcategory of* $\mathbf{CS}^s(\mathcal{K}_{\boldsymbol{\nu}})$ *whose objects are just those control structures to which the unique morphism from* AC$(K_{\boldsymbol{\nu}})$ *is onto and in which* $x \in \mathbf{surf}\,(\langle x\rangle)$.

Proof First we note that the following diagram commutes in the subcategory:



To see this consider that there is a unique morphism from AC$(\mathcal{K}_{\boldsymbol{\nu}})$ to pSKEL(A). We shall now show that Φ : pSKEL $\to$ pSKEL(A) is onto. This will conclude the proof, since by lemma 4.18, Φ is injective. This would make pSKEL and pSKEL(A) isomorphic and since there is a unique morphism from any A to pSKEL(A), the result follows. To show that Φ is onto, we need

$$\forall \mathsf{s} \in \mathsf{pSKEL}(A).\exists s \in \mathsf{pSKEL}.\ \Phi(s) = \mathsf{s}$$

Assume not; that is, there is some $\mathsf{s} \in \mathsf{pSKEL}(A)$ for which there is no $s \in \mathsf{pSKEL}$ such that $\Phi(s) = \mathsf{s}$. Now, since $\mathbf{pskel}_A$ and Ψ are onto, there is some $\mathsf{a} \in A$ such that $\mathbf{pskel}_A(\mathsf{a}) = \mathsf{s}$ and some $a \in \mathsf{AC}(\mathcal{K}_{\boldsymbol{\nu}})$ such that $\Psi(a) = \mathsf{a}$. Hence $\mathsf{s} = \mathbf{pskel}_A(\Psi(a))$. Now let $s = \mathbf{pskel}(a)$. We get $\Phi(\mathbf{pskel}(a)) = \mathbf{pskel}_A(\Psi(a)) = \mathsf{s}$ which gives a contradiction. ∎

Remark We can prove an analogous result concerning the terminality of pSKEL in the full subcategory of $\mathbf{CS}^{\mathsf{rs}}(\mathcal{K})$ (and of $\mathbf{CS}^{\mathsf{res}}(\mathcal{K})$) whose objects are just those reflexive control structures to which the unique morphism from $\mathsf{AC}^{\mathsf{rs}}(\mathcal{K})$ (and, respectively, $\mathsf{AC}^{\mathsf{res}}(\mathcal{K})$) is onto and in which $x \in \mathbf{surf}(\langle x \rangle)$.

4.1.4 Dynamics

We shall now consider pSKEL as a classifier of action calculi. Recall that $\mathbf{pskel} : \mathsf{AC}(\mathcal{K}) \to \mathsf{pSKEL}$ captures the potential of an action to export free names. But so far we have only considered the statics of pSKEL, using the universal relation on its actions as its reaction relation to ensure that any map to it from any control structure trivially preserves the reaction relation. We shall now choose a smaller reaction relation, which will give pSKEL its power as a classifier of dynamics.

The intuition behind what follows relies on the property that whenever an action a reacts to, say, a', then a' should have at least as many exported free names as a had. In other words, reaction can only add exported free names but never retract them. Whether this condition on reaction is one we would wish or expect computational calculi to have universally is not known; however, in all the examples (available to date) of existing computational calculi cast in the action calculi mould, this property does hold. This is not to say that stronger properties do not; indeed, in the following section we will examine what is, in a sense, a stronger form of this property.

There are three equivalent characterisations of the reaction relation on pSKEL all of which provide an elegant way of defining it. We choose to define reaction on the molecular forms.

Definition 4.20 (Pure skeleta: dynamics) *The relation* $\searrow$ *on* pSKEL *is the transitive reflexive closure of the smallest relation such that, for any skeleton* $s = (\vec{x})\langle\vec{y}_1 \star \vec{y}_2\rangle$ *and name* z*:*

$$(\vec{x})\langle\vec{y}_1 \star \vec{y}_2\rangle \searrow (\vec{x})\langle\vec{y}_1 z \vec{y}_2\rangle$$ ■

We must show that the relation we have defined is indeed a reaction relation.

Proposition 4.21 *The relation* $\searrow$ *on* pSKEL *is preserved by the operations of an action structure together with reflexion.*

Proof Mostly routine; we shall show the most interesting case, that for reflexion. Assume $s = (u\vec{x})\langle v\vec{y}\rangle$. Then, if $s \searrow s'$, $s' = (u\vec{x})\langle v'\vec{y}'\rangle$, where $v'\vec{y}'$ is obtained by replacing in $v\vec{y}$ some number of occurrences of $\star$ by names.

Case 1: $u = v$ Then, since $u \in X$, $v \neq \star$ and hence $v = v'$. We get $\uparrow s = (\vec{x})\langle\{\star/u\}\vec{y}\rangle$ and $\uparrow s' = (\vec{x})\langle\{\star/u\}\vec{y}'\rangle$. Since $u \neq \star$, any occurrence of u in $\vec{y}$ indicates a corresponding occurrence in $\vec{y}'$. Hence any occurrences of $\star$ introduced in $\vec{y}$ by the substitution $\{\star/u\}$ are also introduced (in the corresponding places) in $\vec{y}'$.

Case 2: $u \neq v$

Case 2.1: $v \neq \star$ Then $v = v'$ and $\uparrow s = (\vec{x})\langle\{v/u\}\vec{y}\rangle$ and $\uparrow s' = (\vec{x})\langle\{v/u\}\vec{y}'\rangle$. By the same reasoning as for the previous case, any occurrences of v introduced in $\vec{y}$ by the substitution $\{v/u\}$ are also introduced (in the corresponding places) in $\vec{y}'$.

Case 2.2: $v = \star$, $v' \neq u$ Then $\uparrow s = (\vec{x})\langle\{\star/u\}\vec{y}\rangle$ and $\uparrow s' = (\vec{x})\langle\{v'/u\}\vec{y}'\rangle$. Thus, any $\star$ introduced in $\vec{y}$ by the substitution $\{\star/u\}$ is replaced by v'.

Case 2.3: $v = \star$, $v' = u$ Then $\uparrow s = (\vec{x})\langle\{\star/u\}\vec{y}\rangle$ and $\uparrow s' = (\vec{x})\langle\{\star/u\}\vec{y}'\rangle$. Thus, since the occurrences of u in $\vec{y}$ are unchanged in $\vec{y}'$, the result follows. ■

The following proposition captures the essence of reaction for pSKEL: $\langle\star\rangle$ may react to become a datum.

Proposition 4.22 *The relation* $\searrow$ *is the smallest reaction relation on* pSKEL *closed under the following rule:*

$$\boldsymbol{\nu} \searrow \langle x \rangle$$

Proof The reaction $\boldsymbol{\nu} \searrow \langle x \rangle$ is clearly derivable by the rule given in definition 4.20. The reaction

$$(\vec{x})\langle \vec{y}_1 \star \vec{y}_2 \rangle \searrow (\vec{x})\langle \vec{y}_1 z \vec{y}_2 \rangle$$

is derivable since $\widehat{(\vec{x})\langle \vec{y}_1 \star \vec{y}_2 \rangle} = (\vec{x})(\langle \vec{y}_1 \rangle \otimes \boldsymbol{\nu} \otimes \langle \vec{y}_2 \rangle)$. Since reaction is preserved by tensor and abstraction

$$(\vec{x})(\langle \vec{y}_1 \rangle \otimes \boldsymbol{\nu} \otimes \langle \vec{y}_2 \rangle) \searrow (\vec{x})(\langle \vec{y}_1 \rangle \otimes \langle z \rangle \otimes \langle \vec{y}_2 \rangle) = \widehat{(\vec{x})\langle \vec{y}_1 z y_2 \rangle}$$ ■

It is the following, logical form of characterisation that we shall use to demonstrate the role of pSKEL as a classifier of dynamics. The proposition expresses our intuition about the retention of any exported free name under reaction. The imported names, which may be replaced by free names as a result of precomposition by data, are also taken in account.

Proposition 4.23 *For any two skeleta s_1 and s_2 of identical arities, $s_1 \searrow s_2$ if and only if, for all $z \neq \star, \vec{x}, m,\ s_1'$,*

$$\langle \vec{x} \rangle \cdot s_1 = s_1' \cdot (\mathbf{id}_m \otimes \langle z \rangle \otimes \mathbf{id}) \ \Rightarrow\ \exists s_2'. \langle \vec{x} \rangle \cdot s_2 = s_2' \cdot (\mathbf{id}_m \otimes \langle z \rangle \otimes \mathbf{id})$$

Proof ($\Longrightarrow$) Assume $s_1 \searrow s_2$ and $\langle \vec{x} \rangle \cdot s_1 = s_1' \cdot (\mathbf{id}_m \otimes \langle z \rangle \otimes \mathbf{id})$. Then, $\langle \vec{x} \rangle \cdot s_1 \searrow \langle \vec{x} \rangle \cdot s_2$. Now, for some $\vec{y}_1, \vec{y}_2 \subseteq X \cup \{\star\}$ with $\vec{y}_1 : m$, $\langle \vec{x} \rangle \cdot s_1 = \langle \vec{y}_1 z \vec{y}_2 \rangle = \langle \vec{y}_1 \vec{y}_2 \rangle \cdot (\mathbf{id}_m \otimes \langle z \rangle \otimes \mathbf{id})$. Since $z \neq \star$, it follows by the reaction rule on molecular forms that $\langle \vec{x} \rangle \cdot s_2 = \langle \vec{y}_1' z \vec{y}_2' \rangle$ where $\vec{y}_1'$ and $\vec{y}_2'$ are obtained by replacing some occurrences of $\star$ by some names in $\vec{y}_1$ and $\vec{y}_2$ respectively. Hence $\langle \vec{x} \rangle \cdot s_2 = \langle \vec{y}_1' z \vec{y}_2' \rangle = \langle \vec{y}_1' \vec{y}_2' \rangle \cdot (\mathbf{id}_m \otimes \langle z \rangle \otimes \mathbf{id})$.

($\Longleftarrow$) Consider an arbitrary $s_1 = (\vec{x})\langle\vec{u}\rangle$ where $\vec{y} \subseteq X \cup \{\star\}$. By alphaconversion for any s_2 of identical arity, $s_2 = (\vec{x})\langle\vec{v}\rangle$ for some $\vec{v}$. It suffices to show that whenever s_2 satisfies this condition, then it can be obtained from s_1 by replacing some number of occurrences of $\star$ in the skeletal form of s_1 by some names. Assume not. Then there is some name w ($\in X$) such that $\vec{u} = \vec{u}_1 w \vec{u}_2$ which is not equal to the corresponding name in $\vec{v}$. But this is easily shown to violate the property of s_2 regarding the identical provision of exported names under precomposition by arbitrary data. ∎

Lemma 4.24 *Let A be a control structure with a morphism to* pSKEL *such that for all $a \in A$, whenever* $\mathbf{pskel}(a) = s \cdot (\mathbf{id}_m \otimes \langle z\rangle \otimes \mathbf{id})$ *then, for some $a' \in A$, $a = a' \cdot (\mathbf{id}_m \otimes \langle z\rangle \otimes \mathbf{id})$.*

For any $a, a' \in A$ such that $a = a' \cdot (\mathbf{id}_m \otimes \langle z\rangle \otimes \mathbf{id})$ and $a \searrow b$, there exists some b' such that $b = b' \cdot (\mathbf{id}_m \otimes \langle z\rangle \otimes \mathbf{id})$

Proof By $a = a' \cdot (\mathbf{id}_m \otimes \langle z\rangle \otimes \mathbf{id})$ and the fact the **pskel** preserves the operations of a control structure we have, in pSKEL, $\mathbf{pskel}(a) = \mathbf{pskel}(a') \cdot (\mathbf{id}_m \otimes \langle z\rangle \otimes \mathbf{id})$. Choose $\vec{x}$ which is distinct from any names in the surfaces of a and b (and therefore, z). Now, $\langle\vec{x}\rangle \cdot \mathbf{pskel}(a) = \langle\vec{x}\rangle \cdot \mathbf{pskel}(a') \cdot (\mathbf{id}_m \otimes \langle z\rangle \otimes \mathbf{id})$. Since $a \searrow b$ implies $\mathbf{pskel}(a) \searrow \mathbf{pskel}(b)$, by lemma 4.23 we get $\mathbf{pskel}(\langle\vec{x}\rangle \cdot b) = \langle\vec{x}\rangle \cdot \mathbf{pskel}(b) = s \cdot (\mathbf{id}_m \otimes \langle z\rangle \otimes \mathbf{id})$, for some s. By assumption, there is some b' such that $\langle\vec{x}\rangle \cdot b = b' \cdot (\mathbf{id}_m \otimes \langle z\rangle \otimes \mathbf{id})$. Then, abstracting by $\vec{x}$ on either side of this equation gives $b = (\vec{x})b' \cdot (\mathbf{id}_m \otimes \langle z\rangle \otimes \mathbf{id})$. ∎

Remark In the above it is easily shown that $a' \searrow b'$ by applying the context $[\_] \cdot (\mathbf{id}_m \otimes \omega \otimes \mathbf{id})$ to both sides of $a \searrow b$.

We are now in a position to state our main result concerning pSKEL as a classifier of dynamics:

Theorem 4.25 *For any signature $\mathcal{K}$ and reaction rules $\mathcal{R}$, the action calculus* $\mathsf{AC}(\mathcal{K}, \mathcal{R})$ *has a morphism of control structures to* pSKEL *if and only if for all*

$a, a', b \in \mathsf{AC}(\mathcal{K}, \mathcal{R})$ and name $z \neq \star$, whenever $a = a' \cdot (\mathbf{id}_m \otimes \langle z \rangle \otimes \mathbf{id})$ and $a \searrow b$, then, for some b', $b = b' \cdot (\mathbf{id}_m \otimes \langle z \rangle \otimes \mathbf{id})$ and $a' \searrow b'$.

Proof ($\Longrightarrow$) By lemma 4.24, it suffices to show that the morphism **pskel** : $\mathsf{AC}^r(\mathcal{K}, \mathcal{R}) \rightarrow \mathsf{pSKEL}$ has the property

> for all $a \in \mathsf{AC}^r(\mathcal{K}, \mathcal{R})$, whenever $\mathbf{pskel}\,(a) = s \cdot (\mathbf{id}_m \otimes \langle z \rangle \otimes \mathbf{id})$ then, for some a', $a = a' \cdot (\mathbf{id}_m \otimes \langle z \rangle \otimes \mathbf{id})$.

By proposition 4.4 the result follows immediately.

($\Longleftarrow$) We know that there is a (unique) morphism **pskel** in $\mathbf{CS}^s(\mathcal{K})$ from $\mathsf{AC}^s(\mathcal{K})$ to pSKEL. It therefore suffices to show that **pskel** preserves the reaction relation. Assume $a \searrow b$. Now by proposition 4.23, we need just show that, for any $\vec{x}$ and s_1 whenever $\langle \vec{x} \rangle \cdot \mathbf{pskel}\,(a) = s_1 \cdot (\mathbf{id}_m \otimes \langle z \rangle \otimes \mathbf{id})$, then for some s_2, $\langle \vec{x} \rangle \cdot \mathbf{pskel}\,(b) = s_2 \cdot (\mathbf{id}_m \otimes \langle z \rangle \otimes \mathbf{id})$. But, by proposition 4.4, $\langle \vec{x} \rangle \cdot \mathbf{pskel}\,(a) = s_1 \cdot (\mathbf{id}_m \otimes \langle z \rangle \otimes \mathbf{id})$ implies that, for some a', $\langle \vec{x} \rangle \cdot a = a' \cdot (\mathbf{id} \otimes \langle z \rangle \otimes \mathbf{id})$. By assumption, and since $\langle \vec{x} \rangle \cdot a \searrow \langle \vec{x} \rangle \cdot b$, there is some b' such that $\langle \vec{x} \rangle \cdot b = b' \cdot (\mathbf{id}_m \otimes \langle z \rangle \otimes \mathbf{id})$. This clearly implies, $\langle \vec{x} \rangle \cdot \mathbf{pskel}\,(b) = \mathbf{pskel}\,(b') \cdot (\mathbf{id}_m \otimes \langle z \rangle \otimes \mathbf{id})$; hence choosing $\mathbf{pskel}\,(b')$ as s_2 gives the result. ■

Remark As intimated previously, by replacing $\mathbf{CS}^s(\mathcal{K})$ and $\mathsf{AC}^s(\mathcal{K})$ respectively by $\mathbf{CS}^{rs}(\mathcal{K})$ and $\mathsf{AC}^{rs}(\mathcal{K})$, and even by $\mathbf{CS}^{res}(\mathcal{K})$ and $\mathsf{AC}^{res}(\mathcal{K})$, in the statement of the above theorem, we obtain valid theorems. There is however an interesting difference in the morphism **pskel** in each case: for action calculi, there is no guarantee that this morphism, if it exists, is onto (it will depend on the signature $\mathcal{K}$), whereas for the reflexive variants this is always the case.

Discussion Since the existence of a morphism to pSKEL is constrained by the reaction relation of an action calculus; and the same reaction relation depends on the reaction rules $\mathcal{R}$ of the action calculus, it is natural that one should ask which kinds of reaction rule permit and prohibit the existence of such a morphism. It is clear that reaction rules having any of the following forms

$$a \cdot (\mathbf{id} \otimes \langle x \rangle \otimes \mathbf{id}) \searrow K(\vec{b})$$
$$a \cdot (\mathbf{id}_m \otimes \langle x \rangle \otimes \mathbf{id}) \searrow b \cdot (\mathbf{id}_m \otimes \langle y \rangle \otimes \mathbf{id}) \quad (x \neq y)$$

will ensure that no morphism from the action calculus to pSKEL can exist. Conversely, in any action calculus which has a morphism to pSKEL, such rules—indeed, such reactions—are absent. However, a morphism to pSKEL does permit an action calculus to have rules, and reactions, such as the ones shown below:

$$K(\vec{a}) \cdot (\mathbf{id} \otimes (x)\langle xx \rangle \otimes \mathbf{id}) \searrow K'(\vec{b})$$
$$K(\vec{a}) \cdot (\mathbf{id}_m \otimes (x)\langle xx \rangle \otimes \mathbf{id}) \searrow b \cdot (\mathbf{id}_m \otimes \langle yz \rangle \otimes \mathbf{id}) \quad (y \neq z)$$

In both of these examples the identity of the two control bound exported names is lost as a result of reaction. In the first, the loss is to distinct *control bound* names; whereas in the second, distinct *free* names take the position of the identical control bound names. If we want to think of the controls as computational entities which may, upon involvement in computational activity, supply names into the links they command (through binding originating from the control), then such behaviour as display by the above reactions is not acceptable.

4.2 Restriction Skeleta

The intuition behind composition as connection of dataflow channels poses an important question: what can be said to flow through such channels. One simple answer is that it is the names which flow; this is indeed corroborated by the definition of composition for the molecular forms for action calculi. It is worth noting that both free and bound names flow in this way, and therefore, the exclusive consideration of the exported *free* names is flawed if we wish to account for the flow of names (free and bound) through dataflow channels in our semantic treatment of action calculi.

As an illustration of why an exclusive consideration of free (and import-bound) exported names might not be enough, consider the actions $\boldsymbol{\nu} \cdot (x)\langle xx\rangle$ and $\boldsymbol{\nu} \otimes \boldsymbol{\nu}$. As we have seen, both of these actions have the same pure skeleton. We can show that these actions, say in PIC, may cause different behaviour when precomposed to certain actions. One such action which reveals this difference is $(uv)(\mathbf{out}_u \otimes \mathbf{box}_v a)$. Precomposing this action by $\boldsymbol{\nu} \cdot (x)\langle xx\rangle$ unifies the port names parameterised by u and v causing a potential reaction to $(\boldsymbol{\nu}x)(\{^{xx}/_{uv}\}a)$. On the other hand, precomposing the same action by $\boldsymbol{\nu} \otimes \boldsymbol{\nu}$ results in an action which is *inert*, that is, cannot perform further computation.

4.2.1 Skeletal forms

We shall diverge just enough from pure skeleta in order to introduce a distinction between $\boldsymbol{\nu} \cdot (x)\langle xx\rangle$ and $\boldsymbol{\nu} \otimes \boldsymbol{\nu}$. This involves having some means of expressing those bindings which originate from molecules; we do not want to distinguish between the molecules themselves, but only between the bound names originating from them. All that is required in order to achieve this, is some family of particles (molecules of rank 0) whose input arities are all ϵ and whose output arities cover all the primes. This allows the skeletal form of a molecule to be constructed from discard operations (to make up the input arity) and such particles (to make up the output arity). Indeed, we have already encountered such a family of particles: the restriction particles.

Definition 4.26 (Restriction skeleta) *The actions of* restriction skeleta $\boldsymbol{\nu}$SKEL, *ranged over by s have the following form:*

$$s \ ::= \ (\vec{x})\boldsymbol{\nu}S\langle\vec{z}\rangle$$

where $S \subseteq \{\vec{z}\}$. The names $\vec{x}$ and S are all distinct and are binding occurrences; each name in $\vec{z}$ is free unless bound by one of the binding occurrences. ∎

Remark The constraint $S \subseteq \{\vec{z}\}$ in the above definition expresses our requirement to enhance pure skeleta *just enough* to allow the representation of control bound names: names in S which do not bind any name in $\vec{z}$ do not assist in such representation and are therefore, at least, superfluous to our aim.

When we wish to indicate that S is the set of names present in the vector $\vec{y}$ we shall often write $\tilde{y}$. We shall now show that restriction skeleta are strict reflexive control structures and also that they are isomorphic to the quotient of the term algebra $\mathcal{T}(\mathcal{K}_{\boldsymbol{\nu}})$ and the theory AC together with the equations $\boldsymbol{\nu} \cdot \boldsymbol{\omega} = \mathbf{id}_\epsilon$ and $K(\vec{t}) = \boldsymbol{\omega}^m \otimes \boldsymbol{\nu}^n$ for each $K \in \mathcal{K}$.

Definition 4.27 *We define the following operations on* $\boldsymbol{\nu}$SKEL. *Assume* $s_1 = (\vec{u})(\boldsymbol{\nu} S_1)\langle\vec{v}\rangle$, $s_2 = (\vec{x})(\boldsymbol{\nu} S_2)\langle\vec{y}\rangle$ *and* $s = (y\vec{u})(\boldsymbol{\nu} S)\langle x\vec{v}\rangle$ *with the names in* $\vec{u}$, $\vec{x}$, S_1 *and* S_2 *distinct.*

$$\begin{aligned} \mathbf{id}_m &\stackrel{\text{def}}{=} (\vec{x})\langle\vec{x}\rangle \quad (\vec{x} : m) \\ \langle x\rangle &\stackrel{\text{def}}{=} (\,)\langle x\rangle \\ \boldsymbol{\omega} &\stackrel{\text{def}}{=} (x)\langle\,\rangle \end{aligned}$$

$$\begin{aligned} s_1 \cdot s_2 &\stackrel{\text{def}}{=} (\vec{u})\boldsymbol{\nu}((S_1 \cup S_2) \cap \{\sigma\vec{y}\})\langle\sigma\vec{y}\rangle \qquad\qquad (\sigma = \{\vec{v}/\vec{x}\}) \\ s_1 \otimes s_2 &\stackrel{\text{def}}{=} (\vec{u}\vec{x})\boldsymbol{\nu}(S_1 \cup S_2)\langle\vec{v}\vec{y}\rangle \\ \mathbf{ab}_x s_1 &\stackrel{\text{def}}{=} (x\vec{u})\boldsymbol{\nu} S_1\langle x\vec{v}\rangle \\ \uparrow s &\stackrel{\text{def}}{=} \begin{cases} (\vec{u})\boldsymbol{\nu}(S \cap \{\{y/x\}\vec{v}\})\langle\{y/x\}\vec{v}\rangle & \text{if } x \neq y \\ (\vec{u})\boldsymbol{\nu}((S \cup \{x\}) \cap \{\vec{v}\})\langle\vec{v}\rangle & \text{if } x = y \end{cases} \end{aligned}$$

■

Proposition 4.28 *For any set of controls* $\mathcal{K}$, $\boldsymbol{\nu}$SKEL *together with the operations of Definition 4.27, any reaction relation on* $\boldsymbol{\nu}$SKEL *and, for any* $K \in \mathcal{K}$,

$$K(\vec{s}) \stackrel{\text{def}}{=} (\vec{x})\boldsymbol{\nu}\tilde{y}\langle\vec{y}\rangle$$

is a strict reflexive control structure over $\mathcal{K}$.

Proof Consider the molecular forms over the strict reflexive action calculus $\mathsf{AC}^{\mathsf{r}\epsilon}(\mathcal{K})$. We define the map $\boldsymbol{\nu}\mathbf{skel} : \mathsf{AC}^{\mathsf{r}\epsilon}(\mathcal{K}) \to \boldsymbol{\nu}$SKEL as follows: for each $a \in \mathsf{AC}^{\mathsf{r}\epsilon}(\mathcal{K})$ with molecular form $(\vec{u})\vec{\mu}(\vec{v})\langle\vec{w}\rangle$, $\boldsymbol{\nu}\mathbf{skel}(a) \stackrel{\text{def}}{=} (\vec{u})\boldsymbol{\nu} S\langle\vec{w}\rangle$ where $S =$

$\{\vec{v}\} \cap \{\vec{w}\}$. Clearly, ν**skel** is onto. It therefore suffices to show that ν**skel** preserves the operations of a strict reflexive control structure. ■

Remark The mapping $\nu\mathbf{skel} : (\vec{u})\vec{\mu}(\vec{v})\langle\vec{w}\rangle \mapsto (\vec{u})(\nu S)\langle\vec{w}\rangle$ where $S = \{\vec{v}\} \cap \{\vec{w}\}$, is well-defined for the molecular forms of action calculi and also of its reflexive variants. In all these cases the mapping preserves the (non-control) operations of a control structure, and in the case of the reflexive variants, preserves reflexion as well.

The proposition below states that the skeletal form of any control in νSKEL may be built from discard and restriction operations.

Proposition 4.29 *In νSKEL, for any K,*

$$K(\vec{s}) = \omega^m \otimes \nu^n$$

Proof By inspection of the molecular forms. ■

Remark We note that in νSKEL, $\nu \otimes \nu \neq \nu \cdot (x)\langle xx\rangle$. However, we shall define a dynamics for νSKEL where $\nu \otimes \nu$ may react to $\nu \cdot (x)\langle xx\rangle$.

4.2.2 Terms

We shall now give a characterisation of νSKEL as a quotient of the terms $\mathcal{T}(\mathcal{K}_\nu)$.

Definition 4.30 (The theory AC$^{\nu S}$) *Let AC$^{\nu S}$ be the theory resulting from the addition of the following equations to the theory AC:*

$$\begin{aligned} \nu \cdot \omega &= \mathbf{id}_\epsilon \\ K(\vec{t}) &= \omega^m \otimes \nu^n \end{aligned}$$

■

Definition 4.31 (νSKEL to Terms) *Define the translation $\widehat{(-)} : \nu\mathsf{SKEL} \to \mathcal{T}(\mathcal{K}_\nu)$ as follows:*

$$\widehat{(\vec{x})\nu\vec{y}\langle\vec{z}\rangle} \stackrel{\text{def}}{=} (\vec{x})(\nu\vec{y})\langle\vec{z}\rangle$$ ■

Lemma 4.32 *The translation* $\widehat{(-)} : \nu\mathsf{SKEL} \to \mathcal{T}(\mathcal{K}_\nu)/\mathsf{AC}^{\nu\mathsf{S}}$ *is well-defined.*

Proof The translation $\widehat{(-)}$ preserves alphaconvertibility. To show that $\widehat{(-)}$ preserves the permutation of restriction bound names it suffices to show that $(\nu x)(\nu y)t = (\nu y)(\nu x)t$ is provable in the theory $\mathsf{AC}^{\nu\mathsf{S}}$. Assume $x : p, y : q$:

$$\begin{aligned}
(\nu x)(\nu y)t &= (\nu \otimes \mathbf{id}) \cdot (x)((\nu \otimes \mathbf{id}) \cdot (y)t) \\
&= (\nu \otimes \mathbf{id}) \cdot \mathbf{ab}_x(\nu \otimes \mathbf{id}) \cdot (x)(y)t \\
&= (\nu \otimes \mathbf{id}) \cdot (\mathbf{id}_p \otimes \nu \otimes \mathbf{id}) \cdot (x)(y)t && (x \notin \mathbf{fn}(\nu)) \\
&= (\nu \otimes \nu \otimes \mathbf{id}) \cdot (x)(y)t \\
&= (\nu \otimes \nu \otimes \mathbf{id}) \cdot (\mathbf{p}_{p,q} \otimes \mathbf{id}) \cdot (y)(x)t \\
&= (\nu \otimes \nu \otimes \mathbf{id}) \cdot (y)(x)t \\
&= (\nu \otimes \mathbf{id}) \cdot (\mathbf{id}_q \otimes \nu \otimes \mathbf{id}) \cdot (y)(x)t && (y : q) \\
&= (\nu \otimes \mathbf{id}) \cdot \mathbf{ab}_y(\nu \otimes \mathbf{id}) \cdot (y)(x)t && (y \notin \mathbf{fn}(\nu)) \\
&= (\nu \otimes \mathbf{id}) \cdot (y)((\nu \otimes \mathbf{id}) \cdot (x)t)
\end{aligned}$$

■

Lemma 4.33 $\widehat{(-)}$ *preserves the operations of a control structure over any signature* $\mathcal{K}_\nu$.

Proof Routine. ■

Definition 4.34 (Terms to νSKEL) *Define the translation* $[\![\_]\!] : \mathcal{T}(\mathcal{K}_\nu) \to \nu\mathsf{SKEL}$ *to map each constructor to the corresponding operation in* $\nu\mathsf{SKEL}$. ■

Lemma 4.35 *For any two terms* t_1, t_2, *whenever* $\mathsf{AC}^{\nu\mathsf{S}} \vdash t_1 = t_2$, *we have* $[\![t_1]\!] = [\![t_2]\!]$.

Proof Since the map is inductively defined on the operations of a control structure and the skeletal forms in νSKEL satisfy the axioms of $\mathsf{AC}^{\nu\mathsf{S}}$ (by propositions 4.28 and 4.29), the result follows. ■

Proposition 4.36 *The morphism (of control structures) from νSKEL to the quotient $\mathcal{T}(\mathcal{K}_{\nu})/\mathsf{AC}^{\nu\mathsf{S}}$ is an isomorphism.*

Proof We must show both $[\![\hat{s}]\!] = s$ and $\widehat{[\![t]\!]} = t$ for arbitrary pure skeleta s and terms t. For $[\![\hat{s}]\!] = s$, consider $s = (\vec{x})(\nu\vec{y})\langle\vec{z}\rangle$; then $\hat{s} = (\vec{x})(\nu\vec{y})\langle\vec{y}\rangle$. Since $[\![-]\!]$ preserves the operations of a control structure the result follows. For $\widehat{[\![t]\!]} = t$, result follows by the fact that the $[\![-]\!]$ is defined inductively on the operations and $\widehat{(-)}$ preserves all of them. ∎

4.2.3 Effect structures

The notion of *effect*, introduced by Milner in [21] in the context of action structures, provides an abstract description of what entities might be said to flow through dataflow channels. *Effects*, ranged over by e, are defined in terms of the static and dynamic properties of the factorisations (a', e) of each action $a = a' \cdot e$. Essentially, an effect is a *spent* action, one which cannot carry out further computation no matter what "information" it may receive. It may, on the other hand, supply "information" to some other action, causing it to react. These dynamic characteristics are captured by the following definition of *inertia*:

Definition 4.37 (Inertia) *An action a is* inert *if, whenever $b \cdot a \searrow c$, there exists some b' such that $b \searrow b'$ and $c = b' \cdot a$.* ∎

Effects are required to be inert. This, together with the property that a set of effects is closed under composition will allow effects produced by successive reactions to accumulate, thus:

$$a \searrow a' \cdot e \searrow (a'' \cdot e') \cdot e = a'' \cdot (e' \cdot e)$$

The set of effects is required to be closed under the action structure operations. While it is clearly desirable for effects to be closed under composition (if effects are to accumulate), it is debatable whether closure under abstraction is justified in the abstract definition.

Definition 4.38 (Effect structure) *Let A be an action structure, and E a static sub-actionstructure of A. Then E is a* postcomponent *of A if, whenever $a = a_1 \cdot e_1 = a_2 \cdot e_2$, (with $e_1, e_2 \in E$) then for some a' and $e'_1, e'_2 \in E$*

$$a_i = a' \cdot e'_i \quad (i = 1, 2) \text{ and } e'_1 \cdot e_1 = e'_2 \cdot e_2$$

If all the actions in E are inert, then E is an effect structure *for A.* ∎

Remark Our definition of postcomponent differs slightly from the one in [21]. We require that a prospective postcomponent E be a *static* sub-actionstructure rather than a sub-actionstructure of A. This is justified since the notion of postcomponent is inherently a static one— the notion is of relevance even in the absence of any dynamics.

Consider some postcomponent E of A (by our definition) whose reaction relation is the identity relation (i.e. E is effectively a static action structure). Then, if the (images of the) actions of E are inert in A it will also be a sub-actionstructure of A. To see why, consider the injective homomorphism of static action structures $\Phi : E \to A$. We can show that whenever $\Phi e \searrow \Phi e'$ then $\Phi e = \Phi e'$. Assume $\Phi e \searrow \Phi e'$; then clearly, $\mathbf{id} \cdot \Phi e \searrow \Phi e'$. Hence, by inertia, there is some $a \in A$ such that $\mathbf{id} \searrow a$ and $\Phi e' = a \cdot \Phi e$. But, by definition $\mathbf{id} \searrow a$ implies $a = \mathbf{id}$, hence $\Phi e' = \Phi e$. Now, since Φ is injective, $e = e'$. We can now show that the extra condition required for a static sub-actionstructure to be a sub-actionstructure is satisfied; namely that

$$\Phi e \searrow \Phi e' \iff e \searrow e'$$

Since $\Phi e \searrow \Phi e'$ implies $e = e'$, we have $e \searrow e'$ by the reflexivity of reaction. For the other direction, the homomorphism Φ trivially preserves the identity relation on E, again by reflexivity of reaction.

The following definitions lead to a technique for showing that certain static sub-actionstructures are postcomponents.

Definition 4.39 *Let E be a static sub-actionstructure of A. Then the pair (a', e) is a* decomposition *of a for E, if $a = a' \cdot e$ and $e \in E$. We define the following preorder over decompositions for E.*

$$(a_1, e_1) \leq (a_2, e_2) \text{ if } a_1 = a_2 \cdot e \text{ and } e \cdot e_1 = e_2 \text{ for some } e \in E$$

Say the decomposition (a^*, e^*) of a is maximal *if $(a', e) \leq (a^*, e^*)$ for any other decomposition (a', e) of a for E.* ■

For some static sub-actionstructures E of A, there may exist certain actions in A which cannot be decomposed further (in the sense of the above preorder).

Definition 4.40 *Let E be a static sub-actionstructure of the action structure A. Then $a \in A$ is* pure for E *if for every $e \in E$ the decomposition (a, e) is maximal. We say that the decomposition (a, e) is* pure for E *if a is pure for E and $e \in E$.* ■

The proposition below gives sufficient conditions for E to be a postcomponent.

Proposition 4.41 *Let E be a static sub-actionstructure of the action structure A. If every a has a pure decomposition, then E is a postcomponent of A.*

Proof See [21]. ■

It remains to be seen whether the notion of effect is useful in the semantic treatment of action calculi; in any case, our results will be shown for a particular choice of effect and may easily be stated without reference to Milner's definition of such. We shall now describe a concrete action structure which will turn out to be an effect structure for certain action calculi. The intuition behind our choice stems from the illustration we gave earlier of the possible effects of exported control-bound names. We argued that such bound names might need to be distinguished from each other; our definition of *concrete effects* admits all such names that can occur at the export. A *concrete effect* is just a vector of names $\vec{v}$ together with a vector of binding names $\vec{u}$ which identify those names in $\vec{v}$ which are bound by controls.

Definition 4.42 (Concrete effects) *The* concrete effects E *for a control structure* A*, ranged over by* e*, are those actions which can be expressed in form* $(\vec{x})\langle\vec{y}\rangle$ *such that* $\{\vec{x}\} \subseteq \{\vec{y}\}$*.* ■

We will first show that E is closed under the operations of an action structure.

Lemma 4.43 *The concrete effects for* A *are a static sub-actionstructure of* A*.*

Proof It is easy to see that concrete effects are closed under tensor product, composition and abstraction. For the case of composition we show that if $e_1 = (\vec{u})\langle\vec{v}\rangle$ and $e_2 = (\vec{x})\langle\vec{y}\rangle$ with $\{\vec{u}\} \subseteq \{\vec{v}\}$ and $\{\vec{x}\} \subseteq \{\vec{y}\}$, then for $e_1 \cdot e_2 = (\vec{u})\langle\sigma\vec{y}\rangle$, $\{\vec{u}\} \subseteq \{\sigma\vec{y}\}$ where $\sigma = \{\vec{v}/\vec{x}\}$. For any $w \in \{\vec{u}\}$ we have $w \in \vec{v}$. Let the name in the corresponding position in $\vec{x}$ be z. Then $z \in \vec{y}$. By $\{w/z\}z = w$ and $\{w/z\} \in \sigma$, it follows that $w \in \sigma\vec{y}$. ■

We shall require the following fact about effects.

Lemma 4.44 *Let* A *be any control structure for which* E *is a postcomponent. Then, for any* $e \in E$*, there is some* $e^{-1} \in A$ *such that* $e \cdot e^{-1} = \mathbf{id}$*.*

Proof Consider an arbitrary effect $(\vec{x})\langle\vec{y}\rangle$. Then, substitute by a name not in $\vec{x}$ every duplicate occurrence of a name in $\vec{y}$ to get $\vec{y}'$ Hence $\vec{y}'$ consists of distinct names with exactly one occurrence of each name occurring in $\vec{x}$ (by $\vec{x} \subseteq \vec{y}$). Choose $e^{-1} = (\vec{y}')\langle\vec{x}\rangle$. ■

Remark The *retraction* of e, e^{-1} may not be in E. Consider, for instance, $\langle x\rangle$. Its retraction is $\boldsymbol{\omega}$ which is not in E.

We cannot yet show that E is an effect structure for action calculi since that would depend on the reaction rules (unless we limit ourselves to static action calculi). However, it is possible to show that E is a postcomponent of any action calculus.

Proposition 4.45 *For any action calculus* $\mathsf{AC}(\mathcal{K})$*,* E *is a postcomponent.*

Proof By lemma 4.43, E is a static sub-actionstructure of $\mathsf{AC}(\mathcal{K})$. By proposition 4.41, it suffices to identify certain molecular forms as pure actions for E and show that every action in $\mathsf{AC}(\mathcal{K})$ has a pure decomposition. First we shall show that every action a of the form

$$(\vec{x})\vec{\mu}(\vec{y})\langle\vec{z}\rangle$$

where $\vec{z}$ are distinct names and $\{\vec{z}\} \subseteq \{\vec{x}\vec{y}\}$, is pure for E. Consider any effect $e = (\vec{z})\langle\vec{v}\rangle$ (the choice of $\vec{z}$ in the effect does not result in any loss of generality, by alphaconversion) giving $a \cdot e = (\vec{x})\vec{\mu}(\vec{y})\langle\vec{v}\rangle$. We show that whenever $a \cdot e = a' \cdot e'$, then for some e'', $a' = a \cdot e''$ and $e = e'' \cdot e$. For any $e' = (\vec{w})\langle\vec{v}\rangle$ (again, choosing $\vec{v}$ does not reduce generality, by alphaconversion), we choose $e'' = (\vec{z})\langle\vec{w}\rangle$. We must now show that every action a has a pure decomposition. Consider $a = (\vec{x})\vec{\mu}(\vec{y})\langle\vec{z}\rangle$. Now, $a = (\vec{x})\vec{\mu}(\vec{y})\langle\vec{w}\rangle \cdot (\vec{w})\langle\vec{z}\rangle$ for some $\vec{w}$ such that $\{\vec{w}\} = \{\vec{z}\} \cap \{\vec{x}\vec{y}\}$. Clearly $(\vec{w})\langle\vec{z}\rangle \in E$ and $(\vec{x})\vec{\mu}(\vec{y})\langle\vec{w}\rangle$ is a pure action. ∎

Remark The reader will, by now, be unsurprised by the fact that E is a postcomponent for both $\mathsf{AC}^{\mathsf{r}}(\mathcal{K})$ and $\mathsf{AC}^{\mathsf{r}\epsilon}(\mathcal{K})$, for any $\mathcal{K}$.

Proposition 4.46 *E is a postcomponent for ν*SKEL.

Proof Similar to that of proposition 4.45 with pure actions $(\vec{x})\nu\tilde{y}\langle\vec{z}\rangle$ with $\vec{z} \subseteq \{\vec{x}\vec{y}\}$. ∎

4.2.4 Dynamics

We have already hinted at the connection between νSKEL and concrete effects. Proposition 4.46 expresses the precise correspondence between the static structure of νSKEL and the concrete effects. In this section we shall see that, under a natural choice of dynamics for νSKEL, there exists a further connection which makes restriction skeleta an interesting classifier.

Definition 4.47 (Restriction skeleta: dynamics) *The relation* $\searrow$ *on* $\boldsymbol{\nu}$SKEL *is the reflexive transitive closure of the smallest relation such that, for any skeleton* $s = (\vec{x})\boldsymbol{\nu}S\langle\vec{z}\rangle$

$$(\vec{x})\boldsymbol{\nu}S\langle\vec{z}\rangle \searrow (\vec{x})\boldsymbol{\nu}(S - \{u\})\langle\{{}^{v}\!/\!_{u}\}\vec{z}\rangle$$

where $u \in S$. ∎

Proposition 4.48 *The relation* $\searrow$ *on* $\boldsymbol{\nu}$SKEL *is preserved by the operations of an action structure together with reflexion.*

Proof Routine. ∎

The following propositions give a flavour of the dynamics for $\boldsymbol{\nu}$SKEL. We note, by Proposition 4.49, the interesting distinction between pSKEL and $\boldsymbol{\nu}$SKEL, caused by the simple relegation of an equation to a reaction rule. This effectively expresses the intuition that two distinct bound names (two independent dataflow channels) convey less information than two identical bound names (signifying a dataflow channel forked into two).

Proposition 4.49 *The relation* $\searrow$ *is the smallest reaction relation on* $\boldsymbol{\nu}$SKEL *closed under the following rules:*

$$\begin{array}{rcl} \boldsymbol{\nu} & \searrow & \langle x\rangle \\ \boldsymbol{\nu}\otimes\boldsymbol{\nu} & \searrow & \boldsymbol{\nu}\cdot(x)\langle xx\rangle \end{array}$$

Proof Let $\searrow$ be the smallest reaction relation on $\mathcal{T}(\mathcal{K}_{\boldsymbol{\nu}})/\mathsf{AC}^{\nu\mathsf{S}}$ closed under the rules. We can then show that $s \searrow s'$ if and only if $\widehat{s} \searrow \widehat{s'}$.

($\Longrightarrow$) By proposition 4.48, it suffices to show that $\boldsymbol{\nu} \searrow \langle x\rangle$ and $\boldsymbol{\nu}\otimes\boldsymbol{\nu} \searrow \boldsymbol{\nu}\cdot(x)\langle xx\rangle$ in $\boldsymbol{\nu}$SKEL. It is immediate that $\boldsymbol{\nu} \searrow \langle x\rangle$ in $\boldsymbol{\nu}$SKEL, i.e. $(\,)\boldsymbol{\nu}u\langle u\rangle \searrow (\,)\langle x\rangle$; and $\boldsymbol{\nu}\otimes\boldsymbol{\nu} \searrow \boldsymbol{\nu}\cdot(x)\langle xx\rangle$, i.e. $(\,)\boldsymbol{\nu}ux\langle ux\rangle \searrow (\,)\boldsymbol{\nu}x\langle xx\rangle$.

($\Longleftarrow$) To see that the reaction

$$\widehat{(\vec{x})\boldsymbol{\nu}S\langle\vec{z}\rangle} \searrow \widehat{(\vec{x})\boldsymbol{\nu}(S - \{u\})\langle\{{}^{v}\!/\!_{u}\}\vec{z}\rangle}$$

is derivable from the above rules, consider whether v is in S. If it is, then $\widehat{(\vec{x})\boldsymbol{\nu}S\langle\vec{z}\rangle} = (\vec{x})((\boldsymbol{\nu}\otimes\boldsymbol{\nu})\cdot(uv)(\vec{y})\langle\vec{z}\rangle)$ for some $\tilde{y}$ such that $S = \{u, v, \tilde{y}\}$. Clearly, by $\boldsymbol{\nu}\otimes\boldsymbol{\nu} \searrow \boldsymbol{\nu}\cdot(x)\langle xx\rangle$, $(\vec{x})((\boldsymbol{\nu}\otimes\boldsymbol{\nu})\cdot(uv)(\vec{y})\langle\vec{z}\rangle) \searrow (\vec{x})(\boldsymbol{\nu}\cdot(v)\langle vv\rangle\cdot(uv)(\vec{y})\langle\vec{z}\rangle)$ which is equal to $(\vec{x})\boldsymbol{\nu}(S - \widehat{\{u\}})\langle\{v/u\}\vec{z}\rangle$.

If $v \notin S$, then we have $\widehat{(\vec{x})\boldsymbol{\nu}S\langle\vec{z}\rangle} = (\vec{x})(\boldsymbol{\nu}\cdot(u)(\vec{y})\langle\vec{z}\rangle)$ for some $\tilde{y}$ such that $S = \{u, \tilde{y}\}$. By $\boldsymbol{\nu} \searrow \langle v\rangle$, $(\vec{x})(\boldsymbol{\nu}\cdot(u)(\vec{y})\langle\vec{z}\rangle) \searrow (\vec{x})(\langle v\rangle\cdot(u)(\vec{y})\langle\vec{z}\rangle)$ which is equal to $(\vec{x})\boldsymbol{\nu}(S - \widehat{\{u\}})\langle\{v/u\}\vec{z}\rangle$. ∎

The following logical characterisation of the dynamics of $\boldsymbol{\nu}$SKEL is the essence of the qualification of $\boldsymbol{\nu}$SKEL as a classifier of dynamics.

Proposition 4.50 *For any two skeleta s_1 and s_2,*

$$s_1 \searrow s_2 \iff \forall s_1', \vec{x}.\, \langle\vec{x}\rangle\cdot s_1 = s_1'\cdot e \Rightarrow \exists s_2'.\, \langle\vec{x}\rangle\cdot s_2 = s_2'\cdot e$$

Proof ($\Longrightarrow$) First we shall demonstrate that it suffices to show that for some pure s_1^p, if $s_1 \searrow s_2$ and $\langle\vec{w}\rangle\cdot s_1 = s_1^p\cdot e^p$ for some e^p and s^p, then there is some s_2'' such that $s_2 = s_2''\cdot e^p$. Assume that this is true; then if $\langle\vec{w}\rangle\cdot s_1 = s_1'\cdot e$ for any s_1' and e, we have $s_1' = s_1^p\cdot e'$ and $e^p = e'\cdot e$ for some e'. In this case, choosing $s_2' = s_2''\cdot e'$ would give the result since $\langle\vec{w}\rangle\cdot s_2 = s_1''\cdot e^p = s_1''\cdot e'\cdot e$.

Consider $s_1 = (\vec{x})\boldsymbol{\nu}\tilde{y}\langle\vec{z}\rangle$. Then $s_2 = (\vec{x})\boldsymbol{\nu}\tilde{y}'\langle\{\vec{v}/\vec{u}\}\vec{z}\rangle$ where $\vec{y} = \vec{u}\vec{y'}$. Now $\langle\vec{w}\rangle\cdot s_1 = (\boldsymbol{\nu}\tilde{y})\langle\{\vec{w}/\vec{x}\}\vec{z}\rangle$ can be written as the composite of a pure action and an effect $(\boldsymbol{\nu}\tilde{y})\langle\vec{u}\vec{y'}\rangle\cdot(\vec{u}\vec{y'})\langle\{\vec{w}/\vec{x}\}\vec{z}\rangle$. Also, $\langle\vec{w}\rangle\cdot s_2 = (\boldsymbol{\nu}\tilde{y}')\langle\{\vec{w}/\vec{x}\}\{\vec{v}/\vec{u}\}\vec{z}\rangle = (\boldsymbol{\nu}\tilde{y}')\langle\{\vec{v'}/\vec{u}\}\{\vec{w}/\vec{x}\}\vec{z}\rangle$ where $\vec{v'} = \{\vec{w}/\vec{x}\}\vec{v}$. Hence $\langle\vec{w}\rangle\cdot s_2 = (\boldsymbol{\nu}\tilde{y}')\langle\vec{v'}\vec{y'}\rangle\cdot(\vec{u}\vec{y'})\langle\{\vec{w}/\vec{x}\}\vec{z}\rangle$ and result follows.

($\Longleftarrow$) First we note that, for any $\vec{u}, \vec{v}, \vec{y}$ such that $|\vec{v}| = |\vec{y}|$, we have $(\boldsymbol{\nu}\tilde{y})\langle\vec{y}\rangle \searrow (\boldsymbol{\nu}\tilde{u})\langle\vec{v}\rangle$. Now consider s_1 and s_2 of equal arity (if not, our assumption would not hold by an argument based on well-formedness); by alphaconversion we can write $s_1 = (\vec{x})\boldsymbol{\nu}\tilde{y}\langle\vec{z}\rangle$ and $s_2 = (\vec{x})\boldsymbol{\nu}\tilde{u}\langle\vec{w}\rangle$. Then $\langle\vec{x}\rangle\cdot s_1 = (\boldsymbol{\nu}\tilde{y})\langle\vec{z}\rangle = (\boldsymbol{\nu}\tilde{y})\langle\vec{y}\rangle\cdot(\vec{y})\langle\vec{z}\rangle$. By assumption, $\langle\vec{x}\rangle\cdot s_2 = (\boldsymbol{\nu}\tilde{u})\langle\vec{v}\rangle\cdot(\vec{y})\langle\vec{z}\rangle$, for some $\vec{v}$. But by $(\boldsymbol{\nu}\tilde{y})\langle\vec{y}\rangle \searrow (\boldsymbol{\nu}\tilde{u})\langle\vec{v}\rangle$ and the fact that reaction is preserved by composition, $\langle\vec{x}\rangle\cdot s_1 \searrow \langle\vec{x}\rangle\cdot s_2$. Hence $(\vec{x})(\langle\vec{x}\rangle\cdot s_1) \searrow (\vec{x})(\langle\vec{x}\rangle\cdot s_2)$ and since $\vec{x}$ are not free in either s_1 or s_2 we get $s_1 \searrow s_2$. ∎

Lemma 4.51 *For any s_1, s'_1, s_2 and e, whenever $s_1 \searrow s_2$ and $s_1 = s'_1 \cdot e$ then, for some s'_2, $s_2 = s'_2 \cdot e$ and $s'_1 \searrow s'_2$.*

Proof Assume $s_1 \searrow s_2$ and $s_1 = s'_1 \cdot e$. Choose $\vec{x}$ not in the surfaces of s_2 and e. Then $\langle\vec{x}\rangle \cdot s_1 = \langle\vec{x}\rangle \cdot s'_1 \cdot e$. By proposition 4.50, for some s''_2, $\langle\vec{x}\rangle \cdot s_2 = s''_2 \cdot e$. Then $s_2 = (\vec{x})\langle\vec{x}\rangle \cdot s_2 = (\vec{x})(\langle\vec{x}\rangle \cdot s_2) = (\vec{x})(s''_2 \cdot e) = (\vec{x})s''_2 \cdot e$. Choosing $s'_2 = (\vec{x})s''_2$ gives $s_2 = s'_2 \cdot e$. To show that $s'_1 \searrow s'_2$ we apply the context $[\_] \cdot e^{-1}$ to each side of $s_1 \searrow s_2$. ■

First we shall establish an important connection between concrete effects and νSKEL.

Proposition 4.52 *E is an effect structure for νSKEL.*

Proof By proposition 4.46, E is a postcomponent of νSKEL. We now show that all the actions in E are inert in νSKEL. Consider $s \cdot e \searrow s'$, we must show that, for some s'', $s \searrow s''$ and $s' = s'' \cdot e$. By lemma 4.51 result follows immediately. ■

Indeed we can prove something stronger. The following lemma will prepare the ground for our main theorem which justifies the choice of νSKEL as a classifier of dynamics.

Lemma 4.53 *Let A be a control structure for which E is a postcomponent. If there is a morphism $\Phi : A \to \nu\mathsf{SKEL}$ in $\mathbf{CS}^s(\mathcal{K})$ such that*

$$\Phi(a) = s \cdot e \;\Rightarrow\; \exists a' \in A.\ a = a' \cdot e$$

then E is an effect structure for A.

Proof Assume $a \cdot e \searrow b$. Then, $\Phi(a \cdot e) \searrow \Phi(b)$. Now, $\Phi(a \cdot e) = \Phi(a) \cdot e$. By lemma 4.51, for some s, $\Phi(b) = s \cdot e$. Then, by assumption, there is some a' such that $b = a' \cdot e$. Hence $a \cdot e \searrow a' \cdot e$, and applying the context $[\_] \cdot e^{-1}$ to each side gives the result. ■

Theorem 4.54 *For any signature $\mathcal{K}$ and reaction rules $\mathcal{R}$, the action calculus* $\mathsf{AC}(\mathcal{K},\mathcal{R})$ *has a morphism to* $\nu\mathsf{SKEL}$ *in* $\mathbf{CS}(\mathcal{K},\mathcal{R})$ *if and only if the concrete effects E give an effect structure for* $\mathsf{AC}(\mathcal{K},\mathcal{R})$.

Proof ($\Longrightarrow$) By proposition 4.45, E is a postcomponent of $\mathsf{AC}(\mathcal{K})$ and hence of $\mathsf{AC}(\mathcal{K},\mathcal{R})$. Then, by lemma 4.53, it suffices to show that the morphism $\boldsymbol{\nu}\mathbf{skel}$: $\mathsf{AC}(\mathcal{K}) \to \nu\mathsf{SKEL}$ has the property

$$\forall a \in \mathsf{AC}(\mathcal{K}).\ \boldsymbol{\nu}\mathbf{skel}(a) = s \cdot e\ \Rightarrow\ \exists a' \in \mathsf{AC}(\mathcal{K}).\ a = a' \cdot e$$

To show this we note that the mapping $\boldsymbol{\nu}\mathbf{skel}$ takes each pure action in $\mathsf{AC}(\mathcal{K})$ to a pure action in $\nu\mathsf{SKEL}$. Hence, consider an arbirary $a \in \mathsf{AC}(\mathcal{K})$. Then $a = a_p \cdot e_p$ for some pure action a_p; and therefore $\boldsymbol{\nu}\mathbf{skel}(a) = \boldsymbol{\nu}\mathbf{skel}(a_p) \cdot e_p$. But $\boldsymbol{\nu}\mathbf{skel}(a_p)$ is pure in $\nu\mathsf{SKEL}$ and hence, if $a = s \cdot e$, then by the definition of purity, for some e', $s = \boldsymbol{\nu}\mathbf{skel}(a_p) \cdot e'$ and $e_p = e' \cdot e$. Choosing $a' = a_p \cdot e'$ gives the required result.

($\Longleftarrow$) There is a (unique) morphism $\boldsymbol{\nu}\mathbf{skel}$ in $\mathbf{CS}^s(\mathcal{K})$ from $\mathsf{AC}(\mathcal{K})$ to $\nu\mathsf{SKEL}$. It therefore suffices to show that $\boldsymbol{\nu}\mathbf{skel}$ preserves the reaction relation. Assume $a \searrow b$. By proposition 4.50, it suffices to show that, for any $\vec{x}$, s and e, if $\langle\vec{x}\rangle \cdot \boldsymbol{\nu}\mathbf{skel}(a) = s \cdot e$, then $\boldsymbol{\nu}\mathbf{skel}(b) = s' \cdot e$, for some s'. Now $a \searrow b$ implies $\langle\vec{x}\rangle \cdot a \searrow \langle\vec{x}\rangle \cdot b$. But, $\boldsymbol{\nu}\mathbf{skel}(\langle\vec{x}\rangle \cdot a) = \langle\vec{x}\rangle \cdot \boldsymbol{\nu}\mathbf{skel}(a) = s \cdot e$ and therefore, for some a', $\langle\vec{x}\rangle \cdot a = a' \cdot e$. Hence $a' \cdot e \searrow \langle\vec{x}\rangle \cdot b$. But, since e is an effect in $\mathsf{AC}(\mathcal{K})$, it is inert and therefore $\langle\vec{x}\rangle \cdot b = b' \cdot e$ for some b'. Since $\boldsymbol{\nu}\mathbf{skel}$ preserves the operations of a control structure, $\langle\vec{x}\rangle \cdot \boldsymbol{\nu}\mathbf{skel}(b) = \boldsymbol{\nu}\mathbf{skel}(\langle\vec{x}\rangle \cdot b) = \boldsymbol{\nu}\mathbf{skel}(b') \cdot e$. Choosing $s' = \boldsymbol{\nu}\mathbf{skel}(b')$ gives the result. ∎

Let us review what has been achieved. We started with an examination of the information that flows through dataflow channels in the setting of action calculi. Our analysis led us to distinguish the *concrete effects*, a class of actions which are inactive but which may instigate reaction upon being fed to certain actions. We then considered *effect structures* which give an abstract account of what actions can send through dataflow channels. For any action structure A, an effect structure E for A must be a postcomponent of A (a property of the statics) and must consist of inert actions (a property of the dynamics). We then showed that the concrete effects satisfy the postcomponent property for arbitrary action calculi (and their variants). The above theorem states that the inertia property holds for an action calculus *just when* there exists a morphism (in $\mathsf{CS}(\mathcal{K})$) from it to νSKEL, hence the claim that νSKEL acts as a classifier.

Discussion Analogous results to theorem 4.54 can be obtained for the reflexive variants of action calculi with very similar proofs. This suggests that there is some common structure which, when elicited, can be employed to prove our results more abstractly. There are some similarities which are simple to state and which may have bearing on the uniformity with which similar results could be obtained for the variants. For instance, in all three variants, there exists an injection from the (set of) concrete effects to the hom-set consisting of all the actions. Also, in each case, every action has a pure decomposition for the concrete effects.

One also asks whether variants of skeleta arise from other concrete forms of effect (or vice versa). A variation that springs to mind is that which result from removing the constraint (in the definition of restriction skeleta) that the names in the set S bind at least some name in the export vector $\vec{z}$. Does the variation of skeleta given by removing the constraint allow us to obtain analogous results? For such a case, it is natural to take as operations on the skeletal forms those defined exactly for the reflexive molecular forms over the empty signature. This means that strictness is lost, and therefore our scope will exclude strict reflexive action calculi. In this setting, a concrete kind of effect that suggests itself is that given by entities—call them pre-effects—of the form $(\vec{x})\langle\vec{y}\rangle$ with $\vec{x}$ and $\vec{y}$ unconstrained beyond the requirement that $\vec{x}$ consist of distinct names. These pre-effects form a postcomponent of both action calculi and reflexive ones; it is easy to see why by considering the pure actions (for the pre-effects) of the form

$$(\vec{x})[\vec{\mu}(\vec{y})]\langle\vec{x}\vec{y}\rangle$$

Included among the pre-effects, is the discard operation $\boldsymbol{\omega}$ since it is equal to $(x)\langle\ \rangle$. This immediately implies that we lose retractablility – the guaranteed existence of a right inverse – in the action calculi and its reflexive variants. Retractability effectively says that any entity of information (effect) can be discarded and is therefore an intuitively desirable property. The loss of this property also renders our method of proof of the inertia of effects (see proposition 4.51) inapplicable.

Chapter 5

The Reflexive π-calculus

Earlier it was claimed that several existing concrete models of concurrency fit readily in the framework we have developed. One leading example of such models is Milner's π-calculus which allows the expression of independent processes that are able to pass links to each other, hence its claim as a calculus of mobile processes. Several operational models for this calculus have been developed, largely along the lines familiar in mainstream process algebra of which the π-calculus is an instance, if a rather powerful one. Therefore, by presenting an operational semantics of a reflexive action calculus inspired by the π-calculus, we hope to throw some light on the connections between mainstream process algebra and our framework.

In this chapter and the next we shall examine the reflexive π-calculus $\mathsf{PIC}^{\mathsf{r}}$, a reflexive action calculus determined by controls whose behaviour is similar to that of the essential constructs of the original π-calculus. In particular, it is possible to express *mobility*—the ability of processes to exchange (the names of) communication ports—in both calculi. The choice of dealing with the π-calculus cast in the reflexive framework rather than the (non-reflexive) one was deliberate since, as we shall see, the presence of reflexion plays a crucial role in the operational semantics that we shall develop.

Besides the presence of reflexion, there are other important differences between $\mathsf{PIC}^{\mathsf{r}}$ and the original π-calculus. First, the only prefix operator is input prefix in $\mathsf{PIC}^{\mathsf{r}}$, the output being asynchronous as in the ν-calculus of Honda and Yoshida [11]. There are also important enhancements not found even in the full π-calculus:

processes, which in action calculi are represented as (complex) actions, may import as well as export names through the basic operations of datum, abstraction and composition. In short PIC^r is an asynchronous π-calculus with explicit dataflow operators.

It is worth remarking that we have chosen to present the operational semantics for PIC^r rather than PIC, the non-reflexive action calculus PIC determined by the same controls together with restriction (which in the reflexive framework is a derived operation). The reason for this is that the presence of reflexion is crucial for our approach. One of the problems with giving an operational semantics for PIC, is that in analysing actions for redexes, it does not suffice to determine the presence of a complementary pair of controls (e.g. $\langle x\rangle\mathbf{box}a(\vec{y})$ and $\langle x\vec{v}\rangle\mathbf{out}$); care must also be taken to ensure that no links exist between them. In other words, the names $\vec{v}$ must be distinct from the names $\vec{y}$. This requirement arises since the reaction rule $\mathbf{out}_x \otimes \mathbf{box}_x a \searrow a$ requires that the complementary molecules have no common links. This is not the case in the reflexive framework since, by lemma 3.10, every composition $a \cdot b$ can be expressed in terms of the tensor product of a and b (together with permutators and reflexion). Since the occurrence of reaction is to be concluded entirely upon consideration of the labels (rather than the actions or terms which perform the labelled transition), in PIC (but not in PIC^r) this would require labels to include information about the binding structure related to the molecules. This significantly complicates the treatment and for this reason PIC^r was preferred.

Outline In Section 5.1 we present PIC^r and explore its dynamics through examples. The examples will lead to an analysis of reaction and redex formation and their interaction with the operations of the calculus. This analysis will serve as a basis for the formulation of labelled transitions in the following section. In Section 5.2 we introduce labels, which are descriptions of the contribution actions can make towards the formation of redexes; followed by labelled transitions between terms—represented as sequents—and the rules for deriving labelled transition sequents. In Section 5.3 labelled transition relations are defined in terms of derivable sequents. Several important properties of derivable sequents, and thereby, of labelled transition relations, are obtained. The main result in this chapter is that terms of $\mathsf{PIC}^{\mathsf{r}}$ which are provably equal in AC^{r} perform identical transitions to residual terms which are also equal, hence ensuring a well-defined notion of labelled transition on the actions (rather than just the terms) of $\mathsf{PIC}^{\mathsf{r}}$. We also give a characterisation of labelled transitions in the setting of the molecular forms and show that each τ-transition corresponds to a computational step.

5.1 Controls and Reaction

The reflexive π-calculus $\mathsf{PIC}^{\mathsf{r}}$ is determined by the controls that together with the operations of a reflexive control structure give the reflexive action calculus. Informally, parallel composition corresponds to $\otimes$, asynchronous output $\overline{x}\langle v\rangle$ to $\langle v\rangle\cdot\mathbf{out}_x$, and input prefix $x(y).P$ to $\mathbf{box}_x a$, where a corresponds to the abstraction of y from P: $(y)P$ by an abuse of notation.

Definition 5.1 ($\mathsf{PIC}^{\mathsf{r}}$) *The reflexive π-calculus $\mathsf{PIC}^{\mathsf{r}}$ is the reflexive action calculus over the controls $\{\mathbf{out}, \mathbf{box}\}$ together with the following arity rules*

$$\frac{}{\mathbf{out} : p \otimes m \rightarrow \epsilon} \qquad \frac{a : m \rightarrow n}{\mathbf{box}a : p \rightarrow n}$$

and the reaction rule $\mathbf{out}_x \otimes \mathbf{box}_x a \searrow a$ where

$$\begin{aligned}\mathbf{out}_x &\stackrel{\text{def}}{=} ((\langle x\rangle \otimes \mathbf{id}) \cdot \mathbf{out} \\ \mathbf{box}_x a &\stackrel{\text{def}}{=} \langle x\rangle \cdot \mathbf{box}a\end{aligned}$$

■

With reference to the constructs $\mathbf{out}_x$ and $\mathbf{box}_x a$ the name x is sometimes referred to as the *subject name* of the relevant molecule.

Example As an example of reaction in $\mathsf{PIC}^{\mathsf{r}}$, consider the action $(\langle xv\rangle \cdot \mathbf{out}) \otimes \mathbf{box}_x(y)a$. In the theory AC^{r} the following equality is provable:

$$\begin{aligned}(\langle xv\rangle \cdot \mathbf{out}) \otimes \mathbf{box}_x(y)a &= (\langle v\rangle \cdot (\langle x\rangle \otimes \mathbf{id}) \cdot \mathbf{out}) \otimes \mathbf{box}_x(y)a \\ &= \langle v\rangle \cdot (((\langle x\rangle \otimes \mathbf{id}) \cdot \mathbf{out}) \otimes \mathbf{box}_x(y)a) \\ &= \langle v\rangle \cdot (\mathbf{out}_x \otimes \mathbf{box}_x(y)a)\end{aligned}$$

The reaction $\langle v\rangle \cdot (\mathbf{out}_x \otimes \mathbf{box}_x(y)a) \searrow \langle v\rangle \cdot (y)a$ is derivable by the reaction rule $\mathbf{out}_x \otimes \mathbf{box}_x a \searrow a$ together with the condition that reaction is preserved by composition. For any action a, $\langle v\rangle \cdot (y)a = \{v/y\}a$ is immediately provable in AC^{r}. We then note the correspondence with the following transition in the original π-calculus:

$$\overline{x}\langle v\rangle \mid x(y).P \xrightarrow{\tau} \{v/y\}P$$

In the above transition we note that the τ label stands for a *single* interaction, whereas the reaction relation $\searrow$ represents arbitrarily many (including zero) interactions or computational steps. For the treatment of the dynamics of $\mathsf{PIC}^{\mathsf{r}}$, we shall find it useful to define the single-step reaction relation $\overset{1}{\searrow}$. We can then show that the reaction relation is identical to the reflexive transitive closure of the single step reaction relation $\overset{1}{\searrow}$, which is given as the smallest relation satisfying the rules shown in figure 5–1. Then, as in the example above, the single step reaction $\langle v\rangle \cdot (\mathbf{out}_x \otimes \mathbf{box}_x(y)a) \overset{1}{\searrow} \{v/y\}a$ is derivable by applying the rules SYNC, R· and STRUCT in that order. Note also that STRUCT rule ensures that the relation is well defined for the equivalence classes (on terms) induced by $\equiv_{\mathsf{AC}^{\mathsf{r}}}$.

Proposition 5.2 *The reaction relation $\searrow$ is equal to the reflexive transitive closure of the single step reaction relation $(\overset{1}{\searrow})^*$.*

Proof The reaction relation $\searrow$ is the smallest preorder which contains the reaction $\mathbf{out}_x \otimes \mathbf{box}_x t \searrow t$ and is preserved by the action structure operations together with reflexion. Adding reflexivity and transitivity to the rules defining $\overset{1}{\searrow}$ (as the smallest relation satisfying the rules) gives identical rules as those for $\searrow$. ∎

$$\text{L}\cdot\ \frac{a \searrow^1 a'}{a \cdot b \searrow^1 a' \cdot b} \qquad \text{L}\otimes\ \frac{a \searrow^1 a'}{a \otimes b \searrow^1 a' \otimes b} \qquad \mathbf{ab}\ \frac{a \searrow^1 a'}{\mathbf{ab}_x a \searrow^1 \mathbf{ab}_x a'}$$

$$\text{R}\cdot\ \frac{b \searrow^1 b'}{a \cdot b \searrow^1 a \cdot b'} \qquad \text{R}\otimes\ \frac{b \searrow^1 b'}{a \otimes b \searrow^1 a \otimes b'} \qquad \uparrow\ \frac{a \searrow^1 a'}{\uparrow a \searrow^1 \uparrow a'}$$

$$\text{SYNC}\ \frac{}{\mathbf{out}_x \otimes \mathbf{box}_x a \searrow^1 a} \qquad \text{STRUCT}\ \frac{a = b \quad b \searrow^1 b' \quad b' = a'}{a \searrow^1 a'}$$

Figure 5–1: One-Step Reaction Relation

It is informative to consider the mechanics of reaction on the molecular forms, especially for single step reaction. As we shall see, a redex corresponds to two complementary molecules placed side by side. We recall that $[\![-]\!]$ is the unique homomorphism from the term algebra to the molecular forms, and $\widehat{(-)}$ is its inverse. We shall denote molecules $\langle\vec{x}\rangle K\vec{a}(\vec{y})$ by $\mu(\vec{y})$, and $\mu_1(\vec{y}_1), \ldots, \mu_r(\vec{y}_r)$ by $\vec{\mu}(\vec{y})$ with $\vec{\mu} = \mu_1 \cdots \mu_r$ and $\vec{y} = \vec{y}_1 \cdots \vec{y}_r$. Then,

$$\begin{aligned} \widehat{\mu} &\stackrel{\text{def}}{=} \langle\vec{x}\rangle \cdot K\vec{\widehat{a}} \\ \widehat{\vec{\mu}} &\stackrel{\text{def}}{=} \widehat{\mu_1} \otimes \cdots \otimes \widehat{\mu_r} \end{aligned}$$

Proposition 5.3 *For any t, t', $t \searrow^1 t'$ if and only if*

$$[\![t]\!] = (\vec{u})[\langle x\vec{w}\rangle\mathbf{out}, \langle x\rangle\mathbf{box}a(\vec{u}_1), \vec{\mu}(\vec{u}_2)]\langle\vec{v}\rangle \text{ and } [\![t']\!] = {}^*\{{}^{\vec{z}_a\vec{w}}/_{\vec{u}_1\vec{x}_a}\}(\vec{u})[\vec{\lambda}(\vec{y}_a), \vec{\mu}(\vec{u}_2)]\langle\vec{v}\rangle$$

where $a = (\vec{x}_a)\vec{\lambda}(\vec{y}_a)\langle\vec{z}_a\rangle$.

Proof ($\Longrightarrow$) Induction on the depth of derivation of $t \searrow^1 t'$.

($\Longleftarrow$) Let the (unique) inverse map of $[\![-]\!]$ be $\widehat{(-)}$. Then, by the STRUCT rule it suffices to give a derivation of $\widehat{[\![t]\!]} \searrow^1 \widehat{[\![t']\!]}$. By alphaconversion we can assume w.l.o.g. that the names $\vec{x}_a\vec{y}_a$ do not occur except within a.

$$
\begin{array}{lll}
\widehat{[\![t]\!]} & = & \uparrow_{m_1\otimes m_2}(\vec{u}_1\vec{u}_2\vec{u})(((\langle\vec{w}\rangle\cdot\mathbf{out}_x)\otimes\mathbf{box}_x(\vec{x}_a)\vec{\lambda}\widehat{(\vec{y}_a)\langle\vec{z}_a}\rangle\otimes\widehat{\vec{\mu}}\otimes\langle\vec{v}\rangle) \\
 & \searrow^1 & \uparrow_{m_1\otimes m_2}(\vec{u}_1\vec{u}_2\vec{u})(((\langle\vec{w}\rangle\cdot(\vec{x}_a)\vec{\lambda}\widehat{(\vec{y}_a)\langle\vec{z}_a\rangle})\otimes\widehat{\vec{\mu}}\otimes\langle\vec{v}\rangle) \\
 & = & \uparrow_{m_1\otimes m_2}(\vec{u}_1\vec{u}_2\vec{u})([\vec{w}/\vec{x}_a]\uparrow_n(\vec{y}_a)(\widehat{\vec{\lambda}}\otimes\langle\vec{z}_a\rangle)\otimes\widehat{\vec{\mu}}\otimes\langle\vec{v}\rangle) \quad\quad \vec{y}_a : n \\
 & = & \uparrow_{m_1\otimes m_2}(\vec{u}_1\vec{u}_2\vec{u})[\vec{w}/\vec{x}_a](\uparrow_n(\vec{y}_a)(\widehat{\vec{\lambda}}\otimes\langle\vec{z}_a\rangle)\otimes\widehat{\vec{\mu}}\otimes\langle\vec{v}\rangle) \\
 & = & \uparrow_{m_1\otimes m_2}(\vec{u}_1\vec{u}_2\vec{u})\uparrow_m(\vec{x}_a)((\langle\vec{w}\rangle\otimes\uparrow_n(\vec{y}_a)(\widehat{\vec{\lambda}}\otimes\langle\vec{z}_a\rangle)\otimes\widehat{\vec{\mu}}\otimes\langle\vec{v}\rangle) \quad\quad 3.29 \\
 & = & \uparrow_{m\otimes m_1\otimes m_2}(\vec{x}_a\vec{u}_1\vec{u}_2\vec{u})((\langle\vec{w}\rangle\otimes\uparrow_n(\vec{y}_a)(\widehat{\vec{\lambda}}\otimes\langle\vec{z}_a\rangle)\otimes\widehat{\vec{\mu}}\otimes\langle\vec{v}\rangle \quad\quad 3.26^* \\
 & = & \uparrow_{m_1\otimes m\otimes m_2}(\vec{u}_1\vec{x}_a\vec{u}_2\vec{u})(\uparrow_n(\vec{y}_a)(\widehat{\vec{\lambda}}\otimes\langle\vec{z}_a\rangle)\otimes\langle\vec{w}\rangle\otimes\widehat{\vec{\mu}}\otimes\langle\vec{v}\rangle) \\
 & = & \uparrow_{m_1\otimes m\otimes m_2}(\vec{u}_1\vec{x}_a\vec{u}_2\vec{u})\uparrow_n(\vec{y}_a)(\widehat{\vec{\lambda}}\otimes\langle\vec{z}_a\rangle\otimes\langle\vec{w}\rangle\otimes\widehat{\vec{\mu}}\otimes\langle\vec{v}\rangle) \\
 & = & \uparrow_{m_1\otimes m}(\vec{u}_1\vec{x}_a)\uparrow_{m_2\otimes n}(\vec{u}_2\vec{y}_a\vec{u})(\widehat{\vec{\lambda}}\otimes\widehat{\vec{\mu}}\otimes\langle\vec{z}_a\rangle\otimes\langle\vec{w}\rangle\otimes\langle\vec{v}\rangle) \quad\quad 3.26^* \\
 & = & \uparrow_{m_1\otimes m}(\vec{u}_1\vec{x}_a)(\vec{u})[\vec{\lambda}(\vec{y}_a),\widehat{\vec{\mu}(\vec{u}_2)}]\langle\vec{z}_a\vec{w}\vec{v}\rangle \\
 & = & {}^*\{\vec{z}_a\vec{w}/\vec{u}_1\vec{x}_a\}\widehat{(\vec{u})[\vec{\lambda}}(\vec{y}_a),\vec{\mu}(\vec{u}_2)]\langle\vec{v}\rangle \\
 & = & \widehat{[\![t']\!]}
\end{array}
$$

■

5.1.1 Reaction and the operations of PIC^r

Some actions are *inactive*, or unable to go to any action save themselves under reaction. However, certain combinations of inactive actions may themselves be active. Consider the actions $\mathbf{out}_x$ and $\mathbf{box}_x a$: no reaction can be derived from either of them in isolation. However, when combined together by means of $\otimes$, the combination may react to a. Any semantics based on the dynamics must take such interaction into account: the labelled transitions upon which our semantics is based do just that.

Before presenting the operational semantics of PIC^r, we shall first explore some of the interactions between reaction and the operations of the calculus. In particular, we want to identify the components in an action which can contribute to the creation of a redex. We will also examine the way in which the operations can bring such contributions together, possibly resulting in the formation of a complete redex as a result. Later, we shall formalise this by the notion of a labelled transition, with labels representing such contributions. It is insightful to consider these interactions in the setting of molecular forms since the intuitions behind the labelled transitions are most easily explained with reference to them.

1. A single computational step, or reaction, occurs just when a molecule $\langle x\vec{v}\rangle\mathbf{out}(\,)$ encounters a molecule $\langle x\rangle\mathbf{box}a'(\vec{y})$ inside the body of an action a. If a contains $\langle x\vec{v}\rangle\mathbf{out}(\,)$ but no $\langle x\rangle\mathbf{box}a'(\vec{y})$, a reaction may be induced by "placing" the re-

quired complementary molecule in the body of a. In terms of the operations of the calculus, there may be various ways of introducing this complementary molecule. For instance, if the name x is free inside the molecular form of a, one way of placing such a complementary molecule is through a tensor product of the action with $\mathbf{box}_x a'$; another way is to compose a with an action b containing the complementary molecule (with x free in the molecular form of b). Hence, we may regard a as being able to contribute a partial redex $\langle x\vec{v}\rangle\mathbf{out}(\ )$.

2. Consider now, the action $a = (y)(\mathbf{out}_x \otimes \mathbf{box}_y a')$ whose molecular form is

$$(y\vec{v})\,[\langle x\vec{v}\rangle\mathbf{out}(\ ), \langle y\rangle\mathbf{box}a'(\vec{w})]\,\langle\vec{w}\rangle$$

Clearly, a is inactive and the placement of the molecule $\langle x\rangle\mathbf{box}a''(\vec{y})$ in its body can create reaction. However, there are other ways by which reaction can be induced: precomposing $\langle x\rangle \otimes \mathbf{id}$ will cause any free occurrence of y in the body of the action to be replaced by x, thereby creating the redex $\mathbf{out}_x \otimes \mathbf{box}_x a$. Indeed, precomposing by any action which exports the (free) name x at the appropriate position will cause this redex to be formed. Letting $b = (\vec{u})\,\vec{\mu}\,\langle x\vec{z}\rangle$ (with x free in b) gives $b \cdot a = (\vec{u})\,[\vec{\mu}, \langle x\vec{z}\rangle\mathbf{out}(\ ), \langle x\rangle\mathbf{box}a''(\vec{w})]\,\langle\vec{w}\rangle$ where $a'' = \{{}^{x\vec{z}}/_{y\vec{v}}\}a'$. In this case, the essential part of b which determines whether a reaction is created (through name substitution) is its export vector of names $x\vec{z}$. The point that this example makes is that b, while not necessarily contributing any molecules to create a redex in $b \cdot a$, still contributes a component (the free name x) which caused a redex to be formed in the composite action. Consequently, we must take into account not only of the molecules that an action can contribute but also of the free names available at its export.

We emphasise that in this example, the occurrence of x in the export of b has to be free, for otherwise (by the definition of composition on the molecular forms) it would have had to be alphaconverted to some name other than x to avoid clashing with the free occurrence of x in a. This example might suggest that ignoring the bound names in the export vector is justified, but, as the following example illustrates, this is not generally the case.

3. Consider the action $a = (xy)(\mathbf{out}_x \otimes \mathbf{box}_y a')$, where both x and y are bound at its input. Clearly, precomposing by any action which exports two identical names $\langle vv\rangle$ (for any v) will induce reaction. It is important to note that the occurrences

of v need not be free in b, since any pair of identical names at the export of b will create the redex: consequently, the forced alphaconversion of v in b to, say, w cannot prevent the formation of the redex $\mathbf{out}_w \otimes \mathbf{box}_w a'$. We note that in the case that v is bound in b, b can still be factorised into composites b' and $(v)\langle vv\rangle$, for some b', whereas if the occurrence is free, factorisation into some b' and $\langle vv\rangle$ is also possible. In both of these cases b may induce reaction when it is precomposed to a suitable action. This example shows that not all exported names which are bound should be ignored as possible contributions (to a redex).

4. We will now give an example which illustrates the complexity over the original π-calculus resulting from the presence of name export (non-empty output arity). The action $(\,)\,[\nu(x), \langle xv\rangle\mathbf{out}(\,)]\,\langle\,\rangle$ cannot interact with any other action. We would expect such an action to contribute as much to reaction as, for instance, $(\,)\,[\nu(x)]\,\langle\,\rangle$[1]. However, consider the slight perturbation in their molecular forms by introducing the name x at the export to give $(\,)\,[\nu(x), \langle xv\rangle\mathbf{out}(\,)]\,\langle x\rangle$ and $(\,)\,[\nu(x)]\,\langle x\rangle$. For the former action, postcomposing $b = (y)\,[\langle y\rangle\mathbf{box}a(\vec{z})]\,\langle\vec{z}\rangle$ will create a reaction whereas postcomposing with the latter action will not. Hence, even restricted ports can be made visible provided the restricted name is exported. It is clear that in our treatment we must make a distinction between ports whose names are free and visible and those of the kind just described.

5. Last of all, we present an example of how the application of the reflexion operation can create a redex within an action which previously had none. Consider the action a with molecular form

$$(x)[\nu(y), \langle x\vec{v}\rangle\mathbf{out}, \langle y\rangle\mathbf{box}a'(\vec{w})]\langle y\vec{w}\rangle$$

where $x, y : p$. Applying reflexion on a, gives the molecular form:

$$[\nu(y), \langle y\{y/x\}\vec{v}\rangle\mathbf{out}, \langle y\rangle\mathbf{box}\{y/x\}a'(\vec{w})]\langle\vec{w}\rangle$$

[1] These actions are analogous to $(\nu x)\overline{x}\langle v\rangle$ and $(\nu x)\mathbf{0}$ respectively in the π-calculus and indeed, as in there, we would expect these two actions to be identified in any reasonable model for the reflexive π-calculus.

which clearly has a redex. Thus, reflexion, while providing no contribution in itself, enabled the contributions of a to recombine in such a way as to create a complete redex. Indeed, reflexion is *necessary* to create this redex since it is the only operation which can cause the identification of the exported (restricted) name y with the imported name x.

Based upon the notion of "contribution to reaction" illustrated above we would like to formulate an operational semantics of PIC^r. As will be evident in the following sections we will choose to formalise this notion of contribution in the setting of the term algebra rather than directly on the molecular forms. The advantage of working with terms is related to the requirement of showing how the mentioned contributions are affected by arbitrary contexts built from the operations of PIC^r. While the notion of context in the case of terms is straightforward, the same cannot be said in the setting of the molecular forms. The main technical results of this chapter show that the formulation based on the terms corresponds to the intuition supplied with reference to the molecular forms. In particular, a structural lemma (lemma 5.11) ensures that labelled transition relations on molecular forms can be obtained by quotienting the labelled transition relations defined on the corresponding terms.

5.2 Labelled Transition Sequents

In the previous section we presented several examples which motivate the organisation of labelled transitions to reflect the kinds of interaction described. The essential idea behind labelled transitions is that labels should contain enough (ideally, just enough) information about the action to determine whether the reaction will be made possible when the action is placed in certain contexts. The residual of the transition allows the action resulting from such reactions to be constructed. We would like to account for any contribution to a redex no matter how small; for otherwise we cannot expect bisimulation equivalence to be a congruence.

This section is organised in three parts: the first describes the labels which formalise the notion of an action's contribution to a potential redex; the second describes labelled transitions through syntactic constructs which we shall call sequents; while the third describes a set of rules which allow such sequents to be derived.

5.2.1 Labels

As indicated by the examples presented in the previous section, the contribution an action can make towards a redex may consist of exported names and molecules. We have also shown that care must be taken to distinguish between free and bound names occurring both in the molecules (in fact, the subject names suffice) and in the export vectors of actions.

Exported names We shall start with an account of the possible substitutions an action can cause in a postcomposed action. In terms of the molecular forms, these substitutions are determined by the export vector of the precomposed action and the import vector of the postcomposed one. It is also necessary as we have seen to include some description of the freeness or otherwise of the names occurring in the export vector of the precomposed action. Consider the following molecular form:

$$a = (\vec{x})[\langle\ \rangle K(\vec{y})]\langle\vec{z}\rangle$$

The names $\vec{z}$ in the export of a may be bound by any name in $\vec{x}$ and $\vec{y}$. The possible name contributions of a to postcomposed actions could be represented as $(\vec{x}\vec{y})\langle\vec{z}\rangle$. However, we would like to distinguish between bindings originating from the imports of a and those originating from restrictions or controls since precomposition of a by some action can cause names bound by $\vec{x}$ to be instantiated whereas those bound by $\vec{y}$ cannot change (up to alphaconversion) as a result of any (static) operation of the calculus. As an illustration of this point consider the actions

$$b = (x_1x_2)[\langle\ \rangle K(y_1)y_2]\langle x_1x_2\rangle$$
$$b' = (x_1x_2)[\langle\ \rangle K(y_1)y_2]\langle y_1y_2\rangle$$
$$c = (z_1z_2)[\langle z_1\vec{v}\rangle\mathbf{out}, \langle z_2\rangle\mathbf{box}a(\vec{u})]\langle\vec{w}\rangle$$

Now consider the composite actions $b \cdot c$ and $b' \cdot c$; neither of them have a redex (unless due to K). However, further precomposing $\langle zz\rangle$ to each of these actions produces a redex in $\langle zz\rangle \cdot b \cdot c$ but not in $\langle zz\rangle \cdot b' \cdot c$. This is due to the fact that y_1, y_2 are *control bound* and no static operation can unify them. To deal with this aspect of molecular forms we consider factorisations relative to arbitrary substitutions for the imported names. Concretely, this is achieved by precomposing to a an arbitrary vector of data $\langle\vec{v}\rangle$ which we shall call an *environment*. Consider

$$\langle\vec{v}\rangle \cdot a = [\langle\,\rangle K(\vec{y})]\langle\{\vec{v}/\vec{x}\}\vec{z}\rangle$$

We can now factorise $\langle\vec{v}\rangle \cdot a$ into $(\,)[\langle\,\rangle K(\vec{y})]\langle\vec{y}\rangle \cdot (\vec{y})\langle\{\vec{v}/\vec{x}\}\vec{z}\rangle$. The component $(\vec{y})\langle\{\vec{v}/\vec{x}\}\vec{z}\rangle$ is sufficient to determine which substitutions will be created in any action postcomposed to $\langle\vec{v}\rangle \cdot a$. Notice that such components are all of the form $(\vec{x})\langle\vec{y}\rangle$.

Molecules We note that the ability of two molecules to react depends on three factors: they must be constructed of complementary controls, one being **out** and the other **box**; their subject names must be identical; and finally, the links transmitted by the molecule $\langle x\vec{v}\rangle\mathbf{out}(\,)$ (represented by the names $\vec{v} : m$) must be of the same arity (m) as the links accepted by the molecule $\langle x\rangle\mathbf{box}a(\vec{y})$, in other words $a : m \rightarrow n$, for some n.

The labels, if they are to provide a basis for determining whether enough has been contributed to allow reaction, must contain sufficient information to describe these elements. Moreover, the labels must also identify whether the subject names are bound: that a subject name is bound does not necessarily render a molecule inaccessible to a complementary one, as the fourth example in the previous section shows. Note that, as with our consideration of the exported names above, we must also distinguish between bindings which originate from the import of the action with those that originate from controls. Again, we will employ environments for this purpose.

We shall choose to represent the molecular contributions of an action by means of particles, each of which will contain information regarding the subject name, type (**out** or **box**) and the arity of the links handled. Since we have just two types, we can represent the particles as a disjoint sum of pairs of names and arities. The binding will be represented as for the exported names. Thus, a possible concrete representation of the molecular contribution of an action is as $(\vec{u})\vec{\alpha}$ where the bindings are given by $(\vec{u})$ and each particle $\alpha \in (X \times M) + (X \times M)$.

τ particles For the purposes of our semantics, we shall choose to keep track of any redex which has been reduced. This will allow us to obtain a strong semantics, in the spirit of strong bisimilarity familiar in the mainstream process algebriac setting. To achieve this we will introduce an additional kind of particle, τ which we shall assume to be distinct from any other particle defined above.

The exported names and the molecules are distinct contributions but both share the same kind of binding considerations. Morever, as our last two examples in section 1 have indicated, some redexes can only be discovered by considering both kinds of contribution arising from the same action. These points make a case for combining the descriptions of these two kinds of contribution to give a single label. That is what we shall do:

Definition 5.4 (Labels) *Ranged over by ℓ, labels have the form:*

$$(\vec{u})\vec{\alpha}\langle\vec{v}\rangle$$

where each particle *β in $\vec{\alpha}$ (the* body *of ℓ) is in $((X \times M) + (X \times M)) \cup \{\tau\}$, where $\tau \notin X$. We shall associate a pair of arities with the body $\vec{\alpha}$ of a label as follows:*

$$\begin{array}{rcll} \langle 0, \langle x, m\rangle\rangle & : & \epsilon \rightarrow m & \\ \langle 1, \langle x, m\rangle\rangle & : & m \rightarrow \epsilon & \\ \tau & : & \epsilon \rightarrow \epsilon & \\ \vec{\alpha}_1\vec{\alpha}_2 & : & k_1 \otimes k_2 \rightarrow l_1 \otimes l_2 & \quad (\vec{\alpha}_i : k_i \rightarrow l_i) \end{array}$$

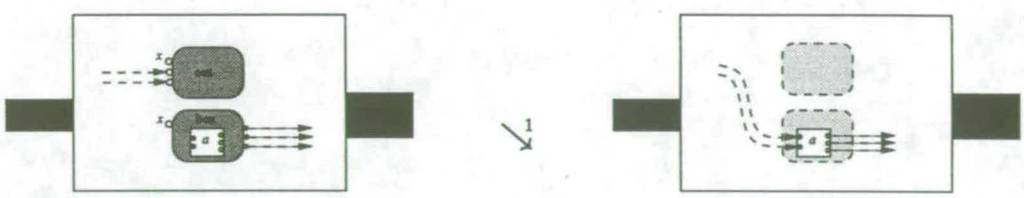
The names $\vec{u}$ are distinct and each name in $\vec{u}$ is binding throughout the label. If a name occurring in ℓ is not bound (i.e. does not occur in $\vec{u}$) then it is called free. *We denote the free names of ℓ by* $\mathbf{fn}(\ell)$. *Name substitution on labels $\{y/x\}\ell$ replaces each free x in ℓ by y renaming bound names to avoid capture. Labels which differ only up to alphaconversion and commutation of τ-particles with any particle in the body of the label are considered identical.* ■

Notation We shall often abbreviate $\langle 0, \langle x, m\rangle\rangle$ to $\overline{x}$ and $\langle 1, \langle x, m\rangle\rangle$ to x when we do not need to refer to the associated m. Each name in PIC′ is associated with a prime arity. The name x, its prime arity p (we write $x : p$) and arity m are called the *subject name*, *subject arity* and *object arity* respectively of the particle in each case. The *object arity* of a label $\ell = (\vec{u})\vec{\alpha}\langle\vec{v}\rangle$, written $|\ell|$ is $m \rightarrow n$ just when $\vec{\alpha} : m \rightarrow n$. If $\vec{u} : m$ and $\vec{v} : n$, the *subject arity* of ℓ is $m \rightarrow n$, written $\ell : m \rightarrow n$. We shall denote the set of labels by $\mathcal{L}$.

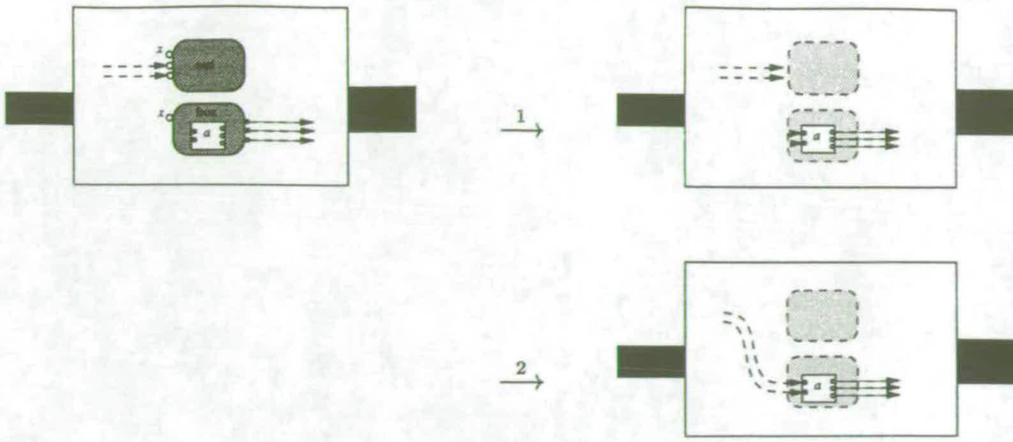
5.2.2 Labelled transition sequents

We shall now describe the next step towards obtaining a collection of relations on the terms of PIC^r which allow us not only to determine the reaction of its actions, but also to elicit the contribution that each action is able to make towards redex formation and the outcome of the resulting reaction under arbitrary contexts.

To explain the role of the labels in describing redex formation and that of labelled transitions in predicting reaction and its outcome, it is best to consider what happens when a reaction takes place between two complementary particles in a redex $\mathbf{out}_x \otimes \mathbf{box}_x a$, where $\mathbf{out}_x : m \to \epsilon$ and $a : m \to n$. The diagram below shows these two molecules side by side ready to react.



The effect of the reaction is the creation of links of arity (or width) m from the input of the $\mathbf{out}_x$ particle to the action contained within the **box** construct. One may view this occurrence as two distinct steps: the first consisting of the controls disintegrating, leaving, in the case of **out** dangling links of width m and, in the case of $\mathbf{box}a$, the exposed action a whose import links (also of width m) are also dangling, waiting for connection with those arising from $\mathbf{out}_x$; the second step establishes the connection itself, in other words, joins the dangling links. The latter step, however, involves a static or dataflow operation. One may think of the first step as a *partial reaction* and the second as a *synchronisation* of partial reactions to produce a completed computational step, or reaction.



In this way we can break the outcome of a reaction into the effect suffered by the participants (the dangling output links in the case of **out** and the exposed a, with its dangling import links, in the case of **box**a) and the static operation of connecting the relevant links. This will allow us to write a labelled transition to represent partial reactions; in other words, the contribution an action can make to a reaction (the label) and the effect it will suffer as a result (the residual), should that reaction occur. In fact, the τ particles will also permit us to record *completed* reaction as well.

Our formal representation of this idea consists of four components: the term describing the action under consideration, called the *principal term*; the *environment* which is a vector of names, causing the import bound names in the action to be replaced by free ones; the *label*, whose role we have described above; and finally, the *residual term*, which describes the action with dangling links in place of each molecule indicated in the label.

Definition 5.5 (Transition sequents) *A* labelled transition sequent *has the form:*

$$\langle \vec{z} \rangle \vdash t \xrightarrow{\ell} t'$$

to be read as: under environment $\vec{z}$, *the* principal term t *goes to the* residual term t' *performing* label ℓ. *Such sequents are well-formed just when*

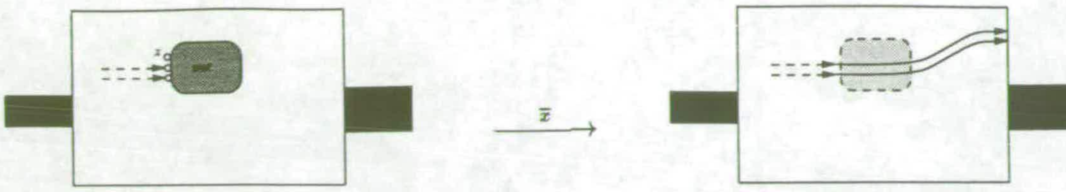
$$
\begin{aligned}
\vec{z} &: m \\
t &: m \rightarrow n \\
t' &: k \rightarrow l \otimes r \\
\ell &: r \rightarrow n \\
|\ell| &= k \rightarrow l
\end{aligned}
$$

■

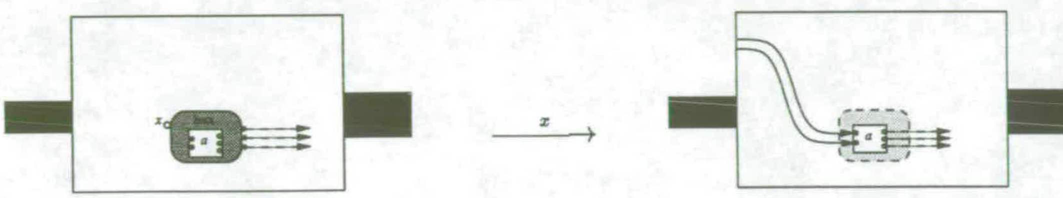
The arity rule for sequents is best explained with reference to the intended interpretation for the sequents. Consider the sequent $\langle\vec{z}\rangle \vdash t \xrightarrow{\ell} t'$. The environment $\langle\vec{z}\rangle$ can be considered as supplying names $\vec{z}$ to t in an identical fashion as occurs in the composite $\langle\vec{z}\rangle \cdot t$. This ensures that the import-bound names in the molecular form of t are replaced by free names, thereby ensuring that any bound name (in the molecular form of the composite $\langle\vec{z}\rangle \cdot t$) is control-bound. Hence, in order for the term $\langle\vec{z}\rangle \cdot t$ to be well formed, whenever $t : m \rightarrow n$, then $\vec{z}$ must have arity m.

Let $\ell = (\vec{u})\vec{\alpha}\langle\vec{v}\rangle$. The part $(\vec{u}) \cdots \langle\vec{v}\rangle$ reflects the exported names $\vec{v}$ of the molecular form of t, of which $\vec{u}$ are bound by controls (including ν, see Discussion below). This essentially signifies a factorisation of $\langle\vec{z}\rangle \cdot t$ into composites t'' (for some such) and $(\vec{u})\langle\vec{v}\rangle$. Thus, if $t : m \rightarrow n$ and $\vec{u} : r$, then $t'' : \epsilon \rightarrow r$ and $\vec{v} : n$.

We shall now account for the emergence of the subject arities $|\ell| : k \rightarrow l$. Informally, if the label ℓ contains the particle $\overline{x} : \epsilon \rightarrow h$ it indicates the existence of a molecule $\langle x\vec{w}\rangle\mathbf{out}$, with $\vec{w} : h$, in the body of the molecular form of $\langle\vec{z}\rangle \cdot t$. Moreover, this same molecule is assumed to have partially reacted in the residual t'. Since we do not know at this point, with which other action or molecule the reaction will take place (i.e. where the complementary part of the redex will come from) we are left with a dangling link of width h (indicated by the output arity of the particle). This link, which originated from an output port, is ready to "connect" with a link arising from a complementary input port. Until this occurs, the link is placed alongside the *exported* links in the residual. The particle $\overline{x} : \epsilon \rightarrow h$ in the label ℓ records that a link of width h is dangling at the export interface of the residual, waiting for connection with any recipient made available through the reduction of the complementary part of the redex.



In this case, such a part must come from a $\mathbf{box}_x a$ molecule, for some $a : h \to h'$. Such a contribution would be reflected as a particle $x : h \to \epsilon$ in the label: the links into a will similarly be made available at the *imports* of the residual (which also includes the action a which has been released from within the **box** construct).

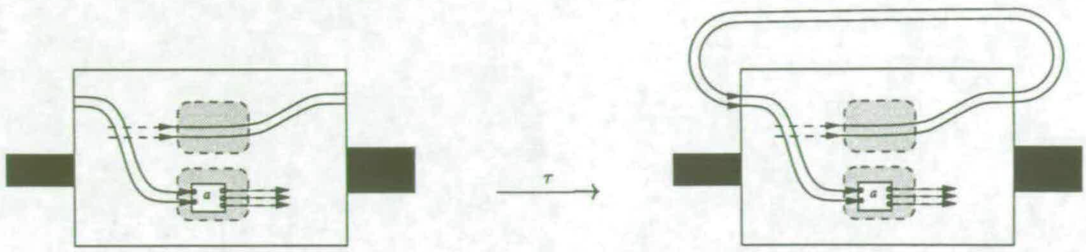


Hence, for each particle in $\vec{\alpha}$ we get an associated increase in arity either to the input or to the output of the residual according to the type (input or output) and subject arity of the particle. Thus, in the above, t' is obtained by redirecting in t'' the appropriate links; those of width k to the import and those of width l to the export resulting in the arity $t' : k \to l \otimes r$. Consider, for instance, the transition

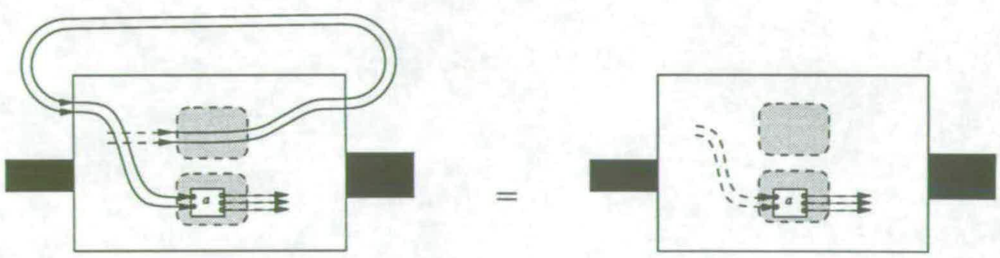
$$\langle\vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\cdots\overline{x}y\cdots\langle\vec{v}\rangle} t'$$

which exposes the existence, in the molecular form of $\langle\vec{z}\rangle \cdot t$, of molecules $\langle x\vec{w}\rangle\mathbf{out}(\ )$ and $\langle y\rangle\mathbf{box}a(\vec{w}')$ (for some $a, \vec{w}, \vec{w}'$) together with exported names $\vec{v}$ with names $\vec{u}$ bound by controls. The names $\vec{w}'$, which are control bound are included in $\vec{u}$, the binding vector occurring in the label. The residual t' contains the links (represented by the names $\vec{w}$ in the molecule $\langle x\vec{w}\rangle\mathbf{out}(\ )$) at its export interface and the links into a at its import interface.

If the same action contains two complementary molecules, then it will have a transition with both $\overline{x}$ and x (for some x) in its label. These complementary molecules can react together, and the result of this reaction can be obtained by connecting, in the residual, the dangling output links to the corresponding input links arising from the partial reactions recorded by the complementary particles in the label. In order to achieve this connection from export to import positions in the residual we need feedback, as provided by the reflexion operation.



This is essentially the idea behind the synchronisation rule SYNC. The occurrence of such a synchronisation is recorded in the label by replacing the complementary particles $\overline{x}, x$ with τ. Since completed reaction does not add any links to the residual (i.e. preserves the arities) the arity of a τ particles is $\epsilon \rightarrow \epsilon$.



Thus, in summary, named particles (in a label) indicate partial reaction, while each τ particle records the sychronisation of partial reactions to achieve completed computational steps (reaction).

Discussion We note that in PIC^r there are two sorts of binding molecule: $\nu(\vec{u})$ and $\langle x \rangle \mathbf{box}a(\vec{u})$. A more constrained version of PIC^r can be obtained by limiting binding to restriction molecules. This can be done by replacing the arity rule for $\mathbf{box}a$ as follows, ensuring that such molecules will be of the form $\langle x \rangle \mathbf{box}a(\;)$:

$$\frac{a : m \rightarrow \epsilon}{\mathbf{box}a : p \rightarrow \epsilon}$$

This constraint does not simplify (at least, not in a direct way) any aspect of our semantics. It does, however, render PIC′ somewhat closer to the original π-calculus. Also, we will then be justified in writing each label $(\vec{u})\vec{\alpha}\langle\vec{v}\rangle$ as $(\nu\vec{u})\vec{\alpha}\langle\vec{v}\rangle$. Such occurrence of restriction in labels is not new; Sangiorgi employs such in his treatment of the higher order π-calculus[38]. On the other hand, we argue that this distinction from the π-calculus—that processes of arbitrary arities can fall within an input prefix—is natural in a world where the arities of processes are other than $\epsilon \to \epsilon$. We also note that, in the more complex setting where there are two kinds of binding molecule (ν and **box**, of which **box** takes an action argument), it is unwieldy to employ the same method used for dealing with such bindings in the labelled transition rules for the original π-calculus; namely the OPEN and CLOSE rules. Our use of reflexion avoids such special case treatment for sending and receiving bound data and the benefit is especially evident when, as in our case, the binding molecules are various and complex. We shall therefore refrain from constraining PIC′ as suggested but the reader should keep in mind that for any term of PIC′ that corresponds to a π-calculus term (for a precise correspondence see [29]), the bindings in labels originate solely from restriction molecules.

5.2.3 Labelled transition rules

We shall now describe a set of rules which allow the transition sequents to be derived, formalising the interpretation we have described above. The rules $\mathcal{R}$ are presented in figures 5–2, 5–3 and 5–4 and in the relevant rules we assume $\vec{\alpha}_i : k_i \to l_i$.

Inspection reveals three kinds of rule: *constructor* rules, which eliminate (from conclusion to premise) the outermost constructor of the principal term, *permutation* rules which permute either the particles or the bindings of the label; and the *sychronisation* rule which is the only rule that introduces τ particles in the label. More interestingly, the constructor rules are responsible for eliciting the contributions that actions may make towards redex formation, in particular, the partial reactions. Each rule performs two functions: from the labels and residuals of the subactions (the labelled transitions of the premises) the rule tells us how to compute the combined label (or, aggregate contribution) and residual resulting from applying the principal constructor to the subactions.

Consider the rules of figure 5–2; in each case, the action resulting from precomposing the environment to the term is analysed and the contribution of exported names (free and bound), partial reactions, and completed ones are included in the label. Note that,

$$\langle x\rangle\ \frac{}{\vdash \langle x\rangle \xrightarrow{(\ )\langle x\rangle} \mathbf{id}_\epsilon} \qquad\qquad \omega\ \frac{}{\langle z\rangle \vdash \omega \xrightarrow{(\ )\langle\ \rangle} \mathbf{id}_\epsilon}$$

$$\mathbf{id}\ \frac{}{\langle \vec{z}\rangle \vdash \mathbf{id} \xrightarrow{(\ )\langle \vec{z}\rangle} \mathbf{id}_\epsilon}\ ,$$

$$\mathbf{out}_1\ \frac{}{\langle \vec{z}\rangle \vdash \mathbf{out} \xrightarrow{(\ )\langle\ \rangle} \langle \vec{z}\rangle \cdot \mathbf{out}} \qquad \mathbf{box}_1\ \frac{}{\langle x\rangle \vdash \mathbf{box}t \xrightarrow{(\vec{u})\langle \vec{u}\rangle} \langle x\rangle \cdot \mathbf{box}t}$$

$$\mathbf{out}_2\ \frac{}{\langle x\vec{z}\rangle \vdash \mathbf{out} \xrightarrow{(\ )\overline{x}\langle\ \rangle} \langle \vec{z}\rangle} \qquad \mathbf{box}_2\ \frac{}{\langle x\rangle \vdash \mathbf{box}t \xrightarrow{(\vec{u})x\langle \vec{u}\rangle} t}\ x \notin \{\vec{u}\}$$

Figure 5–2: Labelled transition rules

in rules $\mathbf{out}_2$ and $\mathbf{box}_2$, the residual registers an increase in the output and input arities respectively. In rule $\mathbf{out}_2$, the data leading into the port $\mathbf{out}_x$ is made available at the export of the residual, while in the rule $\mathbf{box}_2$, the inputs to the term t (contained within the principal term $\mathbf{box}t$) are made available at the imports of the residual (t itself).

The rules of figure 5–3 appear somewhat more complex. In the residual of the conclusion sequent, the links corresponding to the particles in the label must be placed in the correct positions at the import and export. This is achieved by organising the dataflow between the residuals of the premise sequents. The considerable extent of "wiring" necessary gives the appearance of complexity to the rules; however, each is designed upon the same principle that the particle sequence in the labels must reflect the positions of the links created by the partial reactions.

Consider, for instance, the composition rule. The subject arity of the labels $\ell_i = (\vec{u}_i)\vec{\alpha}_i\langle\vec{v}_i\rangle$ is $k_i \to l_i$. Hence, the term t_1' has k_1 import and l_1 export links due to the partial reactions, whereas t_2' has k_2 and l_2 import and export links respectively. When combining t_1' and t_2' to get the residual, we must ensure that the mentioned export links of t_1' are passed to the topmost position, hence the occurrences of $\mathbf{id}_{l_1}$ in the residual term. Similarly, the import links of t_2' necessitate the occurrence of $\mathbf{id}_{k_2}$ to ensure that the links are connected to the imports in the residual. The use of abstraction (viz. $\mathbf{ab}_{\vec{u}}t_2'$) in the residual is due to the fact that in obtaining t_2', the transition of t_2 is

$$\circ\ \frac{\langle\vec{z}\rangle \vdash t_1 \xrightarrow{(\vec{u}_1)\vec{\alpha}_1\langle\vec{v}_1\rangle} t_1' \qquad \langle\vec{v}_1\rangle \vdash t_2 \xrightarrow{(\vec{u}_2)\vec{\alpha}_2\langle\vec{v}_2\rangle} t_2'}{\langle\vec{z}\rangle \vdash t_1 \cdot t_2 \xrightarrow{(\vec{u}_1\vec{u}_2)\vec{\alpha}_1\vec{\alpha}_2\langle\vec{v}_2\rangle} (t_1' \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}_1} t_2') \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r,l_2} \otimes \mathbf{id})} \quad \begin{array}{l}\{\vec{u}_1\} \cap \mathbf{fn}(t_2) = \emptyset \\ \{\vec{u}_2\} \cap \mathbf{fn}(t_1, \vec{z}) = \emptyset \\ \vec{u}_1 : r\end{array}$$

$$\otimes\ \frac{\langle\vec{z}_1\rangle \vdash t_1 \xrightarrow{(\vec{u}_1)\vec{\alpha}_1\langle\vec{v}_1\rangle} t_1' \qquad \langle\vec{z}_2\rangle \vdash t_2 \xrightarrow{(\vec{u}_2)\vec{\alpha}_2\langle\vec{v}_2\rangle} t_2'}{\langle\vec{z}_1\vec{z}_2\rangle \vdash t_1 \otimes t_2 \xrightarrow{(\vec{u}_1\vec{u}_2)\vec{\alpha}_1\vec{\alpha}_2\langle\vec{v}_1\vec{v}_2\rangle} (t_1' \otimes t_2') \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r,l_2} \otimes \mathbf{id})} \quad \begin{array}{l}\{\vec{u}_1\} \cap \mathbf{fn}(t_2, \vec{z}_2) = \emptyset \\ \{\vec{u}_2\} \cap \mathbf{fn}(t_1, \vec{z}_1) = \emptyset \\ \vec{u}_1 : r\end{array}$$

$$\mathbf{ab}\ \frac{\langle\vec{z}\rangle \vdash \{{}^{w}\!/\!{}_{y}\}t \xrightarrow{(\vec{u})\vec{\alpha}\langle\vec{v}\rangle} t'}{\langle w\vec{z}\rangle \vdash \mathbf{ab}_y t \xrightarrow{(\vec{u})\vec{\alpha}\langle w\vec{v}\rangle} t'} \quad w \notin \{\vec{u}\}$$

$$\uparrow_1\ \frac{\langle y\vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\vec{\alpha}\langle y\vec{v}\rangle} t'}{\langle\vec{z}\rangle \vdash \uparrow t \xrightarrow{(y\vec{u})\vec{\alpha}\langle\vec{v}\rangle} (\nu y)(t' \cdot (\mathbf{id}_l \otimes \langle y\rangle \otimes \mathbf{id}))} \quad y \notin \mathbf{fn}(t) \cup \{\vec{z}\}$$

$$\uparrow_2\ \frac{\langle y\vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\vec{\alpha}\langle w\vec{v}\rangle} t'}{\langle\vec{z}\rangle \vdash \uparrow t \xrightarrow{(\vec{u})\{w/y\}\vec{\alpha}\langle\{w/y\}\vec{v}\rangle} \uparrow(y)(t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle w\vec{u}\rangle) \cdot (\mathbf{p}_{l,p} \otimes \mathbf{id}))} \quad \begin{array}{l} y \notin \mathbf{fn}(t) \cup \{\vec{u}, \vec{z}\} \\ w \neq y \\ w, y : p\end{array}$$

Figure 5–3: Labelled transition rules

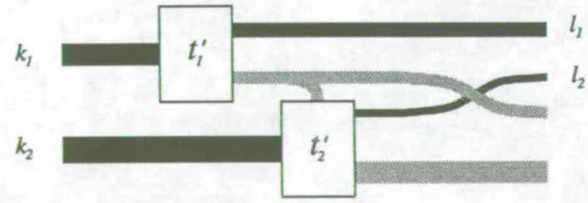
$$\text{SYNC}\frac{\langle\vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\overline{x}x\vec{\alpha}\langle\vec{v}\rangle} t'}{\langle\vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\tau\vec{\alpha}\langle\vec{v}\rangle} \uparrow_m t'} \quad \begin{array}{l} x : m \to \epsilon \\ \overline{x} : \epsilon \to m \end{array}$$

$$\text{PERM}_1\frac{\langle\vec{z}\rangle \vdash t \xrightarrow{(\vec{u}_1\vec{x}_1\vec{x}_2\vec{u}_2)\vec{\alpha}\langle\vec{v}\rangle} t'}{\langle\vec{z}\rangle \vdash t \xrightarrow{(\vec{u}_1\vec{x}_2\vec{x}_1\vec{u}_2)\vec{\alpha}\langle\vec{v}\rangle} t' \cdot (\mathbf{id}_{l\otimes r} \otimes \mathbf{p}_{m_1,m_2} \otimes \mathbf{id})} \quad \begin{array}{l} \vec{x}_i : m_i \\ \vec{u}_1 : r \end{array}$$

$$\text{PERM}_2\frac{\langle\vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\vec{\alpha}_1\vec{\beta}_1\vec{\beta}_2\vec{\alpha}_2\langle\vec{v}\rangle} t'}{\langle\vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\vec{\alpha}_1\vec{\beta}_2\vec{\beta}_1\vec{\alpha}_2\langle\vec{v}\rangle} (\mathbf{id}_{k_1} \otimes \mathbf{p}_{m_2,m_1} \otimes \mathbf{id}) \cdot t' \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{n_1,n_2} \otimes \mathbf{id})} \quad \vec{\beta}_i : m_i \to n_i$$

Figure 5–4: Labelled transitions rules

derived under the environment $\langle\vec{v}_1\rangle$, some of whose names may be bound by $\vec{u}_1$. Finally, the permutor $\mathbf{p}_{r,l_2}$ is necessary to place the export links arising from t'_1 and t'_2 alongside each other.



The rule for tensor can be explained in a similar manner; to obtain the residual, one must direct the topmost l_2 links of t'_2 to the topmost position under the l_1 export links of t'_1. The abstraction rule is straightforward: note the inclusion of w in the export vector in the label alongside $\vec{v}$.

Arguably the most complex rules are those for reflexion. The complexity is partly due to the complexity of the operation itself, as defined on the underlying molecular forms. As in the definition of reflexion on the molecular forms, there are two cases to consider; one in which a link is being reflected onto itself, and the other when this is not the case. In order to detect the occurrence of a link being reflected onto itself, some fresh name y is fed into the input of the topmost import position. If the same name emerges at the other end in the topmost position of the export, then (by virtue of y being fresh) it may be concluded that a link from the topmost import position to the topmost export position is present in t. The molecular form of the residual will consequently include a restriction particle as expressed in rule $\uparrow_1$. Note that any free occurrences of y in the label $(\vec{u})\vec{\alpha}\langle\vec{v}\rangle$ are bound by the restriction particle in molecular form of the term t, hence the introduction of the binding occurrence of y in the label for the conclusion sequent.

In the rule $\uparrow_2$, no restriction particle is introduced by reflexion in the molecular form of $\langle\vec{z}\rangle \cdot \uparrow t$. Note that the name w may or may not be bound by $\vec{u}$. To deal with both cases, the subterm $(\vec{u})\langle w\vec{u}\rangle$ is employed in the residual, with w fed back into t' through the abstraction of y.

So far, all the rules discussed eliminate (towards the premises) the principal term constructor. The rules which we shall now discuss employ identical principal terms (and environments) in both premise and conclusion sequents. There are two permutation rules PERM$_1$ and PERM$_2$ which respectively permute the binding vector and the particle sequence of the label. The latter operation on labels allows complementary particles to migrate towards the required position to permit synchronisation to be derived. In each case the links in the residual corresponding to bindings or particles in the label have to be rerouted to maintain the proper correspondence.

The synchronisation rule identifies the existence of dangling links of equal width which can be joined as a result of reaction. This is indicated by the presence of a complementary pair of particles $\overline{x}, x$ at the rightmost position in the label: $\overline{x} : \epsilon \to m$ indicates the presence of links of width m at the export of t' while $x : m \to \epsilon$ indicates that m links lie at the import. Moreover, since the particles bear the same name x, the links must have arisen from complementary molecules. All is ready to join them: this is achieved by reflecting the topmost m links of t'. This event is marked by the replacement of the rightmost complementary pair by a τ particle in the label.

Note that one sychronisation rule suffices to detect all possible redex formations in any term of PIC^r. This is a remarkable fact and is due to the work each rule performs in analysing the contribution to redex formation in each subterm, recording each such information in the label and preparing the residual for the outcome. It is hard to envisage how this could have been achieved without the use of reflexion.

Examples The following examples illustrate the use of most of the rules. We will first present two simple examples and then a more complex one which allows a comparison between unearthing, on the one hand, a redex by the structural manipulation of terms (by means of the axioms of AC′) and, on the other, eliciting a redex using the rules. Since the rules introduce rather a lot of dataflow, even for simple cases, we shall cope with the complexity of residual terms by writing instead terms which are equal. For this end we shall adopt the notation

$$\langle\vec{z}\rangle \vdash t \xrightarrow{\ell}_{=} t'$$

to mean that for some t'', $\langle\vec{z}\rangle \vdash t \xrightarrow{\ell} t''$ is derivable and $t' = t''$. This is justified, first, because in none of the rules do the premisses or side conditions refer to any property of the residual terms; and second, because we will later show that any two equal terms derive identical transitions to equal residual terms.

1. We shall begin by deriving the reaction $(\langle xv\rangle \cdot \mathbf{out}) \otimes \mathbf{box}_x(y)t \searrow \{{}^{v}\!/\!_{y}\}t$ using the rules. For simplicity we shall assume that $t : \epsilon \to \epsilon$. A τ transition signals the performance of a single computational step; effectively a single use of the reaction rule for PIC′. The derivation of the transition $\vdash ((\langle xv\rangle \cdot \mathbf{out}) \otimes ((\langle x\rangle \cdot \mathbf{box}(y)t) \xrightarrow{\tau}_{=} \{{}^{v}\!/\!_{y}\}t$ is given below:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \langle x\rangle \xrightarrow{\langle x\rangle} \mathbf{id}_\epsilon} \quad \dfrac{}{\vdash \langle v\rangle \xrightarrow{\langle v\rangle} \mathbf{id}_\epsilon}}{\vdash \langle xv\rangle \xrightarrow{\langle xv\rangle}_{=} \mathbf{id}_\epsilon}\otimes \quad \dfrac{}{\langle xv\rangle \vdash \mathbf{out} \xrightarrow{\bar{x}} \langle v\rangle}}{\vdash \langle xv\rangle \cdot \mathbf{out} \xrightarrow{\bar{x}}_{=} \langle v\rangle}\circ \quad \dfrac{\dfrac{}{\vdash \langle x\rangle \xrightarrow{\langle x\rangle} \mathbf{id}_\epsilon} \quad \dfrac{}{\langle x\rangle \vdash \mathbf{box}(y)t \xrightarrow{x} (y)t}}{\vdash \langle x\rangle \cdot \mathbf{box}(y)t \xrightarrow{x}_{=} (y)t}\circ}{\vdash ((\langle xv\rangle \cdot \mathbf{out}) \otimes ((\langle x\rangle \cdot \mathbf{box}(y)t) \xrightarrow{\bar{x}x}_{=} \langle v\rangle \otimes (y)t}\otimes}{\vdash ((\langle xv\rangle \cdot \mathbf{out}) \otimes ((\langle x\rangle \cdot \mathbf{box}(y)t) \xrightarrow{\tau}_{=} \{{}^{v}\!/\!_{y}\}t}\ \textsc{sync}$$

To see that the residual term is indeed equal to $\{{}^{v}\!/\!_{y}\}t$, consider that, by the last rule use, the residual term should be equal to $\uparrow_p(\langle v\rangle \otimes (y)t)$. By lemma 3.29(1), this term is equal to $\{{}^{v}\!/\!_{y}\}t$.

2. The following example illustrates the use of the first reflexion rule $\uparrow_1$. We expect the following rule to be derivable (modulo provable equality):

$$\boldsymbol{\nu}\dfrac{}{\vdash \boldsymbol{\nu} \xrightarrow{(x)\langle x\rangle} \boldsymbol{\nu}}$$

A derivation for the transition $\vdash \boldsymbol{\nu} \xrightarrow{(x)\langle x\rangle}_{=} \boldsymbol{\nu}$ is given below:

$$
\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \langle x\rangle \xrightarrow{\langle x\rangle} \mathbf{id}_\epsilon} \quad \dfrac{}{\vdash \langle x\rangle \xrightarrow{\langle x\rangle} \mathbf{id}_\epsilon}}{\vdash \langle xx\rangle \xrightarrow{\langle xx\rangle}_{=} \mathbf{id}_\epsilon}\,\otimes}{\langle x\rangle \vdash \mathbf{ab}_x\langle xx\rangle \xrightarrow{\langle xxx\rangle}_{=} \mathbf{id}_\epsilon}\,\mathbf{ab} \qquad \dfrac{\dfrac{}{\langle x\rangle \vdash \boldsymbol{\omega} \xrightarrow{(\,)\langle\,\rangle} \mathbf{id}_\epsilon}\,\boldsymbol{\omega} \quad \dfrac{}{\langle xx\rangle \vdash \mathbf{id} \xrightarrow{\langle xx\rangle} \mathbf{id}_\epsilon}\,\mathbf{id}}{\langle xxx\rangle \vdash \boldsymbol{\omega}\otimes\mathbf{id} \xrightarrow{\langle xx\rangle}_{=} \mathbf{id}_\epsilon}\,\otimes}{\dfrac{\langle x\rangle \vdash \mathbf{ab}_x\langle xx\rangle\cdot(\boldsymbol{\omega}\otimes\mathbf{id}) \xrightarrow{\langle xx\rangle}_{=} \mathbf{id}_\epsilon}{\vdash \uparrow\ (\mathbf{ab}_x\langle xx\rangle\cdot(\boldsymbol{\omega}\otimes\mathbf{id})) \xrightarrow{(x)\langle x\rangle}_{=} (\boldsymbol{\nu}x)\langle x\rangle}\,\uparrow_1}\,\circ
$$

Note that the principal term is indeed $\boldsymbol{\nu}$ by $\boldsymbol{\nu} \stackrel{\text{def}}{=} \uparrow(x)\langle xx\rangle \equiv \uparrow(\mathbf{ab}_x\langle xx\rangle\cdot(\boldsymbol{\omega}\otimes\mathbf{id}))$. It is also clear that the residual is equal to $\boldsymbol{\nu}$, since $(\boldsymbol{\nu}x)\langle x\rangle = \boldsymbol{\nu}\cdot(x)\langle x\rangle = \boldsymbol{\nu}\cdot\mathbf{id} = \boldsymbol{\nu}$.

3. We shall now present an example of a term which requires complex structural manipulation for the redex to become apparent. The rules we have given cannot manipulate the principal term structurally—this is indeed their very source of power, which permits redex formation to be analysed in a systematic way. Thus, the rules remove the need for structural manipulations by extracting redex contributions from terms *in situ*. The following example demonstrates this process.

   Consider the term $\uparrow((\boldsymbol{\nu}\cdot\mathbf{ab}_x\mathbf{out}_x)\otimes\mathbf{box}t)$, where, for simplicity, we take $t:\epsilon\rightarrow\epsilon$. We shall first derive reaction by unearthing a redex using equational manipulation of the term. Then, for comparison, the same redex will be reduced through a suitable derivation. We shall assume, for simplicity, that $x, y \notin \mathbf{fn}\,(t)$:

$$
\begin{array}{lll}
\multicolumn{2}{l}{\uparrow((\boldsymbol{\nu}\cdot\mathbf{ab}_x\mathbf{out}_x)\otimes\mathbf{box}t)} & \\
= & \uparrow((\boldsymbol{\nu}\otimes\mathbf{id})\cdot(\mathbf{ab}_x\mathbf{out}_x\otimes\mathbf{box}t)) & \\
= & \uparrow((\mathbf{id}\otimes\boldsymbol{\nu})\cdot\mathbf{p}_{p,p}\cdot(\mathbf{ab}_x\mathbf{out}_x\otimes\mathbf{box}t)) & \zeta \\
= & \boldsymbol{\nu}\cdot\uparrow(\mathbf{p}_{p,p}\cdot(\mathbf{ab}_x\mathbf{out}_x\otimes\mathbf{box}t)) & \rho_3 \\
= & \boldsymbol{\nu}\cdot\uparrow(\mathbf{p}_{p,p}\cdot(x)(\langle x\rangle\otimes\mathbf{out}_x\otimes\mathbf{box}t)) & 2.16(4) \\
= & \boldsymbol{\nu}\cdot(x)\uparrow(\langle x\rangle\otimes\mathbf{out}_x\otimes\mathbf{box}t) & \rho_5 \\
= & \boldsymbol{\nu}\cdot(x)\uparrow(\langle x\rangle\otimes\mathbf{out}_x\otimes((y)(y)\cdot\mathbf{box}t)) & \delta \\
= & \boldsymbol{\nu}\cdot(x)\uparrow(\langle x\rangle\otimes\mathbf{out}_x\otimes(y)((y)\cdot\mathbf{box}t)) & 2.16(1) \\
= & \boldsymbol{\nu}\cdot(x)\uparrow(y)(\langle x\rangle\otimes\mathbf{out}_x\otimes\mathbf{box}_y t) & 2.16(3) \\
= & \boldsymbol{\nu}\cdot(x)(\mathbf{out}_x\otimes\mathbf{box}_x t) & 3.29(1) \\
\searrow & \boldsymbol{\nu}\cdot(x)t & \\
= & \boldsymbol{\nu}\cdot(\boldsymbol{\omega}\otimes t) & \gamma \\
= & (\boldsymbol{\nu}\otimes t)\cdot(x)\mathbf{id}_\epsilon &
\end{array}
$$

   A derivation for the transition representing the reduction of the same redex is given below:

$$
\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \nu \xrightarrow{(x)(x)}{}_{=}\nu} \quad \dfrac{\dfrac{}{\vdash \mathbf{out}_x \xrightarrow{\overline{x}}{}_{=}\mathbf{id}_\epsilon}}{x \vdash \mathbf{ab}_x\mathbf{out}_x \xrightarrow{\overline{x}\langle x\rangle}{}_{=}\mathbf{id}_\epsilon}\,\mathbf{ab}}{\vdash \nu\cdot\mathbf{ab}_x\mathbf{out}_x \xrightarrow{(x)\overline{x}\langle x\rangle}{}_{=}\nu}\,\circ \qquad \dfrac{}{y\vdash \mathbf{box}t \xrightarrow{y} t}\,\mathbf{box}_2}{\langle y\rangle \vdash (\nu\cdot\mathbf{ab}_x\mathbf{out}_x)\otimes\mathbf{box}t \xrightarrow{(x)\overline{x}y\langle x\rangle}{}_{=}\nu\otimes t}\,\otimes}{\vdash \uparrow\ ((\nu\cdot\mathbf{ab}_x\mathbf{out}_x)\otimes\mathbf{box}t) \xrightarrow{(x)\overline{x}x}{}_{=}\uparrow\ (y)((\nu\otimes t)\cdot(x)\langle yx\rangle)}\,\uparrow_2}{\vdash \uparrow\ ((\nu\cdot\mathbf{ab}_x\mathbf{out}_x)\otimes\mathbf{box}t) \xrightarrow{(x)\tau}{}_{=}\nu\otimes t}\,\text{SYNC}
$$

Note that the residual of the transition derived differs slightly from the residual of the reaction derived earlier. This is due to our decision to include in the label *all* of the control bound names occurring in the molecular form of the principal term. The restriction operation in the term $\uparrow((\nu\cdot\mathbf{ab}_x\mathbf{out}_x)\otimes\mathbf{box}t)$ gives rise to a restriction particle $\nu(x)$ in its molecular form, thereby causing the inclusion of the binding (x) in the label $(x)\tau$. Later we shall propose a way to eliminate such unnecessary bindings.

Discussion At first, the rules for deriving labelled transitions may appear complex. Is their complexity justified? There are indeed alternatives which may be simpler in some sense. For instance, we can rewrite the rules for the special case when at most one particle is present in the label being derived[2]. If we write a rule for deriving transitions treating separately each label containing a different type of particle or having an empty body, the rules will be much simplified because in each case, some (in some cases, all) of the subject arities will be ϵ. Here is one of the rules for deriving transitions with label $(\vec{u})\overline{x}\langle\vec{v}\rangle$ for composition; for ease of comparison with our composition rule, we let $\overline{x} : \epsilon \to l_1$:

$$
\dfrac{\langle\vec{z}\rangle \vdash t_1 \xrightarrow{(\vec{u}_1)\overline{x}\langle\vec{v}_1\rangle} t_1' \quad \langle\vec{v}_1\rangle \vdash t_2 \xrightarrow{(\vec{u}_2)\langle\vec{v}_2\rangle} t_2'}{\langle\vec{z}\rangle \vdash t_1\cdot t_2 \xrightarrow{(\vec{u}_1\vec{u}_2)\overline{x}\langle\vec{v}_2\rangle} t_1'\cdot(\mathbf{id}_{l_1}\otimes\mathbf{ab}_{\vec{u}_1}t_2')} \quad \begin{array}{l}\{\vec{u}_1\}\cap\mathbf{fn}(t_2)=\emptyset\\ \{\vec{u}_2\}\cap\mathbf{fn}(t_1,\vec{z})=\emptyset\end{array}
$$

In this case, using our convention for the subject arity of labels in our composition rule, the arities k_1, k_2 and l_2 are all ϵ.

The disadvantage of this approach is that the number of rules required would be much greater than the ones we have presented. For composition alone, we would need

[2]The synchronisation rule would also have to be changed to allow τ transitions to be derived from single-particle labels.

no fewer that seven rules! Moreover, we lose the ability to derive transitions whose labels have multiple particles which allows us to derive a non-interleaving semantics, besides the interleaving semantics that may still be obtained by our system by considering only transitions with labels having at most one particle.

We suggest that such complexity is not excessive given the presence of actions of input and output arities greater than ϵ and the existence of operations such as abstraction, composition and reflexion.

5.3 Labelled Transition Relations

We are now ready to define a collection of labelled transition relations on terms in the familiar manner. One outcome of this is that the standard notion of bisimilarity can be used to give an operational model to our calculus.

Definition 5.6 *For any two terms t, t' and label ℓ, (t, t') are in the relation $\xrightarrow{\langle\vec{v}\rangle,\ell}$ just when the labelled transition $\langle\vec{v}\rangle \vdash t \xrightarrow{\ell} t'$ is derivable by the rules $\mathcal{R}$.* ■

Notation We shall henceforth write $\langle\vec{v}\rangle \vdash t \xrightarrow{\ell} t'$ to mean that (t, t') are in the relation $\xrightarrow{\langle\vec{v}\rangle,\ell}$. In other words, it asserts that the sequent $\langle\vec{v}\rangle \vdash t \xrightarrow{\ell} t'$ is derivable by the rules $\mathcal{R}$.

The main result in this chapter states that terms equal in AC^r have identical transitions to equal residual terms. This immediately provides a well-defined notion of labelled transition relations on the actions of PIC^r. In order to show this result we must first establish a number of properties of the derivations. The first lemma shows that the free names of both the label and the residual come from the environment and the principal term.

Lemma 5.7 (Free names) *Whenever $\langle\vec{z}\rangle \vdash t \xrightarrow{\ell} t'$ then $\mathbf{fn}(\ell) \subseteq \mathbf{fn}(t) \cup \{\vec{z}\}$ and $\mathbf{fn}(t') \subseteq \mathbf{fn}(t) \cup \{\vec{z}\}$.*

Proof Induction on the depth of derivation of $\langle\vec{x}\rangle \vdash t \xrightarrow{\ell} t'$. ■

The following lemma shows that name substitution in both the environment and the principal term is carried over to the label and the residual. Moreover, such substitutions (applied to both the environment and the principal term) do not give rise to additional transitions which cannot be accounted for simply by the substitution on the label and residual.

Lemma 5.8 (Substitution) *Let δ range over all labels not containing τ-particles. Then,*

1. $\langle \vec{z} \rangle \vdash t \xrightarrow{\ell} t' \implies \langle \{y/x\}\vec{z} \rangle \vdash \{y/x\}t \xrightarrow{\{y/x\}\ell}{}_{=} \{y/x\}t'$;
2. $\langle \{y/x\}\vec{z} \rangle \vdash \{y/x\}t \xrightarrow{\delta} t' \implies \exists t'', \delta'.\ \langle \vec{z} \rangle \vdash t \xrightarrow{\delta'} t''$ *with* $t' = \{y/x\}t''$ *and* $\delta = \{y/x\}\delta'$.

Proof Induction on the depth of derivation of premise transition. ∎

Remark To see why it was necessary to impose the constraint on the labels in (2) above, consider the transition:

$$\vdash \{y/x\}(\mathbf{out}_x \otimes \mathbf{box}_y t) \xrightarrow{(\vec{u})\tau\langle\vec{u}\rangle} \{y/x\}t$$

For any ℓ, if $\{y/x\}\ell = (\vec{u})\tau\langle\vec{u}\rangle$ then $\ell = (\vec{u})\tau\langle\vec{u}\rangle$. However, no such labelled transition is possible from $\mathbf{out}_x \otimes \mathbf{box}_y t$.

We shall now obtain a very useful property of derivations. For any derivable transition, it is possible to find a derivation with a specific form, yielding the same transition to an equal residual. The structure of a derivation in this latter form, called the *standard form*, allows all the rules which eliminate term constructors to be applied first. Therefore, for this part of the derivation, each subderivation operates on a strictly smaller term. This allows proof techniques such as structural induction to be used in this part of the derivation. Moreover, all applications of the sychronisation rule occur at the very end of the derivation. This means that the part of the derivation consisting of constructor elimination rules derives labels which do not contain any τ particles. Both these properties will be exploited in the proofs of the main result of this chapter as well as that showing the congruence of bisimilarity, in the next chapter.

Definition 5.9 (Standard derivation) *Let $\mathcal{R}$ be the set of rules given in figures 5–2, 5–3 and 5–4. A derivation obtained by the rules $\mathcal{R}$ is in* standard form (for $\mathcal{R}$) *just when it is constructed in the following manner:*

- *a subderivation consisting of applications of just the constructor elimination rules;*
- *followed by zero or more applications of the permutation rules (*PERM$_1$ *and* PERM$_2$*);*
- *and ending with zero or more applications of the* SYNC *rule.* ■

Lemma 5.10 (Standard derivation) *For any derivable labelled transition $\langle\vec{z}\rangle \vdash t \xrightarrow{\ell} t'$, for some $t'' = t'$ there is a derivation of $\langle\vec{z}\rangle \vdash t \xrightarrow{\ell} t''$ in standard form.*

Proof We show that the permutation rules can be *pushed down* every rule except SYNC and that SYNC can be pushed down every rule. ■

We have now come to the main result of this chapter; that terms which are provably equal in the theory AC$'$ have identical transitions to provably equal residuals.

Lemma 5.11 (Structural) *Whenever $t_1 = t_2$ and $\langle\vec{z}\rangle \vdash t_1 \xrightarrow{\ell} t_1'$ then, for some t_2', $\langle\vec{z}\rangle \vdash t_2 \xrightarrow{\ell} t_2'$ with $t_1' = t_2'$.*

Proof First we shall consider those transitions derived using only the constructor elimination rules i.e. those in which the SYNC and permutation rules do not occur. For each axiom of AC$'$, $t_L = t_R$ we consider the derivable transitions of t_L and t_R under arbitrary environments $\langle\vec{z}\rangle$. We show that whenever there is a derivation of $\langle\vec{z}\rangle \vdash t_L \xrightarrow{\ell} t_L'$ then, for some t_R', there also exists one of $\langle\vec{z}\rangle \vdash t_R \xrightarrow{\ell} t_R'$ with $t_L' = t_R'$ and vice versa.

By the standard derivation lemma, for any derivable $\langle\vec{z}\rangle \vdash t_1 \xrightarrow{\ell} t_1'$, there is a subderivation, for some δ and $t_1'' = t_1'$, of $\langle\vec{z}\rangle \vdash t_1 \xrightarrow{\delta} t_1''$ following which only permutation and SYNC rules are applied. The application of these rules does not depend on the structure of t_1 but only on the labels of the transitions. Moreover, the residual of these rules is obtained by introducing constructions around the residual of the premise which also depend only on the labels. By the above, for some t_2'', of $\langle\vec{z}\rangle \vdash t_2 \xrightarrow{\delta} t_2''$ with $t_1'' = t_2''$. Applying the same sequence of permutation and SYNC rules to this derivation clearly gives a derivation of $\langle\vec{z}\rangle \vdash t_2 \xrightarrow{\ell} t_2'$ for some t_2' which is equal to t_1'.

For the detailed proof the reader is referred to Appendix A.2. ■

τ-transitions and reaction We will now formally establish the relationship between τ-transitions and reaction. To do so, it will be useful to establish first the correspondence between partial reactions in the molecular forms and labelled transitions. One outcome of the structural lemma is that labelled transition relations on the molecular forms can be obtained through factorisation by structural equality. In other words, one can define

$$\langle\vec{z}\rangle \vdash a \xrightarrow{\ell} a' \iff \exists t, t'.\ \langle\vec{z}\rangle \vdash t \xrightarrow{\ell} t' \text{ with } [\![t]\!] = a \text{ and } [\![t']\!] = a'$$

This approach at relating labelled transitions on terms to corresponding ones on the molecular forms does not give any immediate insight regarding the relationship between the structure of a molecular form and the labelled transitions it may perform. Nor does it relate reaction to τ-transitions. It simply assures us that it makes sense to talk about labelled transitions in the world of the molecular forms. In particular, it fails to link our informal explanation of partial reaction on the molecular forms—and the formal one for (complete) single-step reaction given in propsition 5.3—to the labelled transitions. We shall therefore start with a characterisation of simple labelled transitions in terms of the structure of the underlying molecular forms.

Lemma 5.12 *$\langle\vec{z}\rangle \vdash t \xrightarrow{\ell}_= t'$ if and only if $\vdash \langle\vec{z}\rangle \cdot t \xrightarrow{\ell}_= t'$.*

Proof ($\Longrightarrow$) Immediate by applying the composition rule.

($\Longleftarrow$) By standard derivation lemma, for some t'' there is a subderivation of $\vdash \langle\vec{z}\rangle \cdot t \xrightarrow{\delta} t''$ where δ is obtained by replacing each τ in ℓ by some pair of complementary particles in the leftmost position of the label (i.e. a sequence of applications of the SYNC rule suffices to derive $\vdash \langle\vec{z}\rangle \cdot t \xrightarrow{\ell}_= t'$) and $t' = \uparrow_m t''$, with $m \to m$ being the arities of the introduced particles . Then, by inspection of the last *constructor* rule (i.e. composition rule) in the standard derivation of $\vdash \langle\vec{z}\rangle \cdot t \xrightarrow{\ell}_= t'$, we are assured that $\langle\vec{z}\rangle \vdash t \xrightarrow{\delta'}_= t''$ is derivable, where δ' is obtained from δ by the permutations resulting from the permutation rules in the derivation of $\vdash \langle\vec{z}\rangle \cdot t \xrightarrow{\delta} t''$. Then by applying the same sequence of PERM and SYNC rules as in the standard derivation of $\vdash \langle\vec{z}\rangle \cdot t \xrightarrow{\ell}_= t'$ the required transition is derived.■

Proposition 5.13

1. $\langle\vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\langle\vec{v}\rangle}_= t' \iff [\![\langle\vec{z}\rangle \cdot t]\!] = [\vec{\mu}(\vec{u}')]\langle\vec{v}\rangle$ *and* $[\![t']\!] = [\vec{\mu}(\vec{u}')]\langle\vec{u}\rangle$;

2. $\langle\vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\overline{x}\langle\vec{v}\rangle}_= t' \iff [\![\langle\vec{z}\rangle \cdot t]\!] = [\langle x\vec{w}\rangle\mathbf{out}, \vec{\mu}(\vec{u}')]\langle\vec{v}\rangle$ *and* $[\![t']\!] = [\vec{\mu}(\vec{u}')]\langle\vec{w}\vec{u}\rangle$;

3. $\langle\vec{z}\rangle \vdash t \xrightarrow{(\vec{u})x\langle\vec{v}\rangle}_= t' \iff [\![\langle\vec{z}\rangle \cdot t]\!] = [\langle x\rangle\mathbf{box}a(\vec{u}_1), \vec{\mu}(\vec{u}_2)]\langle\vec{v}\rangle$ *and*
$[\![t']\!] = {}^*\{\vec{z}_a/\vec{u}_1\}(\vec{x}_a)[\vec{\lambda}(\vec{y}_a), \vec{\mu}(\vec{u}_2)]\langle\vec{u}\rangle$.

with $a = (\vec{x}_a)\vec{\lambda}(\vec{y}_a)\langle\vec{z}_a\rangle$ *and* $\{\vec{u}\} = \{\vec{u}_1\vec{u}_2\} = \{\vec{u}'\}$.

Proof ($\Longrightarrow$) Induction on the depth of derivation of $\langle\vec{z}\rangle \vdash t \xrightarrow{\ell} t'$.

($\Longleftarrow$) Let the (unique) inverse map of $[\![-]\!]$ be $\widehat{(-)}$. Then, by structural lemma and lemma 5.12, it suffices to give a derivation of $\vdash \widehat{[\![\langle\vec{z}\rangle \cdot t]\!]} \xrightarrow{\ell}_{=} \widehat{[\![t']\!]}$.

For detailed proof the reader is referred to Appendix A.3. ■

Lemma 5.14 $\langle\vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\overline{x}y\langle\vec{v}\rangle}_{=} t'$ *if and only if, for some* $a, \vec{\mu}, \vec{u}_1, \vec{u}_2$*:*

$$\begin{aligned} [\![\langle\vec{z}\rangle \cdot t]\!] &= [\langle x\vec{w}\rangle\mathbf{out}, \langle y\rangle\mathbf{box}a(\vec{u}_1), \vec{\mu}(\vec{u}_2)]\langle\vec{v}\rangle \\ [\![t']\!] &= {}^{*}\{^{\vec{z}_a}/_{\vec{u}_1}\}(\vec{x}_a)[\vec{\lambda}(\vec{y}_a), \vec{\mu}(\vec{u}_2)]\langle\vec{w}\vec{u}\rangle \end{aligned}$$

with $a = (\vec{x}_a)\vec{\lambda}(\vec{y}_a)\langle\vec{z}_a\rangle$ *and* $\{\vec{u}\} = \{\vec{u}_1\vec{u}_2\}$.

Proof ($\Longrightarrow$) By induction on the depth of derivation, we show the stronger result that $\langle\vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\overline{x}y\langle\vec{v}\rangle}_{=} t'$ *or* $\langle\vec{z}\rangle \vdash t \xrightarrow{(\vec{u})y\overline{x}\langle\vec{v}\rangle}_{=} t'$ implies that, for some $a, \vec{\mu}, \vec{u}_1, \vec{u}_2$,

$$\begin{aligned} [\![\langle\vec{z}\rangle \cdot t]\!] &= [\langle x\vec{w}\rangle\mathbf{out}, \langle y\rangle\mathbf{box}a(\vec{u}_1), \vec{\mu}(\vec{u}_2)]\langle\vec{v}\rangle \\ [\![t']\!] &= {}^{*}\{^{\vec{z}_a}/_{\vec{u}_1}\}(\vec{x}_a)[\vec{\lambda}(\vec{y}_a), \vec{\mu}(\vec{u}_2)]\langle\vec{w}\vec{u}\rangle \end{aligned}$$

with $a = (\vec{x}_a)\vec{\lambda}(\vec{y}_a)\langle\vec{z}_a\rangle$ and $\{\vec{u}\} = \{\vec{u}_1\vec{u}_2\}$.

($\Longleftarrow$) By structural lemma and lemma 5.12, it suffices to give a derivation of

$$\vdash \widehat{[\![\langle\vec{z}\rangle \cdot t]\!]} \xrightarrow{\ell}_{=} \widehat{[\![t']\!]}$$

The proof follows similar lines to that for proposition 5.13. ■

The following theorem states that a τ-transition corresponds to a single computational step. The legitimacy of our claim that our operational semantics is computationally meaningful rests mainly upon this fact. While in this thesis no direct characterisation in terms of reaction is given for the bisimulation semantics we obtain in the next chapter, the proposition below serves to establish a preliminary formal connection between reaction-based semantics and labelled transition semantics.

Theorem 5.15

1. *Whenever* $\langle\vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\tau\langle\vec{v}\rangle} t'$ *then* $\langle\vec{z}\rangle \cdot t \searrow^{1} t' \cdot (\vec{u})\langle\vec{v}\rangle$*;*

2. *whenever $t \searrow_{\mathbf{i}}^{1} t'$ then, for any $\vec{z}$, there is some $t'', \vec{u}, \vec{v}$ such that $\langle\vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\tau\langle\vec{v}\rangle} t''$ and $t'' \cdot (\vec{u})\langle\vec{v}\rangle = \langle\vec{z}\rangle \cdot t'$.*

Proof

1. By standard derivation lemma $\langle\vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\overline{x}x\langle\vec{v}\rangle}_{=} t''$ for some t'' such that $t' = \uparrow_m t''$ if the subject arity of the particle pair $\overline{x}x$ is $m \to m$. By lemma 5.14 we have, for some $a, \vec{\mu}, \vec{u}_1, \vec{u}_2$:

$$\begin{aligned} [\![\langle\vec{z}\rangle \cdot t]\!] &= [\langle x\vec{w}\rangle\mathbf{out}, \langle x\rangle\mathbf{box}a(\vec{u}_1), \vec{\mu}(\vec{u}_2)]\langle\vec{v}\rangle \\ [\![t']\!] &= {}^{*}\{\vec{z}_a/\vec{u}_1\}(\vec{x}_a)[\vec{\lambda}(\vec{y}_a), \vec{\mu}(\vec{u}_2)]\langle\vec{w}\vec{u}\rangle \end{aligned}$$

with $a = (\vec{x}_a)\vec{\lambda}(\vec{y}_a)\langle\vec{z}_a\rangle$ and $\{\vec{u}\} = \{\vec{u}_1\vec{u}_2\}$. By alphaconversion, we can assume w.l.o.g. that $\{\vec{w}\} \cap \{\vec{x}_a\vec{y}_a\} = \emptyset$. Also, if $a : m \to n$ then $\vec{w}, \vec{x}_a : m$ and $\vec{u}_1, \vec{z}_a : n$. Now $t' = \uparrow_m t''$. We can write $[\![t'']\!]$ as $\uparrow_n^{\mathcal{M}}(\vec{u}_1\vec{x}_a)[\vec{\lambda}(\vec{y}_a), \vec{\mu}(\vec{u}_2)]\langle\vec{z}_a\vec{w}\vec{u}\rangle$. Hence:

$$\begin{aligned} [\![t' \cdot (\vec{u})\langle\vec{v}\rangle]\!] &= (\uparrow_m\uparrow_n(\vec{u}_1\vec{x}_a)[\vec{\lambda}(\vec{y}_a), \vec{\mu}(\vec{u}_2)]\langle\vec{z}_a\vec{w}\vec{u}\rangle) \cdot (\vec{u})\langle\vec{v}\rangle \\ &= (\uparrow_{n\otimes m}(\vec{u}_1\vec{x}_a)[\vec{\lambda}(\vec{y}_a), \vec{\mu}(\vec{u}_2)]\langle\vec{z}_a\vec{w}\vec{u}\rangle) \cdot (\vec{u})\langle\vec{v}\rangle \\ &= \uparrow_{n\otimes m}(\vec{u}_1\vec{x}_a)[\vec{\lambda}(\vec{y}_a), \vec{\mu}(\vec{u}_2)]\langle\vec{z}_a\vec{w}\vec{v}\rangle \\ &= {}^{*}\{\vec{z}_a\vec{w}/\vec{u}_1\vec{x}_a\}(\vec{u})[\vec{\lambda}(\vec{y}_a), \vec{\mu}(\vec{u}_2)]\langle\vec{v}\rangle \end{aligned}$$

By proposition 5.3 $\langle\vec{z}\rangle \cdot t \searrow_{\mathbf{i}}^{1} t' \cdot (\vec{u})\langle\vec{v}\rangle$.

2. By proposition 5.3 we have

$$[\![t]\!] = (\vec{z}')[\langle x\vec{w}\rangle\mathbf{out}, \langle x\rangle\mathbf{box}a(\vec{u}_1), \vec{\mu}(\vec{u}_2)]\langle\vec{v}\rangle$$
$$[\![t']\!] = {}^{*}\{\vec{z}_a\vec{w}/\vec{u}_1\vec{x}_a\}(\vec{z}')[\vec{\lambda}(\vec{y}_a), \vec{\mu}(\vec{u}_2)]\langle\vec{v}\rangle$$

where $a = (\vec{x}_a)\vec{\lambda}(\vec{y}_a)\langle\vec{z}_a\rangle$. Choose $t'' = \{\vec{z}/\vec{z}'\}\widehat{{}^{*}\{\vec{z}_a\vec{w}/\vec{u}_1\vec{x}_a\}[\vec{\lambda}(\vec{y}_a), \vec{\mu}(\vec{u}_2)]}\langle\vec{u}\rangle$. Writing $[\![t'']\!]$ as $\langle\vec{z}\rangle \cdot \uparrow_{n\otimes m}^{\mathcal{M}}(\vec{u}_1\vec{x}_a\vec{z}')[\vec{\lambda}(\vec{y}_a), \vec{\mu}(\vec{u}_2)]\langle\vec{z}_a\vec{w}\vec{u}\rangle$ it is easy to show that $[\![t'']\!] \cdot (\vec{u})\langle\vec{v}\rangle = \langle\vec{z}\rangle \cdot [\![t']\!]$ and hence that $t'' \cdot (\vec{u})\langle\vec{v}\rangle = \langle\vec{z}\rangle \cdot t'$. By structural lemma and lemma 5.12, it suffices to give a derivation of $\vdash \widehat{[\![\langle\vec{z}\rangle \cdot t]\!]} \xrightarrow{(\vec{u})\tau\langle\vec{v}\rangle}_{=} \widehat{[\![t'']\!]}$. This easily follows by lemma 5.14 and the SYNC rule. ■

Remark As remarked previously, and shown in [29], terms of the (asynchronous) π-calculus are representable as terms of arity $\epsilon \to \epsilon$ in PIC^r. For such terms t, t', the transition $\langle \vec{z} \rangle \vdash t \xrightarrow{(\vec{u})\tau\langle \vec{v} \rangle} t'$ collapses to $\langle\ \rangle \vdash t \xrightarrow{(\ )\tau\langle\ \rangle} t'$ which we can write as $t \xrightarrow{\tau} t'$. Then, by proposition 5.15, we can write

$$t \xrightarrow{\tau}_{=} t' \iff t \searrow^{1} t'$$

which corresponds precisely to our intuition of τ-transitions in the traditional treatment of the π-calculus.

Chapter 6

Bisimilarities

A common method of obtaining an operational semantics for a process calculus is through the notion of bisimilarity on a collection of labelled transition relations. In the previous chapter we defined such a collection; however, we are not obliged to base our definition of bisimilarity on the *entire* collection of labelled transition relations. In this chapter we shall consider a way of obtaining various bisimilarities by choosing different subsets of the collection of labelled transition relations we have defined.

In order to assist us in showing that the bisimilarities we shall define are congruences, a proof technique will be introduced. This technique may have applications beyond our specific setting and so, it shall be presented separately for some unspecified process calculus. For this process calculus, we assume, as given, appropriate notions of process term, context (term with a single hole, or process metavariable) and labelled transition. Let $P, Q, \ldots$ range over *process terms*, $\mathcal{C}$ range over *contexts* and α over labels of labelled transition relations $\xrightarrow{\alpha}$ over process terms. As usual we shall write $P \xrightarrow{\alpha} Q$ for $(P, Q) \in \xrightarrow{\alpha}$ and $\mathcal{C}[P]$ to denote the instantiation of the metavariable in $\mathcal{C}$ by P. The definitions of bisimulation and bisimilarity are standard:

Definition 6.1 *A* bisimulation $\mathcal{S}$ *is a symmetric binary relation on process terms such that, for any* $(P, Q) \in \mathcal{S}$, *whenever* $P \xrightarrow{\alpha} P'$, *then for some* Q', $Q \xrightarrow{\alpha} Q'$ *with* $(P', Q') \in \mathcal{S}$.

Bisimilarity $\sim$ *is the largest bisimulation relation on process terms. Say that* P *and* Q *are* bisimilar *if* $(P, Q) \in \sim$. ∎

It is usually desirable to determine whether $\sim$ is a congruence over the process terms, in other words, if process terms P and Q are bisimilar, then so must be $\mathcal{C}[P]$ and $\mathcal{C}[Q]$, for arbitrary P and Q. This may be done by showing that for each process term constructor C we have, for any $\vec{R}$, $C(P,\vec{R}) \sim C(Q,\vec{R})$. The theory of bisimulation asserts that in order to show that P and Q are bisimilar it suffices to give a bisimulation $\mathcal{S}$ such that $(P,Q) \in \mathcal{S}$. Hence, the proof of congruence may be accomplished by constructing a bisimulation relation containing $(C(P,\vec{R}), C(Q,\vec{R}))$ for each C. This technique is only advantageous if showing $\mathcal{S}$ to be a bisimulation is easier than a more direct proof of the bisimilarity of $C(P,\vec{R})$ and $C(Q,\vec{R})$. However, for certain process calculi it may be difficult to find simple bisimulations which are easily shown to be such. This difficulty may arise, for instance, from a disparity between the (syntactic) structure of the principal and residual terms in the rules for deriving transitions. As an example, consider the composition constructor in PIC′: we would like to determine whether whenever $t_1 \sim t_2$, we also have $t_1 \cdot t \sim t_2 \cdot t$. Assume that $\langle\vec{z}\rangle \vdash t_1 \cdot t \xrightarrow{\ell} s_1$ is derived by the composition rule from premises $\langle\vec{z}\rangle \vdash t_1 \xrightarrow{(\vec{u})\vec{\alpha}(\vec{v})} t_1'$ and $\langle\vec{v}\rangle \vdash t \xrightarrow{(\vec{x})\vec{\beta}(\vec{y})} t'$. Then, for appropriate k_i, l_i and r, we have $s_1 \equiv (t_1' \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}} t') \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r,l_2} \otimes \mathbf{id})$. Clearly, the transition can be matched by $t_2 \cdot t$ to give $s_2 \equiv (t_2' \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}} t') \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r,l_2} \otimes \mathbf{id})$ for some t_2' where $t_1' \sim t_2'$. Hence, in specifying the putative bisimulation relation containing $(t_1 \cdot t, t_2 \cdot t)$ we must ensure that (s_1, s_2) is also present. We can ensure this by specifying closure under abstraction, tensor and composition—but, of course, that involves including almost everything! Alternatively, we note that the terms s_1 and s_2 differ only in the subterms t_1' and t_2' which are in fact bisimilar. Our proof technique takes advantage of this observation.

Lemma 6.2 *Let $\cong$ be some bisimulation equivalence over process terms. Assume that*

> $(*)$ *for any context $\mathcal{C}$ and label α, whenever $P_1 \sim P_2$ and $\mathcal{C}[P_1] \xrightarrow{\alpha} Q_1$, then for some Q_2, $\mathcal{C}[P_2] \xrightarrow{\alpha} Q_2$ and there exist some $\mathcal{C}', P_1', P_2'$ such that $P_1' \sim P_2'$ and $Q_i \cong \mathcal{C}'[P_i']$ (for $i \in \{1,2\}$).*

Then $\sim$ is a congruence.

Proof Consider $\mathcal{S} = \{(Q_1,Q_2) \mid \exists \mathcal{C}, P_1, P_2.\ P_1 \sim P_2, Q_1 \cong \mathcal{C}[P_1], Q_2 \cong \mathcal{C}[P_2]\}$. First we shall show that $\mathcal{S} = \sim$.

$(\sim \subseteq \mathcal{S})$ Consider arbitrary P_1, P_2 such that $P_1 \sim P_2$. Then, taking $\mathcal{C} \equiv [\_]$, it is immediately clear that $(P_1, P_2) \in \mathcal{S}$.

$(\mathcal{S} \subseteq \sim)$ It suffices to show that $\mathcal{S}$ is a bisimulation, since if this is the case then $\sim$ must include $\mathcal{S}$ by definition. Consider an arbitrary $(Q_1, Q_2) \in \mathcal{S}$. Hence, by definition, there exist $\mathcal{C}, P_1, P_2$ such that $P_1 \sim P_2$, $Q_1 \cong \mathcal{C}[P_1]$ and $Q_2 \cong \mathcal{C}[P_2]$. Assume $Q_1 \xrightarrow{\alpha} Q_1'$. Now, since $\cong$ is a bisimulation, $\mathcal{C}[P_1] \xrightarrow{\alpha} Q_1''$ for some $Q_1'' \cong Q_1'$. Also, by assumption (*), we have $\mathcal{C}[P_2] \xrightarrow{\alpha} Q_2''$ and there exist some $\mathcal{C}', P_1', P_2'$ such that $P_1' \sim P_2'$, $Q_1'' \cong \mathcal{C}'[P_1']$ and $Q_2'' \cong \mathcal{C}'[P_2']$. Now, by $Q_2 \cong \mathcal{C}[P_2]$, we get, for some Q_2', $Q_2 \xrightarrow{\alpha} Q_2'$ with $Q_2'' \cong Q_2'$. By transitivity of $\cong$ we have $Q_1' \cong \mathcal{C}[P_1']$ and $Q_2' \cong \mathcal{C}[P_2']$. Hence, by definition, $(Q_1', Q_2') \in \mathcal{S}$. This concludes the demonstration that $\mathcal{S}$ is a bisimulation.

We must now show that $\mathcal{S}$ is a congruence, i.e. it is closed under arbitrary contexts. Assume (P, Q) in $\mathcal{S}$. Then $P \sim Q$ since $\mathcal{S} = \sim$. By reflexivity of $\cong$ we have $(\mathcal{C}[P], \mathcal{C}[Q])$ in $\mathcal{S}$, hence $\mathcal{C}[P] \sim \mathcal{C}[Q]$. ■

Remarks

1. The above technique is useless unless the demonstration of the property (*) is tractable for the process calculus in question. In the case of PIC^r we have been able to prove this property by an induction on the depth of derivation of the labelled transitions. Whether this approach will serve just as well in other process calculi has not been explored.

2. The weakest choice of $\cong$ is bisimilarity itself and the strongest is $\equiv$ (syntactic equality). Often, as in the case of PIC^r, there will be some structural equality which is stronger than $\sim$ but weaker than syntactic equality. This is the one that we shall use for the treatment of bisimilarity in PIC^r.

Outline In Section 6.1 we will examine the bisimilarity obtained by the obvious choice of taking the entire collection of labelled transition relations defined in the previous chapter. This will yield a bisimilarity which is very strong; indeed too strong to give an interesting model. In the next section we will set the scene for obtaining weaker semantics by parameterising bisimilarity by sets of labels; effectively, by sets of labelled transition relations. Several general properties of such parameterised bisimilarities can be obtained. In particular we shall adapt the proof technique described above to the setting of PIC^r. In Section 6.3 we argue that while this technique provides a way of obtaining weaker bisimilarities, it still does not allow (without identifying too much) the identification of certain actions which we expect to be behaviourally indistinguishable. A possible solution is outlined, involving the addition of an extra rule for deriving sequents. In the following section we outline further applications of our technique of specifying bisimilarities by sets of labels to obtain diverse operational models of $\mathsf{PIC}^{\mathsf{r}}$. The final section consists of suggestions for further work.

6.1 Strong Bisimilarity

We shall now define the obvious form of bisimilarity based on the entire collection of labelled transition relations derivable by the rules $\mathcal{R}$.

Definition 6.3 (Strong bisimulation) *A* strong bisimulation *is an indexed set of relations $\mathcal{S} = \{S_{m,n} \mid m, n \in M\}$, where each $S_{m,n}$ is a symmetric binary relation on terms of arity $m \rightarrow n$ and for any $S \in \mathcal{S}$,*

(∗) *given any $t_1 S\, t_2$, environment $\vec{z}$ and a label ℓ, whenever $\langle\vec{z}\rangle \vdash t_1 \xrightarrow{\ell} t_1'$, then for some t_2', $\langle\vec{z}\rangle \vdash t_2 \xrightarrow{\ell} t_2'$ with $t_1' S'\, t_2'$ where $S' \in \mathcal{S}$.*

We shall write $t_1 \mathcal{S}\, t_2$ if $t_1 S\, t_2$ for some $S \in \mathcal{S}$. Strong bisimilarity $\sim$ *is the strong bisimulation where each relation is the largest symmetric binary relation satisfying the property* (∗). ■

The lemma below follows immediately from lemma 5.11.

Lemma 6.4 *Structural equality is a strong bisimulation.*

As the following proposition states, strong bisimilarity and $\equiv_{\mathsf{AC}^{\mathsf{r}}}$ are not identical.

Proposition 6.5 *Strong bisimilarity $\sim$ strictly includes structural equality $\equiv_{\mathsf{AC}^{\mathsf{r}}}$.*

Proof By lemma 6.4 it suffices to show that there is a pair of strongly bisimilar terms which are not provably equal. The following pair has such a property:

$$\uparrow_p(\omega \otimes \mathbf{box}(\nu \cdot (x)\langle xx\rangle))$$
$$\uparrow_p(x)\mathbf{box}\langle xx\rangle$$

To show that they are not provably equal it suffices to consider their molecular forms: the molecular form for the first term has a restriction particle which is absent in that of the second. ■

Discussion It is rather difficult to find pairs of terms which are strongly bisimilar yet not provably equal in AC^r. For instance, even the terms $\boldsymbol{\nu} \cdot \boldsymbol{\omega}$ and $\mathbf{id}_\epsilon$ are distinguished despite being provable in $\mathsf{AC}^{r\epsilon}$. Indeed, we conjecture that, in the version of PIC^r with the constraint that $\mathbf{box}a$ is only well formed when $a : m \to \epsilon$, bisimilarity coincides with structural equality. This may suggest that all the machinery we have introduced is unjustified. However, as we shall see, by limiting the kind of labelled transitions that may be taken into account in comparing actions in terms of their behaviour we shall effectively obtain weaker equivalences.

The labels give a kind of syntactic description of the "dynamic interface" of an action. Unfortunately, some labels do not really reflect any potential for interaction. Consider, for instance the terms $(\nu x)(\langle xv \rangle \cdot \mathbf{out})$ and $\mathbf{id}_\epsilon$. Neither of these terms can ever interact with any other action either through the provision of names or through the contribution of molecules for reaction. Hence we would like a semantics which identifies them. The term $(\nu x)(\langle xv \rangle \cdot \mathbf{out})$ can have the labelled transition

$$\vdash (\nu x)(\langle xv \rangle \cdot \mathbf{out}) \xrightarrow{(x)\overline{x}\langle\ \rangle}_{=} \langle v \rangle$$

while the only one for $\mathbf{id}_\epsilon$ is $\vdash \mathbf{id}_\epsilon \xrightarrow{(\ )\langle\ \rangle} \mathbf{id}_\epsilon$. Clearly, these two are not strongly bisimilar.

Inspection of the label $(x)\overline{x}\langle\ \rangle$ reveals that there is no context which will furnish the required particle x, since the name x is rendered private by the binding. Nor is the private name exported and hence it can never be present in an external action. This suggests that such labels should be disregarded in the definition of bisimilarity.

6.2 Parameterising Bisimilarity

We shall now examine a way by which weaker forms of bisimilarity may be obtained, motivated by the reasons given above. The method we shall adopt involves restricting consideration to a subclass of the labelled transition relations in determining whether two actions (or terms) are bisimilar. A similar approach was taken by Milner in [21] through the notion of *incident sets*. In [21], the choice of the subset of labels (and consequently, labelled transition relations)—the incidents—was not arbitrary but was subject to certain conditions. Here, we shall impose no such conditions a priori; although, in our examples, the choice of labels will in each case be defined through some structural property of the labels.

Definition 6.6 (Strong Λ-bisimulation) *Let Λ be a set of labels in $\mathcal{L}$. A* strong Λ-bisimulation *is an indexed set of relations $\mathcal{S} = \{S_{m,n} \mid m, n \in M\}$, where each $S_{m,n}$ is a symmetric binary relation on terms of arity $m \to n$ and for any $S \in \mathcal{S}$,*

(*) *given any $t_1 S\, t_2$ and a label $\ell \in \Lambda$, whenever $\langle\vec{z}\rangle \vdash t_1 \xrightarrow{\ell} t_1'$, then for some t_2', $\langle\vec{z}\rangle \vdash t_2 \xrightarrow{\ell} t_2'$ with $t_1' S'\, t_2'$ where $S' \in \mathcal{S}$.*

We shall write $t_1 \mathcal{S}\, t_2$ if $t_1 S\, t_2$ for some $S \in \mathcal{S}$. Strong Λ-bisimilarity $\overset{\Lambda}{\sim}$ *is the strong bisimulation where each relation is the largest symmetric binary relation satisfying the property (*).* ■

The following two simple lemmas hold for strong Λ bisimulation, given any $\Lambda \subseteq \mathcal{L}$.

Lemma 6.7 *For any $\Lambda' \subseteq \Lambda$, if $\mathcal{S}$ is a strong Λ-bisimulation, then it is also a strong Λ'-bisimulation.*

Proof Assume $a_1 \mathcal{S} a_2$ and consider the transition $\langle\vec{z}\rangle \vdash a_1 \xrightarrow{\ell} a_1'$ for an arbitrary $\ell \in \Lambda'$. Now, since $\ell \in \Lambda$ and $\mathcal{S}$ is a strong Λ-bisimulation, we have $\langle\vec{z}\rangle \vdash a_2 \xrightarrow{\ell} a_2'$ for some a_2' such that $a_1' \mathcal{S} a_2'$. Hence result. ■

This immediately gives the following result:

Corollary 6.8 *For any Λ, structural equality $\equiv_{\mathsf{AC}'}$ is a strong Λ-bisimulation.*

Proof Immediate, by lemma 5.11 and lemma 6.7. ■

Definition 6.9 (Contexts) *A* context *in PIC' is a term with a single hole (metavariable) $[\_]$, generated as follows:*

$$C \;::=\; [\_] \mid t \otimes C \mid C \otimes t \mid t \cdot C \mid C \cdot t \mid \mathbf{ab}_x C \mid \uparrow C \mid \mathbf{box} C$$

We write $C[t]$ to mean the replacement of the hole occurring in C by t. ■

Lemma 6.10 *Assume that for any context C and label $\ell \in \Lambda$, whenever $t_1 \overset{\Lambda}{\sim} t_2$ and $\langle\vec{z}\rangle \vdash C[t_1] \xrightarrow{\ell} s_1$, then for some s_2, $\langle\vec{z}\rangle \vdash C[t_2] \xrightarrow{\ell} s_2$ and, there exist some C', t_1', t_2' where $t_1' \sim t_2'$ and $s_i = C'[t_i']$ (for $i \in \{1,2\}$). Then $\overset{\Lambda}{\sim}$ is a congruence.*

Proof The proof involves a straightforward application of the technique introduced at the start of this chapter. By lemma 6.2, it suffices to show that equality $=$ is a strong Λ-bisimulation. Hence, by corollary 6.8 the result follows. ■

We will now use the above lemma to show that strong bisimilarity $\sim$ is a congruence:

Lemma 6.11 *Let $t_1 \sim t_2$. Then, for any context C and label ℓ, whenever $\langle \vec{z} \rangle \vdash C[t_1] \xrightarrow{\ell} s_1$, then for some s_2, $\langle \vec{z} \rangle \vdash C[t_2] \xrightarrow{\ell} s_2$ and, there exist some C', t_1', t_2' where $t_1' \sim t_2'$ and $s_i = C'[t_i']$ (for $i \in \{1,2\}$).*

Proof Whenever $C \equiv [\_]$ the result follows by definition of bisimulation. For $C \not\equiv [\_]$ the result is obtained by induction on the depth of derivation of $\langle \vec{z} \rangle \vdash C[t_1] \xrightarrow{\ell} s_1$. ■

Theorem 6.12 *Strong bisimilarity $\sim$ is a congruence on the terms of* PIC^r.

Proof Immediate by lemma 6.10 and lemma 6.11. ■

6.3 Discarding Redundant Bindings

While the technique described in the previous section allows a great variety of bisimilarities to be obtained, the fineness with which the strength (or weakness) of the resulting model can be controlled is limited by the available labelled transition relations. In other words, there may be terms which cannot be identified by any model thus obtained without resulting in other identifications, possibly undesirable, being made. In this section we shall give an example of such a circumstance together with a simple solution for changing the set of available labelled transitions which, in addition to the technique described in Section 6.2, allows us to obtain an interesting model.

Consider the transitions in figure 6–1; the transitions are exhaustive for the terms shown. We would not like to distinguish between any of the terms in each pair on behavioural grounds; yet, it is clear that they do not derive the same transitions. The difference between the labels in each case is also easy to discern: for one of the terms the label has an extra binding occurrence and significantly, this extra name does not bind anything in the label.

$$\vdash \boldsymbol{\nu} \cdot \boldsymbol{\omega} \xrightarrow{(x)\langle\,\rangle}_{=} \boldsymbol{\nu}$$
$$\vdash \mathbf{id}_\epsilon \xrightarrow{(\,)\langle\,\rangle}_{=} \mathbf{id}_\epsilon$$

$$\vdash (\boldsymbol{\nu} x)\mathbf{box}_y\langle x\rangle \xrightarrow{(xu)\langle u\rangle}_{=} \boldsymbol{\nu} \qquad (y \notin \{x, u\})$$
$$\vdash \mathbf{box}_y \boldsymbol{\nu} \xrightarrow{(u)\langle u\rangle}_{=} \boldsymbol{\nu} \qquad (y \neq u)$$

$$\vdash (\boldsymbol{\nu} x)\mathbf{box}_y\langle x\rangle \xrightarrow{(xu)y\langle u\rangle}_{=} \boldsymbol{\nu} \qquad (y \notin \{x, u\})$$
$$\vdash \mathbf{box}_y \boldsymbol{\nu} \xrightarrow{(u)y\langle u\rangle}_{=} \boldsymbol{\nu} \qquad (y \neq u)$$

$$\vdash (\mathbf{box}_y\langle x\rangle) \cdot \boldsymbol{\omega} \xrightarrow{(u)\langle\,\rangle}_{=} \boldsymbol{\nu} \qquad (y \neq u)$$
$$\vdash \mathbf{box}_y \mathbf{id}_\epsilon \xrightarrow{(\,)\langle\,\rangle}_{=} \mathbf{id}_\epsilon$$

$$\vdash (\mathbf{box}_y\langle x\rangle) \cdot \boldsymbol{\omega} \xrightarrow{(u)y\langle\,\rangle}{}'_{=} \boldsymbol{\nu} \qquad (y \neq u)$$
$$\vdash \mathbf{box}_y \mathbf{id}_\epsilon \xrightarrow{(\,)y\langle\,\rangle}{}'_{=} \mathbf{id}_\epsilon$$

Figure 6–1: Distinctions caused by redundant bindings

Prompted by the technique described in section 2, we could try to obtain an appropriate model via the bisimilarity induced by just those labels in which such redundant bindings do not occur. However, on its own this measure will not result in a weaker bisimilarity. This is because the vector of binding occurrences in labels is predetermined (up to permutation) by the structure of the molecular form of the term undergoing the transition (see proposition 5.13). Inspection of these propositions reveals that, for any given term, in each of its transitions the vector of bindings of the label is some permutation of the binding occurrences originating from the controls (including restriction particles) present in its molecular form.

This means that, for any non-empty set of labels Λ, the resulting Λ-bisimilarity will distinguish some terms, such as those of figure 6–1, which are distinguishable (by bisimilarity) solely upon the difference in the mentioned binding vectors. To see why, take any term t with a labelled transition whose label is in Λ. Then $t \otimes (\boldsymbol{\nu} \cdot \boldsymbol{\omega})$ will be distinguished from t (although behaviourally we do not expect the distinction) since for *any* labelled transition of $t \otimes (\boldsymbol{\nu} \cdot \boldsymbol{\omega})$, the label will differ from that for t in the binding vector.

In order to rectify this, we shall introduce a new rule DISCARD which allows redundant binding occurrences in labels to be discarded. This will break the uniqueness of binding vectors for each given term and will in fact allow us to obtain the required form of bisimilarity. The DISCARD rule simply takes a redundant binding occurrence from the label and places it at the export of the residual. This is accomplished by deleting the occurrence and postcomposing with the residual a discard operation ($\boldsymbol{\omega}$) in the appropriate place. We will show that when this rule is added to the other rules $\mathcal{R}$ we will still be able to obtain the relevant counterpart of the structural lemma. Unless otherwise stated we shall henceforth use the notation $\langle\vec{z}\rangle \vdash t \xrightarrow{\ell} t'$ to denote a transition which is derivable by the rules $\mathcal{R}$ together with DISCARD. As before, we shall assume that $\vec{\alpha} : k \to l$ in the rule below:

$$\text{DISCARD}\;\frac{\langle\vec{z}\rangle \vdash t \xrightarrow{(x\vec{u})\vec{\alpha}\langle\vec{v}\rangle} t'}{\langle\vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\vec{\alpha}\langle\vec{v}\rangle} t' \cdot (\mathbf{id}_l \otimes \boldsymbol{\omega} \otimes \mathbf{id})}\; x \notin \mathbf{fn}(\vec{\alpha}) \cup \{\vec{v}\}$$

Definition 6.13 (Standard derivation) *Let $\mathcal{R}_D$ be the set $\mathcal{R}$ together with the* DISCARD *rule. Then a derivation is in* standard form for $\mathcal{R}_D$ *just when it consists in a subderivation which is in standard form for $\mathcal{R}$ followed by zero or more applications of the* DISCARD *rule.* ∎

Lemma 6.14 (Standard derivation) *For any labelled transition $\langle\vec{z}\rangle \vdash t \xrightarrow{\ell} t'$ derivable by the rules $\mathcal{R}_D$, there exists some t'' such that $t'' = t'$ for which there is a derivation of $\langle\vec{z}\rangle \vdash t \xrightarrow{\ell} t''$ in standard form (for $\mathcal{R}_D$).*

Proof We show that the DISCARD rule can be pushed down every rule. Hence there is a derivation consisting of a subderivation not containing an applications of the DISCARD rule followed by some number of applications of the DISCARD rule. By lemma 5.10 this subderivation can be replaced by a subderivation which is in standard form for $\mathcal{R}$. ∎

Lemma 6.15 (Structural) *Whenever $t_1 = t_2$ and $\langle\vec{z}\rangle \vdash t_1 \xrightarrow{\ell} t_1'$ then, for some t_2', $\langle\vec{z}\rangle \vdash t_2 \xrightarrow{\ell} t_2'$ with $t_1' = t_2'$.*

Proof By the standard derivation lemma, for any derivable $\langle\vec{z}\rangle \vdash t_1 \xrightarrow{\ell} t_1'$, there is a subderivation, for some δ and $t_1'' = t_1'$, of $\langle\vec{z}\rangle \vdash t_1 \xrightarrow{\ell'} t_1''$ following which only the DISCARD rule is applied. The application of this rule does not depend on the structure of t_1 but only on the labels of the transitions. Moreover, the residual is obtained by introducing contructions around the residual of the premise which depend only on the label of the premise transition. By lemma 5.11, for some t_2'', $\langle\vec{z}\rangle \vdash t_2 \xrightarrow{\ell'} t_2''$ with $t_1'' = t_2''$. Applying the same sequence of DISCARD rules to this derivation clearly gives a derivation of $\langle\vec{z}\rangle \vdash t_2 \xrightarrow{\ell} t_2'$ for some t_2' which is equal to t_1'. ∎

Definition 6.16 *A label $(\vec{u})\vec{\alpha}\langle\vec{v}\rangle$* has redundant bindings *if there is some $x \in \{\vec{u}\}$ which occurs neither in $\vec{\alpha}$ nor in $\vec{v}$.* ∎

We shall now consider bisimulation on transitions whose labels do not contain redundant bindings. We shall henceforth let $\mathcal{L}_b$ stand for the set of labels with no redundant bindings, i.e. those labels $\ell = (\vec{u})\vec{\alpha}\langle\vec{v}\rangle$ where $\{\vec{u}\} \subseteq \mathbf{fn}(\vec{\alpha}) \cup \{\vec{v}\}$.

Lemma 6.17 *Structural equality $\equiv_{\mathsf{AC}'}$ is a strong $\mathcal{L}_b$-bisimulation.*

Proof Immediate, by lemma 6.15. ∎

We shall now show that strong $\mathcal{L}_b$-bisimilarity is a congruence. We note that the proof of lemma 6.10 depends on the set of rules used for deriving the sequents only insofar as structural congruence is a bisimilarity. Since adding the DISCARD rule preserves this property of structural congruence (for arbitrary sets of labels Λ) we can use the same technique.

However, we cannot use a straightforward induction on the depth of derivation of $\langle\vec{z}\rangle \vdash C[t_1] \xrightarrow{\ell} s_1$ to get the required result as stated in corollary 6.20, since in the case of the DISCARD rule, we would not be able to apply the inductive hypothesis to its premise (in which the label has at least one redundant binding and therefore is not in $\mathcal{L}_b$).

Notation Let $\ell = (\vec{u})\vec{\alpha}\langle\vec{v}\rangle$. Then we shall write $\widehat{\ell}$ to denote the label obtained by discarding all binding occurrences in ℓ which do not bind any name (in ℓ). Hence, $\widehat{\ell} = (\vec{u}')\vec{\alpha}\langle\vec{v}\rangle$ where,

1. $\vec{u}' = \vec{u}_1\vec{u}_2\cdots\vec{u}_{n+1}$;
2. $\vec{u} = \vec{u}_1 w_1 \cdots \vec{u}_n w_n \vec{u}_{n+1}$;
3. $\{\vec{u}'\} \subseteq \mathbf{fn}(\vec{\alpha}) \cup \{\vec{v}\}$;
4. $\{w_1, \ldots, w_n\} \cap (\mathbf{fn}(\vec{\alpha}) \cup \{\vec{v}\}) = \emptyset$

In other words the binding occurrences w_i are redundant in ℓ while the binding occurrences $\vec{u}'$ are not.

Lemma 6.18 *Whenever* $\langle\vec{z}\rangle \vdash t \xrightarrow{\ell} t'$, *then*

$$\langle\vec{z}\rangle \vdash t \xrightarrow{\widehat{\ell}}_{=} t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle u'\rangle)$$

where $\ell = (\vec{u})\vec{\alpha}\langle\vec{v}\rangle$ *and* $\widehat{\ell} = (\vec{u}')\vec{\alpha}\langle\vec{v}\rangle$.

Proof Let $\vec{u} = \vec{u}_1 w_1 \cdots \vec{u}_n w_n \vec{u}_{n+1}$ and $\vec{u}' = \vec{u}_1\vec{u}_2\cdots\vec{u}_{n+1}$, i.e. $w_1 \cdots w_n$ redundant. We proceed by induction on n.

Base Case: $n = 0$ Immediate.

Inductive Step: $n = j + 1$ Assume $\langle\vec{z}\rangle \vdash t \xrightarrow{\ell} t'$. Then, by applying PERM1 to pull the name w_{j+1} in the leftmost position, we get $\langle\vec{z}\rangle \vdash t \xrightarrow{\ell'} t' \cdot (\mathbf{id}_l \otimes \mathbf{p}_{m,p} \otimes \mathbf{id})$ where $\ell' = (w_{j+1}\vec{u}_1 w_1 \cdots w_j \vec{u}_{j+1}\vec{u}_{j+2})\vec{\alpha}\langle\vec{v}\rangle$ with $u_1 w_1 \cdots w_j \vec{u}_{j+1} : m$ and $w_{j+1} : p$.

Applying the DISCARD rule to remove the redundant binding occurrence of w_{j+2} we are left with the transition $\langle\vec{z}\rangle \vdash t \xrightarrow{\ell''} t' \cdot (\mathbf{id}_l \otimes \mathbf{p}_{m,p} \otimes \mathbf{id}) \cdot (\mathbf{id}_l \otimes \omega \otimes \mathbf{id})$, with $\ell'' = (\vec{u}_1 w_1 \cdots w_j \vec{u}_{j+1}\vec{u}_{j+2})\vec{\alpha}\langle\vec{v}\rangle$. We can now apply the inductive hypothesis, getting,

$$\langle\vec{z}\rangle \vdash t \xrightarrow{\widehat{\ell''}} t''$$

where $t'' = t' \cdot (\mathbf{id}_l \otimes \mathbf{p}_{m,p} \otimes \mathbf{id}) \cdot (\mathbf{id}_l \otimes \omega \otimes \mathbf{id}) \cdot (\mathbf{id}_l \otimes (\vec{u}_1 w_1 \cdots w_j \vec{u}_{j+1}\vec{u}_{j+2})\langle\vec{u}'\rangle)$.

But $\widehat{l''} = \widehat{l}$; and

$(\mathbf{id}_l \otimes \mathbf{p}_{m,p} \otimes \mathbf{id}) \cdot (\mathbf{id}_l \otimes \omega \otimes \mathbf{id}) = \mathbf{id}_l \otimes (\vec{u}_1 w_1 \cdots w_j \vec{u}_{j+1} w_{j+1} \vec{u}_{j+2})\langle\vec{u}_1 w_1 \cdots w_j \vec{u}_{j+1}\vec{u}_{j+2}\rangle)$.

Hence we have $t'' = t'_1 \cdot (\mathbf{id}_l \otimes (\vec{u}_1 w_1 \cdots w_j \vec{u}_{j+1} w_{j+1} \vec{u}_{j+2})\langle\vec{u}'\rangle)$. ■

Lemma 6.19 *Let $t_1 \overset{\mathcal{L}_b}{\sim} t_2$. Then, for any context C, whenever $\langle\vec{z}\rangle \vdash C[t_1] \xrightarrow{\ell} s_1$, then for some s_2, $\langle\vec{z}\rangle \vdash C[t_2] \xrightarrow{\widehat{\ell}} s_2$ and, there exist some C', t'_1, t'_2 where $t'_1 \overset{\mathcal{L}_b}{\sim} t'_2$, $C'[t'_1] = s_1 \cdot (\mathbf{id}_l \otimes (\vec{u})\langle\vec{u}'\rangle)$ and $C'[t'_2] = s_2$; where $\ell = (\vec{u})\vec{\alpha}\langle\vec{v}\rangle$ and $\widehat{\ell} = (\vec{u}')\vec{\alpha}\langle\vec{v}\rangle$.*

Proof Assume $C \equiv [\_]$. Let $\langle\vec{z}\rangle \vdash t_1 \xrightarrow{\ell} s_1$. Then, by lemma 6.18, $\langle\vec{z}\rangle \vdash t_1 \xrightarrow{\widehat{\ell}}_= s_1 \cdot (\mathbf{id}_l \otimes (\vec{u})\langle u'\rangle)$. By definition of $\mathcal{L}_b$-bisimilarity, $\langle\vec{z}\rangle \vdash t_1 \xrightarrow{\widehat{\ell}} s_2$ such that, by lemma 6.17 the transitivity of bisimilarity $s_1 \cdot (\mathbf{id}_l \otimes (\vec{u})\langle u'\rangle) \overset{\mathcal{L}_b}{\sim} s_2$.

Assume $C \not\equiv [\_]$. We proceed by induction on the depth of derivation of $\langle\vec{z}\rangle \vdash C[t_1] \xrightarrow{\ell} s_1$. ■

Corollary 6.20 *Let $t_1 \overset{\mathcal{L}_b}{\sim} t_2$. Then, for any context C and label $\ell \in \mathcal{L}_b$, whenever $\langle\vec{z}\rangle \vdash C[t_1] \xrightarrow{\ell} s_1$, then for some s_2, $\langle\vec{z}\rangle \vdash C[t_2] \xrightarrow{\ell} s_2$ and, there exist some C', t'_1, t'_2 where $t'_1 \overset{\mathcal{L}_b}{\sim} t'_2$ and $s'_i = C'[t'_i]$ (for $i \in \{1, 2\}$).*

Proof By lemma 6.19, since for any label ℓ with no redundant bindings, $\widehat{\ell} = \ell$. ■

Theorem 6.21 *Strong bisimilarity $\overset{\mathcal{L}_b}{\sim}$ is a congruence on the terms of* PIC^r.

Proof The proof follows that of lemma 6.10, which cannot be applied here as it was shown in the context of the rules $\mathcal{R}$ and not $\mathcal{R}_D$.

Consider $\mathcal{S} = \{(s_1, s_2) \mid \exists C, t_1, t_2.\ t_1 \overset{\mathcal{L}_b}{\sim} t_2, s_1 = C[t_1], s_2 = C[t_2]\}$. Clearly, $\mathcal{S}$ contains $\overset{\mathcal{L}_b}{\sim}$ (choosing $C \equiv [\ ]$) and is closed under contexts. Therefore, if we show that $\mathcal{S}$ is a $\mathcal{L}_b$-bisimulation then we are done since that would imply that $\mathcal{S} = \overset{\mathcal{L}_b}{\sim}$.

Consider an arbitrary $(s_1, s_2) \in \mathcal{S}$. Assume $\langle\vec{z}\rangle \vdash s_1 \xrightarrow{\ell} s'_1$, where $\ell \in \Lambda$. Since $(s_1, s_2) \in \mathcal{S}$ there exist C, t_1, t_2 such that $t_1 \overset{\mathcal{L}_b}{\sim} t_2$, $s_1 = C[t_1]$ and $s_2 = C[t_2]$. By the structural lemma $\langle\vec{z}\rangle \vdash C[t_1]\xrightarrow{\ell}_{=}s_1'$ and by corollary 6.20, we have $\langle\vec{z}\rangle \vdash C[t_2] \xrightarrow{\ell} s_2'$ and, for some, C', t_1', t_2' such that $t_1' \overset{\mathcal{L}_b}{\approx} t_2'$, $s_1' = C'[t_1']$ and $s_2' = C'[t_2']$. Hence, since $s_2 = C[t_2]$, we have $\langle\vec{z}\rangle \vdash s_2\xrightarrow{\ell}_{=}s_2'$ with $(s_1', s_2') \in \mathcal{S}$. ■

Examples The following are some examples of terms which are not provably equal in AC′ but are bisimilar.

$$\begin{array}{rcl} \boldsymbol{\nu}\cdot\mathbf{id} & \overset{\mathcal{L}_b}{\approx} & \mathbf{id}_\epsilon \\ (\mathbf{box}_x a)\cdot\boldsymbol{\omega} & \overset{\mathcal{L}_b}{\approx} & \mathbf{box}_x(a\cdot\boldsymbol{\omega}) \\ (\nu y)\mathbf{box}_x a & \overset{\mathcal{L}_b}{\approx} & \mathbf{box}_x(\nu y)a \quad (x \neq y) \end{array}$$

Discussion We may consider adding the axiom ρ_0 (which holds in the model obtained above), to the equations on terms defining structural equality, giving us $\mathsf{PIC}^{r\epsilon}$: the reflexive π-calculus with garbage collection. In this setting, the structural lemma (for $\equiv_{\mathsf{AC}^{r\epsilon}}$) fails. This is illustrated by the equation $\boldsymbol{\nu}\cdot\boldsymbol{\omega} = \mathbf{id}_\epsilon$ provable in $\mathsf{AC}^{r\epsilon}$, where the transition $\vdash \boldsymbol{\nu}\cdot\boldsymbol{\omega} \xrightarrow{(x)\langle\,\rangle} \boldsymbol{\nu}$ cannot be matched by $\mathbf{id}_\epsilon$. However, such transitions should hardly matter since we have decided to ignore them in our semantics. Instead, it should be possible to show the weaker result that $\equiv_{\mathsf{AC}^{r\epsilon}}$ is a strong $\mathcal{L}_b$-bisimulation.

6.4 Other models

There are several interesting semantics which can be defined in terms of sets of labels. While it remains to be checked whether the bisimilarities concerned are congruences, the following examples illustrate some computationally meaningful choices for the mentioned sets.

Non-interleaving semantics At the end of section 6.1, it was suggested that for any labelled transition $\langle\vec{z}\rangle \vdash t \xrightarrow{\ell} t'$, no context applied to t can provide complementary particles to those particles whose names are bound in ℓ but not exported. Such labels were at least partially responsible for the distinction between terms which we expect to be identified in an operational model. We can develop a semantics based on those transition relations whose labels do not contain such particles.

Definition 6.22 (Active labels) *A label* $(\vec{u})\vec{\alpha}\langle\vec{v}\rangle$ *is said to be* active *if,*

1. *it has no redundant bindings; and,*

2. *for any particle in $\vec{\alpha}$ whenever its subject name is bound (occurs in $\vec{u}$), then the same name is also exported (occurs in $\vec{v}$).*

The set of active labels is denoted by $\mathcal{L}_a$.

Examples Below are some pairs of $\mathcal{L}_a$-bisimilar terms:

$$\begin{array}{rcl} \boldsymbol{\nu}\cdot\boldsymbol{\omega} & \stackrel{\mathcal{L}_a}{\sim} & \mathbf{id}_\epsilon \\ (\boldsymbol{\nu}x)(\mathbf{out}_x) & \stackrel{\mathcal{L}_a}{\sim} & \boldsymbol{\omega}\otimes\cdots\otimes\boldsymbol{\omega} \\ (\boldsymbol{\nu}x)(\mathbf{box}_x a) & \stackrel{\mathcal{L}_a}{\sim} & \boldsymbol{\nu}\otimes\cdots\otimes\boldsymbol{\nu} \\ (\boldsymbol{\nu}y)\mathbf{box}_x a & \stackrel{\mathcal{L}_a}{\sim} & \mathbf{box}_x(\boldsymbol{\nu}y)a \quad (x\neq y) \end{array}$$

Interleaving semantics The bisimilarities described so far have the common feature that they all give a non-interleaving semantics. We shall weaken the semantics further by basing bisimilarity on the set of just those active labels at most one particle in their bodies. This will give a weaker (strong) bisimilarity that $\stackrel{\mathcal{L}_a}{\sim}$. We let $\mathcal{L}_i = \{\ell\in\mathcal{L}_a \mid \ell = (\vec{u})\vec{\alpha}\langle\vec{v}\rangle, |\vec{\alpha}|\leq 1\}$.

Examples The following pairs of terms are $\mathcal{L}_i$-bisimilar:

$$\mathbf{box}_x(\mathbf{box}_x\mathbf{id}_\epsilon) \stackrel{\mathcal{L}_i}{\sim} \mathbf{box}_x\mathbf{id}_\epsilon\otimes\mathbf{box}_x\mathbf{id}_\epsilon$$

$$(\boldsymbol{\nu}xy)(\mathbf{out}_x\otimes\mathbf{box}_x(\mathbf{out}_y\otimes\mathbf{box}_y\mathbf{id}_\epsilon)) \stackrel{\mathcal{L}_i}{\sim} (\boldsymbol{\nu}xy)(\mathbf{out}_x\otimes\mathbf{box}_x\mathbf{id}_\epsilon\otimes\mathbf{out}_y\otimes\mathbf{box}_y\mathbf{id}_\epsilon)$$

Restriction skeleta revisited We conjecture that νSKEL can be obtained by a suitable choice of labels. Let $\mathcal{L}_s = \{(\vec{u})\langle\vec{v}\rangle \mid \vec{u}\subseteq\vec{v}\}$. Note that $\mathcal{L}_s$ is a subset of all the sets of labels considered so far, hence resulting in the weakest model. Indeed, factoring the terms of PIC^r by strong $\mathcal{L}_s$-bisimilarity $\stackrel{\mathcal{L}_s}{\sim}$ should give (a reflexive control structure isomorphic to) νSKEL.

6.5 The Asynchronous π-calculus

Throughout this thesis we have informally referred to a correspondence between PIC′ and the π-calculus; therefore, a natural task would be to make this correspondence precise. This may be achieved by, first, defining a translation from the terms of the asynchronous π-calculus to those of PIC′ followed by an comparison between the manifestations of labelled transition relations and strong bisimulation in both calculi. We shall now briefly illustrate what this involves, confining ourselves to the monadic version mainly for simplicity of exposition.

The terms of the asynchonous π-calculus $\mathcal{P}$ essentially correspond to the fragment of the full π-calculus, or more closely, to the ν-calculus of Honda and Yoshida in [11,10].

$$P ::= \mathbf{0} \mid \overline{x}\langle v\rangle \mid x(y).P \mid (\nu x)P \mid P|Q$$

To obtain processes, the terms of $\mathcal{P}$ are factored by a structural congruence $\equiv$ induced by the following equations:

$$\begin{array}{lll} P|\mathbf{0} \equiv P & (\nu x)(\nu y)P \equiv (\nu y)(\nu x)P & \\ P|Q \equiv Q|P & (\nu x)(P|Q) \equiv P|(\nu x)Q & (x \notin \mathbf{fn}\,(P)) \\ P|(Q|R) \equiv (P|Q)|R & (\nu x)P \equiv (\nu y)(\{y/x\}P) & (y \notin \mathbf{fn}\,(P)) \\ & z(x).P \equiv z(y).(\{y/x\}P) & (y \notin \mathbf{fn}\,(P)) \end{array}$$

where $\mathbf{fn}\,(P)$ denotes the *free names* of P, with the occurrence of any name x in P being free unless bound in some subterm Q of P, by a $(\nu x)Q$ or $z(x).Q$ construct, whose scope extends throughout the subterm Q.

The dynamics are given in terms of reduction $\rightarrow$ the smallest relation over $\mathcal{P}$ closed under $\equiv$ and the following rules:

$$\text{COM}: \overline{x}\langle z\rangle|x(y).P \rightarrow \{z/y\}P$$

$$\text{PAR}\frac{P \rightarrow P'}{P|Q \rightarrow P'|Q} \qquad \text{RES}\frac{P \rightarrow P'}{(\nu x)P \rightarrow (\nu x)P'}$$

In [29], Milner has shown the correspondence between the processes in $\mathcal{P}$ and PIC. The translation to PIC^r is identical:

$$\begin{aligned}
\widehat{0} &\stackrel{\text{def}}{=} \mathbf{id}_\epsilon \\
\widehat{\bar{x}\langle v\rangle} &\stackrel{\text{def}}{=} \langle v\rangle \cdot \mathbf{out}_x \\
\widehat{x(y).P} &\stackrel{\text{def}}{=} \mathbf{box}_x(y)\widehat{P} \\
\widehat{(\nu x)P} &\stackrel{\text{def}}{=} (\nu x)\widehat{P} \\
\widehat{P|Q} &\stackrel{\text{def}}{=} \widehat{P} \otimes \widehat{Q}
\end{aligned}$$

The terms of $\mathcal{P}$ translate to PIC^r terms of arity $\epsilon \to \epsilon$. Then, from [29], we have

(1) $P \equiv Q$ if and only if $\widehat{P} = \widehat{Q}$.

(2) If $P \to Q$ then $\widehat{P} \searrow^1 \widehat{Q}$.

(3) If $\widehat{P} \searrow^1 t$ then for some P', $P \to P'$ and $\widehat{P'} = t$.

Labelled Transitions In figure 6–3 we give the derivation rules for transitions terms in $\mathcal{P}$. The rules allow the derivation of *early* transitions allowing a precise correspondence between labelled transitions in $\mathcal{P}$ and PIC^r to be stated.

The relationship expected between τ transitions in PIC^r and reductions in $\mathcal{P}$ is fairly easy to establish. It may be obtained through the intermediate relationship of both relations with single-step reaction. Recall that theorem 5.15 states that, for actions $\widehat{P}, \widehat{P'}$ of arities $\epsilon \to \epsilon$:

$$\widehat{P} \xrightarrow{\tau}_= \widehat{P'} \iff \widehat{P} \searrow^1 \widehat{P'}$$

This, together with above relationship between reaction and labelled transition relations gives:

$$P \to Q \iff \widehat{P} \xrightarrow{\tau}_= \widehat{Q}$$

However, we still do not have any information about the relationship between labelled transition relations; and more importantly, between the models of each given by bisimilarity. In particular we expect the following to hold:

$$P \xrightarrow{\tau} P' \iff \vdash \widehat{P} \xrightarrow{\tau}_= \widehat{P'}$$

$$\text{OUT}\ \frac{}{\overline{x}\langle w\rangle \xrightarrow{\overline{x}\langle w\rangle} \mathbf{0}}$$

$$\text{IN}\ \frac{}{x(y).P \xrightarrow{x\langle w\rangle} \{{}^{w}\!/\!{}_{y}\}P}$$

$$\text{RES}\ \frac{P \xrightarrow{\alpha} P'}{(\nu x)P \xrightarrow{\alpha} (\nu x)P'}\ x \notin \mathrm{n}(\alpha)$$

$$\text{OPEN}\ \frac{P \xrightarrow{\overline{x}\langle y\rangle} P'}{(\nu y)P \xrightarrow{\overline{x}(w)} \{{}^{w}\!/\!{}_{y}\}P'}\ w \notin \mathbf{fn}(P)$$

$$\text{PAR-L}\ \frac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q}\ \mathbf{bn}(\alpha) \cap \mathbf{fn}(Q) = \emptyset$$

$$\text{PAR-R}\ \frac{Q \xrightarrow{\alpha} Q'}{P|Q \xrightarrow{\alpha} P|Q'}\ \mathbf{bn}(\alpha) \cap \mathbf{fn}(P) = \emptyset$$

$$\text{CLOSE-1}\ \frac{P \xrightarrow{\overline{x}(v)} P' \quad Q \xrightarrow{x\langle v\rangle} Q'}{P|Q \xrightarrow{\tau} (\nu v)(P'|Q')}$$

$$\text{CLOSE-2}\ \frac{P \xrightarrow{x\langle v\rangle} P' \quad Q \xrightarrow{\overline{x}(v)} Q'}{P|Q \xrightarrow{\tau} (\nu v)(P'|Q')}$$

$$\text{COM-1}\ \frac{P \xrightarrow{\overline{x}\langle y\rangle} P' \quad Q \xrightarrow{x\langle y\rangle} Q'}{P|Q \xrightarrow{\tau} P'|Q'}$$

$$\text{COM-2}\ \frac{P \xrightarrow{x\langle y\rangle} P' \quad Q \xrightarrow{\overline{x}\langle y\rangle} Q'}{P|Q \xrightarrow{\tau} P'|Q'}$$

Figure 6–2: Transition rules for $\mathcal{P}$

A question of greater significance is whether we can capture the model obtained from strong bisimilarity (as given in definition 6.1)

$$P \sim Q \text{ if and only if } \widehat{P} \stackrel{\wedge}{\sim} \widehat{Q}.$$

by any of the bisimilarities suggested in chapter 6. $\mathcal{L}_i$-bisimilarity, which gives an interleaving semantics, seems a likely candidate.

6.5.1 The Asynchronous π-calculus

In the preceding chapters we have informally referred to a correspondence between PIC′ and the π-calculus; therefore, a natural task would be to make this correspondence precise. This may be achieved by, first, defining a translation from the terms of the asynchronous π-calculus to those of PIC′ followed by an comparison between the manifestations of labelled transition relations and strong bisimulation in both calculi. We shall now briefly illustrate what this involves, confining ourselves to the monadic version mainly for simplicity of exposition.

The terms of the asynchonous π-calculus $\mathcal{P}$ essentially correspond to the fragment of the full π-calculus, or more closely, to the ν-calculus of Honda and Yoshida in [11,10].

$$P \;::=\; \mathbf{0} \mid \overline{x}\langle v\rangle \mid x(y).P \mid (\nu x)P \mid P|Q$$

To obtain processes, the terms of $\mathcal{P}$ are factored by a structural congruence $\equiv$ induced by the following equations:

$$\begin{array}{lll}
P|\mathbf{0} \equiv P & (\nu x)(\nu y)P \equiv (\nu y)(\nu x)P & \\
P|Q \equiv Q|P & (\nu x)(P|Q) \equiv P|(\nu x)Q & (x \notin \mathbf{fn}\,(P)) \\
P|(Q|R) \equiv (P|Q)|R & (\nu x)P \equiv (\nu y)(\{y/x\}P) & (y \notin \mathbf{fn}\,(P)) \\
 & z(x).P \equiv z(y).(\{y/x\}P) & (y \notin \mathbf{fn}\,(P))
\end{array}$$

where $\mathbf{fn}\,(P)$ denotes the *free names* of P, with the occurrence of any name x in P being free unless bound in some subterm Q of P, by a $(\nu x)Q$ or $z(x).Q$ construct, whose scope extends throughout the subterm Q.

The dynamics are given in terms of reduction $\rightarrow$ the smallest relation over $\mathcal{P}$ closed under $\equiv$ and the following rules:

$$\textsc{com} : \overline{x}\langle z\rangle | x(y).P \rightarrow \{z/y\}P$$

$$\textsc{par}\frac{P \rightarrow P'}{P|Q \rightarrow P'|Q} \qquad \textsc{res}\frac{P \rightarrow P'}{(\nu x)P \rightarrow (\nu x)P'}$$

In [29], Milner has shown the correspondence between the processes in $\mathcal{P}$ and PIC. The translation to PIC$'$ is identical:

$$\begin{array}{rcl}
\widehat{0} & \stackrel{\text{def}}{=} & \mathbf{id}_\epsilon \\
\widehat{\overline{x}\langle v\rangle} & \stackrel{\text{def}}{=} & \langle v\rangle \cdot \mathbf{out}_x \\
\widehat{x(y).P} & \stackrel{\text{def}}{=} & \mathbf{box}_x(y)\widehat{P} \\
\widehat{(\nu x)P} & \stackrel{\text{def}}{=} & (\nu x)\widehat{P} \\
\widehat{P|Q} & \stackrel{\text{def}}{=} & \widehat{P} \otimes \widehat{Q}
\end{array}$$

The terms of $\mathcal{P}$ translate to PIC$'$ terms of arity $\epsilon \rightarrow \epsilon$. Then, from [29], we have

(1) $P \equiv Q$ if and only if $\widehat{P} = \widehat{Q}$.

(2) If $P \rightarrow Q$ then $\widehat{P} \searrow^1_\tau \widehat{Q}$.

(3) If $\widehat{P} \searrow^1_\tau t$ then for some P', $P \rightarrow P'$ and $\widehat{P'} = t$.

Labelled Transitions In figure 6–3 we give the derivation rules for transitions terms in $\mathcal{P}$. The rules allow the derivation of *early* transitions allowing a precise correspondence between labelled transitions in $\mathcal{P}$ and PIC$'$ to be stated.

The relationship expected between τ transitions in PIC$'$ and reductions in $\mathcal{P}$ is fairly easy to establish. It may be obtained through the intermediate relationship of both relations with single-step reaction. Recall that theorem 5.15 states that, for actions $\widehat{P}, \widehat{P'}$ of arities $\epsilon \rightarrow \epsilon$:

$$\widehat{P} \xrightarrow{\tau}_= \widehat{P'} \iff \widehat{P} \searrow^1_\tau \widehat{P'}$$

$$\text{OUT}\ \frac{}{\overline{x}\langle w\rangle \xrightarrow{\overline{x}\langle w\rangle} \mathbf{0}}$$

$$\text{IN}\ \frac{}{x(y).P \xrightarrow{x\langle w\rangle} \{w/y\}P}$$

$$\text{RES}\ \frac{P \xrightarrow{\alpha} P'}{(\nu x)P \xrightarrow{\alpha} (\nu x)P'}\ x \notin \mathbf{n}(\alpha)$$

$$\text{OPEN}\ \frac{P \xrightarrow{\overline{x}(y)} P'}{(\nu y)P \xrightarrow{\overline{x}(w)} \{w/y\}P'}\ w \notin \mathbf{fn}(P)$$

$$\text{PAR-L}\ \frac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q}\ \mathbf{bn}(\alpha) \cap \mathbf{fn}(Q) = \emptyset$$

$$\text{PAR-R}\ \frac{Q \xrightarrow{\alpha} Q'}{P|Q \xrightarrow{\alpha} P|Q'}\ \mathbf{bn}(\alpha) \cap \mathbf{fn}(P) = \emptyset$$

$$\text{CLOSE-1}\ \frac{P \xrightarrow{\overline{x}(y)} P' \quad Q \xrightarrow{x\langle y\rangle} Q'}{P|Q \xrightarrow{\tau} (\nu v)(P'|Q')}$$

$$\text{CLOSE-2}\ \frac{P \xrightarrow{x\langle y\rangle} P' \quad Q \xrightarrow{\overline{x}(y)} Q'}{P|Q \xrightarrow{\tau} (\nu v)(P'|Q')}$$

$$\text{COM-1}\ \frac{P \xrightarrow{\overline{x}\langle y\rangle} P' \quad Q \xrightarrow{x\langle y\rangle} Q'}{P|Q \xrightarrow{\tau} P'|Q'}$$

$$\text{COM-2}\ \frac{P \xrightarrow{x\langle y\rangle} P' \quad Q \xrightarrow{\overline{x}\langle y\rangle} Q'}{P|Q \xrightarrow{\tau} P'|Q'}$$

Figure 6–3: Transition rules for $\mathcal{P}$

This, together with above relationship between reaction and labelled transition relations gives:

$$P \rightarrow Q \iff \widehat{P} \xrightarrow{\tau}_{=} \widehat{Q}$$

However, we still do not have any information about the relationship between labelled transition relations; and more importantly, between the models of each given by bisimiliarity. In particular we expect the following to hold:

$$P \xrightarrow{\tau} P' \iff \vdash \widehat{P} \xrightarrow{\tau}_{=} \widehat{P'}$$

A question of greater significance is whether we can capture the model obtained from strong bisimilarity (as given in definition 6.1)

$$P \sim Q \text{ if and only if } \widehat{P} \mathrel{\hat{\sim}} \widehat{Q}.$$

by any of the bisimilarities suggested in the previous section. In particular, $\mathcal{L}_i$-bisimilarity, which gives an interleaving semantics, seems a likely candidate.

Chapter 7

Conclusions and Further Work

In this chapter we present some current work on control structures and outline possible directions for further work. The chapter is concluded by a summary of what has been achieved in this thesis.

7.1 Current Research in Control Structures

In all the categories of control structures presented in this thesis, the names X and arities M have been assumed fixed. Milner [27] and Power [35] have considered how this condition can be relaxed while still obtaining the initiality results for action calculi. Both approaches result in attributing greater structure to naming, than present in our definitions where a set of names X suffices. Milner observes that it is easy to refine the structure of names from a set X to the free monoid $(X, \otimes, 1)$ generated by X; with data and abstraction extended as follows:

$$\begin{array}{rcll} \mathbf{ab}_{x_1 \otimes \cdots \otimes x_r} a & \stackrel{\text{def}}{=} & \mathbf{ab}_{x_1} \cdots \mathbf{ab}_{x_r} a & (r \geq 0) \\ \langle x_1 \otimes \cdots \otimes x_r \rangle & \stackrel{\text{def}}{=} & \langle x_1 \rangle \otimes \cdots \otimes \langle x_r \rangle a & (r \geq 0) \end{array}$$

Milner's account then considers which class of monoids—of which $(X, \otimes, 1)$ is a member—contains sufficient structure to allow a generalisation of control structure morphism which removes the requirement that such morphisms act as the identity on the names. This extraction of the essential structure from the free monoid, brings us closer to an abstract account of naming. Power [35] shows how such naming monoids can arise from the arity monoid in a natural fashion.

Another approach in which names are rendered implicit is taken by Gardner [6] who introduced *closed action calculi*—essentially a *name-free* variant of action calculi—and established the precise correspondence with the action calculi (including the reflexive variant) referred to in this thesis. This effort aims to demonstrate that while names play a useful presentational role they are not essential.

An abstract treatment in which names are implicit—but *naming* explicit—in the spirit of categorical logic [15] is provided by Power and Hermida in their *fibrational control structures* [8]. A generalisation of this account is developed by Power [36]; providing connections between control structures and his work with Robinson on a general semantic theory of "notions of computation" [37].

Throughout this thesis we have relied on the idea of dataflow to give an intuitive interpretation of the operations encountered. Indeed, this visualisation of the structure of actions as graphs where links are dataflow channels and nodes are molecules has been of great assistance in developing equational proofs, and also in formulating the labelled transition rules for the reflexive π-calculus. In a recent paper [29], Milner introduced *action graphs* which formalise this intuition. A rigorous treatment of these graphs is to be presented in Ole Jensen's forthcoming PhD thesis [12].

The intuition of actions as graphs informs not just our enquiry into the structure of actions but also that concerning their dynamics: as a result of computation the static structure of an action (the controls and dataflow links) may evolve. The transformation of the action graph resulting from computation may be used to compare the dynamic characteristics of diverse action calculi. A classifier IMGRAPH is being developed by Leifer [17] based on this idea: only for those action calculi in which mobility is not expressible does there exist a homomorphism of control structures to IMGRAPH.

7.2 Further Work

As the work on action structures is relatively recent there is an abundance of virgin territory to explore. Taking the contents of this thesis as a starting point various directions suggest themselves. For instance, the development of classifiers, as examplified by skeleta in chapter 4, could prove a fruitful way of studying the kind of dynamic behaviour expressible by various models. It may also be possible to give a generic form of operational semantics in terms of skeleta, for instance, through relations S with the property that, whenever $a_1 S a_2$, then

i. $a_1 \searrow a_1' \implies a_2 \searrow a_2'$ with $a_1' S a_2'$
ii. $\nu\mathbf{skel}(a_1) = \nu\mathbf{skel}(a_2)$

The contribution of restriction skeleta in the above is highlighted by the fact that the largest binary relation on processes $\cong$ having the property

$$a_1 \cong a_2 \text{ and } a_1 \searrow a_1' \implies a_2 \searrow a_2' \text{ with } a_1' \cong a_2'$$

is the universal relation, which gives a trivial semantics. Thus by examining the pattern of reaction in the image of the action calculus on νSKEL, a comparison of the actions may be made on their ability to generate effects as a result of computation. Indeed, such a comparison may also be made between terms for distinct action calculi, allowing the notion of encoding (of a process term in one action calculus by another term in the other calculus). Such encodings deserve study in their own right; and we suggest that the framework we have presented can be developed to assist such study.

7.2.1 Embeddings

One of the aims of developing control structures is to allow the comparison of concrete models by providing a framework where each model may be represented. One form of comparison may be based on expressiveness, but this in turn requires agreement of what entities are to be expressed; in other words, a common model. A special case in our context arises when the controls of one action calculus AC($\mathcal{K}$) can be encoded in terms of the operations of another AC($\mathcal{K}'$). The encoding, if compositional, can easily be captured as a morphism of static control structures (over $\mathcal{K}$). However, an action a in AC($\mathcal{K}$) and its encoding Φa are to be accepted as expressing the same entity, then, some suitable relationship between the dynamics of a and those of Φa is required.

In order to see the kind of properties such a relationship is expected to imply, consider one possible application for such embeddings: the idea of an implementation. One may think of an implementation for a concrete model as a compiler to a lower level (also concrete) model which may have more objects which are expressible in it. Such a compiler can be expressed as a morphism of control structures from one action calculus (high level) to another (low level). Note that we should not expect the morphism to have an inverse, indeed, nor expect it to be onto. The idea of source and machine languages comes to mind: there may be many machine code programs which are not generated by any Pascal program.

Homomorphisms of action structures (and their refinements) provide a suitable starting point for talking about such embeddings. However, while homomorphisms preserve the operations (giving us a compositional translation from source to target codes, so to speak) they may be too weak to guarantee an acceptable computational correspondence between source and target. We recall that a homomorphism of action structures $\Phi : \mathcal{A} \to \mathcal{B}$ (and hence of control structures, reflexive and strictly reflexive ones) preserve reaction:

$$a \searrow^{\mathcal{A}} a' \implies \Phi(a) \searrow^{\mathcal{B}} \Phi(a')$$

This means that the target object must have *at least* matching computational behaviour to the source object. However, it may also have additional behaviour: this means that it is *not precluded that* the target program will behave as one expects from the source program but there is *no guarantee* that it will not follow some other path in its computation tree! This is, of course, unacceptable as a notion of implementation and, consequently, we require homomorphisms of reflexive control structures that preserve the reaction relation in a stricter fashion. Say that a homomorphism of action structures $\Phi : \mathcal{A} \to \mathcal{B}$ *confines reaction* just when the following property holds for all $a \in \mathcal{A}$:

$$\Phi(a) \searrow^{\mathcal{B}} b \implies \exists a'. a \searrow^{\mathcal{A}} a' \text{ with } b \searrow^{\mathcal{B}} \Phi(a')$$

The intuition behind this condition is that the target object can have additional computational behaviour to the source; however, any such behaviour will necessarily consist of intermediate computations that are guaranteed to lead to a state that is matched by one in the source.

Such morphisms are closed under composition and clearly, the identity morphism confines reaction; therefore, one can speak of categories of control structures in which the morphisms confine reaction. Even when present, the action calculus $\mathsf{AC}(\mathcal{K}, \mathcal{R})$ is not necessarily initial in any such category $\mathbf{CS}^{\mathbf{i}}(\mathcal{K})$, since for any control structure $\mathcal{A}$ in the category, the unique homomorphism from $\mathsf{AC}^{\mathsf{r}}(\mathcal{K})$ to $\mathcal{A}$ in $\mathbf{CS}(\mathcal{K})$ might not be reaction confining and therefore not present in $\mathbf{CS}^{\mathbf{i}}(\mathcal{K})$. If we limit our interest to embeddings of a given action calculus over some signature $\mathcal{K}$ and reaction rules $\mathcal{R}$, then as a suitable category one could take any subcategory of $\mathbf{CS}(\mathcal{K})$ in which the unique morphism from $\mathsf{AC}(\mathcal{K},\mathcal{R})$ to the objects of the subcategory confines reaction.

It is fairly easy to show that the morphism determined by the quotient of a control structure by any *reduction-closed* congruence necessarily confines reaction. Since the universal relation on actions is reduction closed, the unique morphism from $\mathsf{AC}(\mathcal{K},\mathcal{R})$ to the terminal control structure is reaction confining. Therefore, terminal control structure is not excluded from any such subcategory as described above; but the terminal control structure can hardly be considered a suitable structure in which to embed $\mathsf{AC}(\mathcal{K},\mathcal{R})$! One way to exclude such candidates is to impose additional conditions on the morphisms. Here again, classifiers may be useful; requiring that the morphism to the classifier be preserved by the embedding morphism may exclude undesirable candidates and, depending on the choice of classifier, such a condition might be justified by computational considerations.

It will be interesting to explore existing examples of embeddings, such as that of the polyadic π-calculus in the monadic version given in [22], in order to see whether the resulting morphism is indeed reaction confining and also to gain insight in what additional properties such morphisms may be expected to have.

7.3 Summary and Conclusions

In this thesis we have taken a concrete class of action structures—that given by the molecular forms—as a promising starting point in the development of an abstract algebraic account of process construction and concurrent computation. The identification of a suitable abstract structure which underlies the molecular forms, and, it is hoped, concurrent computation at large, was achieved in two broad steps: the first consisting of a term algebra, providing a sort of half-way house between syntax and algebra; and the second step involving an abstract semantic treatment of the operations defining the term algebra. Phrased differently; the first step provides a compositional syntax for representing processes and the second, a space of models for the processes thus specified. In going from action calculi (the term algebra) to control structures (abstract algebra), we were obliged to give a semantic treatment of names: this was achieved by means of the notion of *surface*. While surface has a specific definition which depends on the operations found in control structures, the issue that it serves—the behavioural significance of names beyond their "traditional" role as place-holders—is arguably of wider relevance within the quest for abstract models of concurrency.

The feasibility of the molecular forms as a syntactic framework for representing concrete models validates much of the abovementioned achievement. However, feasibility does not imply optimality, and therefore the consideration of alternatives to, or at least variants of, the molecular forms was a natural step in our enquiry. Two variants were considered and given an analogous semantic treatment. Whether either of the variants will emerge as the preferred structure remains to be seen; it is clear, however, that present in the variants are some intuively appealing aspects, such as greater expressivity of dataflow; a semantic treatment of restriction; and, in the most variant case, garbage collection of restricted but unused names and a revealing characterisation of surface in terms of restriction.

While the treatment of process constructors (statics) reveals rich structural issues, our algebraic framework provides significant support for studying the dynamical aspects of processes. In concurrency theory, the manifestation of interaction and computation is greatly varied and establishing a common basis for representing these dynamic aspects poses a considerable challenge. It is to be expected that a structure which fits all must be a modest one; as indeed is the one employed in our framework: the humble preorder! With so little inherent *abstract* structure, how does one study dynamics in a general fashion? One answer is to adapt existing techniques for obtaining models—such as those based on bisimulation—by recasting them in terms of the generic structure present in all action calculi; in particular, reaction. We have not done this; instead, we have presented a concrete instance of the technique to obtain an operational semantics of the π-calculus cast in our framework. A characterisation of the bisimilarities we have obtained in terms of reaction will provide valuable insight into how the technique can be adapted. An alternative path towards the study of dynamics across action calculi (and their reflexive variants) is through classifiers: by examining the dynamics in the images of the calculi on a common *static* model (the classifier), we can derive insightful comparison based upon their dynamic characteristics. A simple manifestation of this is achieved by equipping the classifier with a specific reaction relation; then, a simple comparison is obtained by the existence or otherwise of a reaction preserving homomorphism. We have shown, by two examples, that with a judicious choice of reaction relation, the basis for such a comparison can be computationally meaningful.

Bibliography

[1] S. Abramsky. Interaction categories and the foundations of typed concurrent programming. In *Proc. '94 Marktoberdorf Summer School.* Springer-Verlag, 1995.

[2] J. P. Banâtre and D. Metayer. The GAMMA model and its discipline of programming. *Science of Computer Programming*, 15:55–77, 1990.

[3] G. Berry and G. Boudol. Concurrency and atomicity. *Theoretical Computer Science*, 96:217–48, 1992.

[4] E. Best, R. Devillers, and J. G. Hall. The box calculus: a new causal algebra with multi-label communication. In G. Rozenberg, editor, *Advances in Petri Nets '92*, volume 609 of *LNCS*, pages 21–69. Springer-Verlag, 1995.

[5] S. L. Bloom and Z. Esik. **Iteration Theories**. Springer-Verlag, 1993.

[6] P. Gardner. A name-free account of action calculi. In *Proc. 11th Conference on Mathematical Foundations of Programming Semantics, Tulane*, 1995.

[7] R. van Glabbeek. The linear time - branching time spectrum. In Eike Best, editor, *CONCUR '93, 4th International Conference on Concurrency Theory*, volume 715 of *LNCS*, pages 66–81. Springer-Verlag, August 1993.

[8] C. Hermida and A. J. Power. Fibrational control structures. In *CONCUR '95, 6th International Conference on Concurrency Theory.* Springer-Verlag, 1995.

[9] C.A.R Hoare. Communicating sequential processes. *Communications of the ACM*, 21:666–677, 1978.

[10] K. Honda and Y. Nobuko. Combinatory representation of mobile processes. In *POPL'94, Conference Record of the 21st Annual Symposium on Principles of Programming Languages*, pages 348–360, 1994.

[11] K. Honda and Y. Nobuko. On reduction-based process semantics. *Theoretical Computer Science*, 151:437–486, 1995.

[12] O. Jensen. Forthcoming PhD thesis, University of Cambridge.

[13] A. Joyal, M. Nielsen, and G. Winskel. Bisimulation from open maps. Technical Report BRICS RS-94-7, Computer Science Department, Aarhus University, 1994.

[14] A. Joyal, R. Street, and D. Verity. Traced monoidal categories. In *Mathematical Proceedings of the Cambridge Philosophical Society*, 1994. To appear.

[15] J. Lambek and P. J. Scott. **Introduction to Higher-Order Categorical Logic**, volume 7 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, 1986.

[16] S. Mac Lane. **Categories for the working mathematician**. Springer-Verlag, 1971.

[17] J. Leifer. Private communication.

[18] J. Meseguer. Conditional rewriting logic as a unified model of concurrency. *Theoretical Computer Science*, 96:73–155, 1992.

[19] J. Meseguer and U. Montanari. Petri nets are monoids. *Journal of Information and Computation*, 88:105–155, 1990.

[20] A. Mifsud, R. Milner, and J. Power. Control structures. In *LICS '95, 10th Annual IEEE Symposium*. IEEE Ccomputer Society, 1995.

[21] R. Milner. Action structures. Research Report ECS-LFCS-92-249, Laboratory for the Foundations of Computer Science, Department of Computer Science, University of Edinburgh, December 1992.

[22] R. Milner. The polyadic π-calculus: A tutorial. Research Report ECS-LFCS-91-180, Laboratory for the Foundations of Computer Science, Department of Computer Science, University of Edinburgh, December 1992.

[23] R. Milner. Action calculi IV : Molecular forms. Draft research note, November 1993.

[24] R. Milner. Action calculi, or syntactic action structures. In *Proc. Mathematical Foundations of Computer Science, Gdansk, Poland*, volume 711 of *LNCS*, pages 105–121. Springer-Verlag, 1993.

[25] R. Milner. Action calculi V : Reflexive molecular forms (with Appendix by Ole Jensen). Research note., November 1993.

[26] R. Milner. Action structures for the π-calculus. Research Report ECS-LFCS-93-264, Laboratory for the Foundations of Computer Science, Department of Computer Science, University of Edinburgh, May 1993.

[27] R. Milner. Control structures II : Naming monoids. Draft, July 1994.

[28] R. Milner. Higher-order action calculi. In Karl Meinke, editor, *Proc. Computer Science Logic, 1992*, volume 832 of *LNCS*, pages 238–260. Springer-Verlag, 1994.

[29] R. Milner. Calculi for Interaction. Draft, April 1995.

[30] R. Milner, J. Parrow, and D.Walker. A calculus of mobile processes, Parts I and II. *Journal of Information and Computation*, 100:1–77, 1992.

[31] Robin Milner. **Communication and Concurrency**. Prentice-Hall, 1989.

[32] M. Nielsen, V. Sassone, and G. Winskel. A classification of models for concurrency. In Eike Best, editor, *CONCUR '93, 4th International Conference on Concurrency Theory*, volume 715 of *LNCS*, pages 82–96. Springer-Verlag, August 1993.

[33] M. Nielsen and G. Winskel. Models for concurrency. Technical Report DAIMI PB-429, Computer Science Department, Aarhus University, 1992.

[34] C.A. Petri. Fundamentals of a theory of asynchronous information flow. In *Proc. IFIP Congress '62*, pages 386–390. North Holland, 1962.

[35] A. J. Power. Control structures III : Arity monoids and their associated naming monoids. Draft, August 1994.

[36] A. J. Power. Elementary control structures. Submitted, January 1996.

[37] A. J. Power and E. P. Robinson. Premonoidal categories and notions of computation. Submitted, 1996.

[38] D. Sangiorgi. Bisimulation in higher-order process calculi. Submitted for IFIP'94, 1994.

[39] G. Stefănescu. Feedback theories (a calculus for isomorphism classes of flow-chart schemes). Technical Report, Preprint Series in Mathematics No. 24, The National Institute for Scientific and Technical Creation, Bucharest, 1986.

Appendix A

Proofs

Note All derivable sequents referred to in this appendix are assumed derivable by the rules $\mathcal{R}$; in other words, the DISCARD rule is not used.

A.1 Auxiliary Results

The following lemmas are used in those proofs deferred from the main text to this appendix. The results in this section are of purely technical necessity and were not deemed sufficiently interesting for inclusion in the main text.

Lemma A.1 *Let $\vec{x}, \vec{y} : m$ such that $\{\vec{x}\} \cap \{\vec{y}\} = \emptyset$. Then*

$$\uparrow_m(\vec{x})(t_1 \cdot (\vec{u})(\langle\vec{y}\rangle \otimes t_2)) = \uparrow_m(\vec{x})(t_1 \cdot (\vec{u})(\langle\vec{y}\rangle \otimes \{\vec{y}/\vec{x}\}t_2))$$

Proof Induction on $r = |\vec{x}|$.

Base Case: $r = 0$ Immediate.

Inductive Step: $r = j + 1$ Assume $t_1 : k \to l$, $(\vec{u})t_2 : l \to n$ and, by alphaconversion, $\{\vec{u}\} \cap \mathbf{fn}\,(t_1) = \emptyset$.

$$
\begin{aligned}
&\uparrow_{p\otimes m}(v\vec{x})(t_1 \cdot (\vec{u})(\langle w\vec{y}\rangle \otimes t_2)) \\
&= \uparrow_m(\vec{x})\uparrow_p(v)(t_1 \cdot (\vec{u})(\langle w\rangle \otimes \langle\vec{y}\rangle \otimes t_2)) && 3.26(1) \\
&= \uparrow_m(\vec{x})\uparrow_p(v)\uparrow_l(\mathbf{p}_{l,k} \cdot (t_1 \otimes (\vec{u})(\langle w\rangle \otimes \langle\vec{y}\rangle \otimes t_2))) && 3.10(1) \\
&= \uparrow_m(\vec{x})\uparrow_p(v)\uparrow_l(\mathbf{p}_{l,k} \cdot \mathbf{p}_{k,l} \\
&\qquad \cdot(\vec{u})(\langle w\rangle \otimes \langle\vec{y}\rangle \otimes t_2 \otimes t_1) \cdot \mathbf{p}_{p\otimes m\otimes n,l}) && 2.24(1),\ 2.33(2) \\
&= \uparrow_m(\vec{x})\uparrow_p(v)\uparrow_l(\vec{u})((\langle w\rangle \otimes \langle\vec{y}\rangle \otimes t_2 \otimes t_1) \cdot \mathbf{p}_{p\otimes m\otimes n,l}) && S_2,\ 2.33(1) \\
&= \uparrow_m(\vec{x})\uparrow_l(\vec{u})\uparrow_p(v)((\langle w\rangle \otimes \langle\vec{y}\rangle \otimes t_2 \otimes t_1) \\
&\qquad \cdot\mathbf{p}_{p\otimes m\otimes n,l} \cdot (\mathbf{p}_{l,p} \otimes \mathbf{id})) && 3.26(4) \\
&= \uparrow_m(\vec{x})\uparrow_l(\vec{u})\uparrow_p(v)((\langle w\rangle \otimes \langle\vec{y}\rangle \otimes t_2 \otimes t_1) \\
&\qquad \cdot(\mathbf{id}_p \otimes \mathbf{p}_{m\otimes n,l})) && 2.24(3),\ S_2 \\
&= \uparrow_m(\vec{x})\uparrow_l(\vec{u})(\uparrow_p(v)(\langle w\rangle \otimes \langle\vec{y}\rangle \otimes t_2 \otimes t_1) \cdot \mathbf{p}_{m\otimes n,l}) && \rho_3 \\
&= \uparrow_m(\vec{x})\uparrow_l(\vec{u})(\{w/v\}(\langle\vec{y}\rangle \otimes t_2 \otimes t_1) \cdot \mathbf{p}_{m\otimes n,l}) && 3.29(1) \\
&= \uparrow_m(\vec{x})\uparrow_l(\vec{u})(\{w/v\}(\langle\vec{y}\rangle \otimes \{w/v\}t_2 \otimes t_1) \cdot \mathbf{p}_{m\otimes n,l}) \\
&\ \vdots \qquad\qquad \text{by reverse argument} \\
&= \uparrow_m(\vec{x})\uparrow_p(v)(t_1 \cdot (\vec{u})(\langle w\rangle \otimes \langle\vec{y}\rangle \otimes \{w/v\}t_2)) \\
&= \uparrow_p(v)\uparrow_m(\vec{x})(t_1 \cdot (\vec{u})(\langle w\rangle \otimes \langle\vec{y}\rangle \otimes \{w/v\}t_2) \\
&\qquad \cdot(\mathbf{p}_{p,m} \otimes \mathbf{id})) && 3.26(4) \\
&= \uparrow_p(v)\uparrow_m(\vec{x})(t_1 \cdot (\vec{u})(\langle\vec{y}\rangle \otimes \langle w\rangle \otimes \{w/v\}t_2)) && \zeta,\ 2.24(4) \\
&= \uparrow_p(v)\uparrow_m(\vec{x})(t_1 \cdot (\vec{u})(\langle\vec{y}\rangle \otimes \{\vec{y}/\vec{x}\}\langle w\rangle \otimes \{\vec{y}/\vec{x}\}\{w/v\}t_2)) && \text{induction} \\
&= \uparrow_p(v)\uparrow_m(\vec{x})(t_1 \cdot (\vec{u})(\langle\vec{y}\rangle \otimes \langle w\rangle \otimes \{w\vec{y}/v\vec{x}\}t_2)) \\
&= \uparrow_{p\otimes m}(v\vec{x})(t_1 \cdot (\vec{u})(\langle w\vec{y}\rangle \otimes \{w\vec{y}/v\vec{x}\}t_2)) && 3.26,\ \zeta,\ 2.24(4)
\end{aligned}
$$

■

Lemma A.2 *Let $\vec{x},\vec{y} : k$ and $\vec{z}_1 : m, \vec{z}_2 : n$.*

1. $\langle\vec{y}\vec{z}\rangle \vdash (\vec{x})t \xrightarrow{\ell}_= t' \iff \langle\vec{z}\rangle \vdash \{\vec{y}/\vec{x}\}t \xrightarrow{\ell}_= t'$
2. $\langle\vec{z}_1\vec{z}_2\rangle \vdash \mathbf{p}_{m,n} \xrightarrow{\langle\vec{z}_2\vec{z}_1\rangle}_= \mathbf{id}_\epsilon$.

where $\{\vec{y}/\vec{x}\}t$ is the simultaneous substitution *of $\vec{y}$ for $\vec{x}$ in t.*

Proof

(1) Induction on $r = |\vec{x}|$. Base case follows immediately. For the inductive step of ($\Longrightarrow$), consider the standard derivation of $\langle w\vec{y}\vec{z}\rangle \vdash (u\vec{x})t \xrightarrow{\ell}_= t'$. For some

t'', there is some subderivation giving $\langle w\vec{y}\vec{z}\rangle \vdash (u\vec{x})t \xrightarrow{\delta}_{=} t''$ which consists only of constructor rules, and from which the resulting derivation is obtained by a sequence of PERM and SYNC rules. Consider the last part of such a derivation, with $\delta = (\vec{u})\vec{\alpha}\langle\vec{v}\rangle$; in the standard derivation it must have the following form:

$$\dfrac{\dfrac{\langle\vec{y}\vec{z}\rangle \vdash \{^{w}\!/\!_{u}\}(\vec{x})t \xrightarrow{(\vec{u})\vec{\alpha}\langle\vec{v}\rangle} t''}{\langle w\vec{y}\vec{z}\rangle \vdash \mathbf{ab}_u(\vec{x})t \xrightarrow{(\vec{u})\vec{\alpha}\langle w\vec{v}\rangle} t''}\,\mathbf{ab} \quad \dfrac{\langle w\rangle \vdash \omega \xrightarrow{\langle\,\rangle} \mathbf{id}_\epsilon \quad \langle\vec{v}\rangle \vdash \mathbf{id} \xrightarrow{\langle\vec{v}\rangle} \mathbf{id}_\epsilon}{\langle w\vec{v}\rangle \vdash \omega \otimes \mathbf{id} \xrightarrow{\langle\vec{v}\rangle}_{=} \mathbf{id}_\epsilon}\,\otimes}{\langle w\vec{y}\vec{z}\rangle \vdash \mathbf{ab}_u(\vec{x})t \cdot (\omega \otimes \mathbf{id}) \xrightarrow{(\vec{u})\vec{\alpha}\langle\vec{v}\rangle}_{=} t''}\,\circ$$

Since $\{^{w}\!/\!_{u}\}(\vec{x})t \equiv (\vec{x}')\{^{w}\!/\!_{u}\}\{^{\vec{x}'}\!/\!_{\vec{x}}\}t$, for some $\vec{x}'$ such that $\{u, w\} \cap \{\vec{x}'\} = \emptyset$. Hence, we have

$$\langle\vec{y}\vec{z}\rangle \vdash (\vec{x}')\{^{w}\!/\!_{u}\}\{^{\vec{x}'}\!/\!_{\vec{x}}\}t \xrightarrow{(\vec{u})\vec{\alpha}\langle\vec{v}\rangle} t''$$

and by inductive hypothesis:

$$\langle\vec{z}\rangle \vdash \{^{\vec{y}}\!/\!_{\vec{x}'}\}\{^{w}\!/\!_{u}\}\{^{\vec{x}'}\!/\!_{\vec{x}}\}t \xrightarrow{(\vec{u})\vec{\alpha}\langle\vec{v}\rangle}_{=} t''$$

But, $\{^{\vec{y}}\!/\!_{\vec{x}'}\}\{^{w}\!/\!_{u}\}\{^{\vec{x}'}\!/\!_{\vec{x}}\}t \equiv \{^{w\vec{y}}\!/\!_{u\vec{x}}\}t$. Hence, by applying the same sequence of PERM and SYNC rules as in the standard derivation of the left transition gives the required result.

For the inductive step of ($\Longleftarrow$), we use the fact that $\{^{w\vec{y}}\!/\!_{u\vec{x}}\}t \equiv \{^{\vec{y}}\!/\!_{\vec{x}'}\}\{^{w}\!/\!_{u}\}\{^{\vec{x}'}\!/\!_{\vec{x}}\}t$. Then, by inductive hypothesis, we have

$$\langle\vec{y}\vec{z}\rangle \vdash (\vec{x}')\{^{w}\!/\!_{u}\}\{^{\vec{x}'}\!/\!_{\vec{x}}\}t \xrightarrow{\ell}_{=} t'$$

But $(\vec{x}')\{^{w}\!/\!_{u}\}\{^{\vec{x}'}\!/\!_{\vec{x}}\}t \equiv \{^{w}\!/\!_{u}\}(\vec{x})t$. Then, since $\ell = (\vec{u})\vec{\alpha}\langle\vec{v}\rangle$ for some $\vec{u}, \vec{v}, \vec{\alpha}$, by the above derivation the required result follows.

(2) For any $\vec{x}$, it is demonstrable by easy induction on $|\vec{x}|$, that $\vdash \langle\vec{x}\rangle \xrightarrow{\langle\vec{x}\rangle}_{=} \mathbf{id}_\epsilon$. Then, since $\mathbf{p}_{m,n} \equiv (\vec{x}_1\vec{x}_2)\langle\vec{x}_2\vec{x}_1\rangle$, for some distinct names $\vec{x}_1\vec{x}_2 : m \otimes n$, the result follows immediately by (1). ■

Lemma A.3 *Let $\vec{\alpha} : k \to l$, $\vec{x}, \vec{w} : m$, and $\{\vec{x}\} \cap \{\vec{w}\} = \emptyset$. Then whenever $\langle\vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\vec{\alpha}\langle\vec{w}\vec{v}\rangle} t'$ the following is derivable:*

$$\langle\vec{z}\rangle \vdash \uparrow_m(\vec{x})t \xrightarrow{(\vec{u})\{\vec{w}/\vec{x}\}\vec{\alpha}(\{\vec{w}/\vec{x}\}\vec{v})}_{=} \uparrow_m(\vec{x})(t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle\vec{w}\vec{u}\rangle) \cdot (\mathbf{p}_{l,m} \otimes \mathbf{id}))$$

Proof Induction on $r = |\vec{x}|$.

Base Case: $r = 0$ Immediate.

Inductive Step: $r = j + 1$ Let $y, y' : p$ such that $\{\vec{x}y\} \cap \{\vec{w}y'\} = \emptyset$. Since, by lemma 3.26(1), $\uparrow_{m\otimes p}(\vec{x})(y)t = \uparrow_p(y)\uparrow_m(\vec{x})t$, we can consider of $\uparrow_p(y)\uparrow_m(\vec{x})t$ as principal term (by structural lemma). Assume $\langle\vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\vec{\alpha}\langle\vec{w}y'\vec{v}\rangle} t'$ is derivable. Then, by the inductive hypothesis, and since $\{\vec{w}/\vec{x}\}y' = y'$, we have:

$$\langle\vec{z}\rangle \vdash \uparrow_m(\vec{x})t \xrightarrow{(\vec{u})\{\vec{w}/\vec{x}\}\vec{\alpha}\langle y'\{\vec{w}/\vec{x}\}\vec{v}\rangle}_{=} \uparrow_m(\vec{x})(t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle\vec{w}\vec{u}\rangle) \cdot (\mathbf{p}_{l,m} \otimes \mathbf{id}))$$

By lemma A.2(1), we get

$$\langle y\vec{z}\rangle \vdash (y)\uparrow_m(\vec{x})t \xrightarrow{(\vec{u})\{\vec{w}/\vec{x}\}\vec{\alpha}\langle y'\{\vec{w}/\vec{x}\}\vec{v}\rangle}_{=} \uparrow_m(\vec{x})(t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle\vec{w}\vec{u}\rangle) \cdot (\mathbf{p}_{l,m} \otimes \mathbf{id}))$$

and by the rule $\uparrow_2$, we can derive:

$$\langle\vec{z}\rangle \vdash \uparrow_p(y)\uparrow_m(\vec{x})t \xrightarrow{(\vec{u})\{y'\vec{w}/y\vec{x}\}\vec{\alpha}\langle\{y'\vec{w}/y\vec{x}\}\vec{v}\rangle}_{=} t''$$

where $t'' = \uparrow_p(y)(\uparrow_m(\vec{x})(t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle\vec{w}\vec{u}\rangle) \cdot (\mathbf{p}_{l,m} \otimes \mathbf{id})) \cdot (\mathbf{id}_l \otimes (\vec{u})\langle y'\vec{u}\rangle) \cdot (\mathbf{p}_{l,p} \otimes \mathbf{id}))$.
We must now show that the residual term is equal to the term we expect:

$$\begin{aligned}
&\uparrow_p(y)(\uparrow_m(\vec{x})(t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle\vec{w}\vec{u}\rangle) \cdot (\mathbf{p}_{l,m} \otimes \mathbf{id})) \cdot (\mathbf{id}_l \otimes (\vec{u})\langle y'\vec{u}\rangle) \cdot (\mathbf{p}_{l,p} \otimes \mathbf{id})) \\
&\quad = \uparrow_p(y)\uparrow_m(\vec{x})(t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle\vec{w}\vec{u}\rangle) \cdot (\mathbf{p}_{l,m} \otimes \mathbf{id}) \\
&\qquad\qquad \cdot(\mathbf{id}_{m\otimes l} \otimes (\vec{u})\langle y'\vec{u}\rangle) \cdot (\mathbf{id}_m \otimes \mathbf{p}_{l,p} \otimes \mathbf{id})) && \rho_3, 2.16(1) \\
&\quad = \uparrow_p(y)\uparrow_m(\vec{x})(t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle\vec{w}\vec{u}\rangle) \cdot (\mathbf{id}_{l\otimes m} \otimes (\vec{u})\langle y'\vec{u}\rangle) \\
&\qquad\qquad \cdot(\mathbf{p}_{l,m} \otimes \mathbf{id}) \cdot (\mathbf{id}_m \otimes \mathbf{p}_{l,p} \otimes \mathbf{id})) && \zeta \\
&\quad = \uparrow_p(y)\uparrow_m(\vec{x})(t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle\vec{w}\vec{u}\rangle) \\
&\qquad\qquad \cdot(\mathbf{id}_{l\otimes m} \otimes (\vec{u})\langle y'\vec{u}\rangle) \cdot (\mathbf{p}_{l,m\otimes p} \otimes \mathbf{id})) && \mathrm{S}_3 \\
&\quad = \uparrow_p(y)\uparrow_m(\vec{x})(t' \cdot (\mathbf{id}_l \otimes ((\vec{u})\langle\vec{w}\vec{u}\rangle) \\
&\qquad\qquad \cdot(\mathbf{id}_m \otimes (\vec{u})\langle y'\vec{u}\rangle))) \cdot (\mathbf{p}_{l,m\otimes p} \otimes \mathbf{id})) \\
&\quad = \uparrow_p(y)\uparrow_m(\vec{x})(t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle\vec{w}y'\vec{u}\rangle) \cdot (\mathbf{p}_{l,m\otimes p} \otimes \mathbf{id})) && 2.16(1), \sigma \\
&\quad = \uparrow_{m\otimes p}(\vec{x}y)(t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle\vec{w}y'\vec{u}\rangle) \cdot (\mathbf{p}_{l,m\otimes p} \otimes \mathbf{id})) && 3.26
\end{aligned}$$

■

Lemma A.4 *Let $\alpha : k \to l$. Then, whenever $\langle\vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\vec{\alpha}\langle\vec{v}\rangle} t'$, we also have, for any $\vec{w}$ such that $\{\vec{w}\} = \{\vec{u}\}$:*

$$\langle\vec{z}\rangle \vdash t \xrightarrow{(\vec{w})\vec{\alpha}\langle\vec{v}\rangle} t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle\vec{w}\rangle)$$

Proof Every permutation $\vec{w}$ of $\vec{u}$ can be obtained from $\vec{u}$ by some number n of successive commutations of adjacent names. The proof is by a straightforward induction on n. ∎

Lemma A.5 *The following is derivable:*

$$\vdash \widehat{\mu_1} \otimes \cdots \otimes \widehat{\mu_r} \otimes \langle\vec{v}\rangle \xrightarrow{(\vec{u})\langle\vec{w}\vec{v}\rangle} (\widehat{\mu_1} \otimes \cdots \otimes \widehat{u_r}) \cdot (\vec{w})\langle\vec{u}\rangle$$

for any $\vec{u}, \vec{w}$ such that

1. $\{\vec{u}\} \cap \{\vec{v}\} = \emptyset$
2. $\{\vec{u}\} = \{\vec{w}\}$.

Proof Straightforward induction on r. ∎

A.2 Structural Lemma

In this section we shall give a proof of the structural lemma in considerable detail. In the equational proofs, the lemmas used for each step should be obvious in most cases, and explicit reference is only made when the lemma used appears in the appendix.

Lemma 5.11 (Structural) *Whenever* $t_1 \doteq t_2$ *and* $\langle\vec{z}\rangle \vdash t_1 \xrightarrow{\ell} t_1'$ *then, for some* t_2', $\langle\vec{z}\rangle \vdash t_2 \xrightarrow{\ell} t_2'$ *with* $t_1' = t_2'$.

Proof First we shall consider those transitions derived using only the constructor elimination rules i.e. those in which the SYNC and permutation rules do not occur. For each axiom, $t_L = t_R$ we consider the derivable transitions of t_L and t_R under arbitrary environments $\langle\vec{z}\rangle$. We show that whenever there is a derivation of $\langle\vec{z}\rangle \vdash t_L \xrightarrow{\ell} t_L'$ using *just* the constructor rules, then for some t_R', there also exists a derivation (using any of the rules) of $\langle\vec{z}\rangle \vdash t_R \xrightarrow{\ell} t_R'$ with $t_L' = t_R'$ and vice versa.

We shall adopt the following method. For the constructor part of the derivation, each rule applied reduces the size of the term. Now each axiom has the form $C[\vec{t}] = C'[\vec{t}]$. For each side of the axiom we give the final part of all possible derivations up to premisses whose principal term is one of $\vec{t}$. For each derivation with one side of the axiom as principal term, we are done if we can find a matching derivation (with identical label and equal residual) *starting from the same premisses incorporating the terms* $\vec{t}$ with the other side as principal term. Indeed, we need not be so strict about the premisses, since by the substitution lemma, we can be sure of the existence of derivations for variants of the premisses which differ by the replacement of free names throughout the sequent. Thus, to keep the proof relatively short and readable we will present matching derivations for both sides of each axiom, and show that the residuals in each case are equal. We will not explicitly point out the use of the substitution lemma, as in all cases it is quite clear. An important point is that both parts of the substitution lemma may be used since there cannot be any τ-particles in those labels occurring in sequents derived using just the constructor rules.

The proof for some of the axioms (such as $\mathbf{C}_1$, $\mathbf{P}_1$ etc.) is straightforward. We shall describe the proof in the case of $\mathbf{C}_1$ but not of the others as they are either very simple or follow similar lines.

<u>Axiom C_1: $t \cdot \mathbf{id} = t = \mathbf{id} \cdot t$</u> Assume $\langle\vec{z}\rangle \vdash t \cdot \mathbf{id} \xrightarrow{\ell} t'$ by constructor rules. Clearly, the last rule applied must be that for composition. In this case, the following derivation for a transition incorporating the label $\ell = (\vec{u})\vec{\alpha}\langle\vec{v}\rangle$ is unique for the term $t \cdot \mathbf{id}$.

$$\frac{\langle\vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\vec{\alpha}\langle\vec{v}\rangle} t'' \quad \langle\vec{v}\rangle \vdash \mathbf{id} \xrightarrow{\langle\vec{v}\rangle} \mathbf{id}_\epsilon}{\langle\vec{z}\rangle \vdash t \cdot \mathbf{id} \xrightarrow{(\vec{u})\vec{\alpha}\langle\vec{v}\rangle} (t'' \otimes \mathbf{id}_\epsilon) \cdot (\mathbf{id}_\epsilon \otimes \mathbf{ab}_{\vec{u}}\mathbf{id}_\epsilon) \cdot (\mathbf{id}_\epsilon \otimes \mathbf{p}_{r,\epsilon} \otimes \mathbf{id})} \circ$$

Clearly, $t'' = t'$. Hence we can use this subderivation both to show the existence of a derivation for the transition $\langle\vec{z}\rangle \vdash t \xrightarrow{\ell} t''$ (for some $t'' = t'$) from that for $t \cdot \mathbf{id}$ and also as construction of the derivation of $\langle\vec{z}\rangle \vdash t \cdot \mathbf{id} \xrightarrow{\ell}_= t'$ (replacing t'' by t' in the above derivation) from the derivation of $\langle\vec{z}\rangle \vdash t \xrightarrow{\ell} t'$.

The result for the axiom $t = \mathbf{id} \cdot t$ follows in a similar manner by the subderivation shown below:

$$\frac{\langle\vec{z}\rangle \vdash \mathbf{id} \xrightarrow{\langle\vec{z}\rangle} \mathbf{id}_\epsilon \quad \langle\vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\vec{\alpha}\langle\vec{v}\rangle} t''}{\langle\vec{z}\rangle \vdash \mathbf{id} \cdot t \xrightarrow{(\vec{u})\vec{\alpha}\langle\vec{v}\rangle} (\mathbf{id}_\epsilon \otimes \mathbf{id}_k) \cdot (\mathbf{id}_\epsilon \otimes t'') \cdot (\mathbf{id}_\epsilon \otimes \mathbf{p}_{\epsilon,l} \otimes \mathbf{id})} \circ$$

<u>Axiom C_2: $t_1 \cdot (t_2 \cdot t_3) = (t_1 \cdot t_2) \cdot t_3$</u> We shall write the last part of the derivation in each case until subderivations with principal terms t_1, t_2 and t_3. It is easy to see by comparing the derivation, that given the existence of one, one can construct the other. Let $(\vec{u})\vec{\alpha}\langle\vec{v}\rangle = (\vec{u}_1\vec{u}_2\vec{u}_3)\vec{\alpha}_1\vec{\alpha}_2\vec{\alpha}_3\langle\vec{v}_3\rangle$.

Left term t_L: In the following derivation, we also have side conditions

1. $|\vec{u}_1| = r_1$
2. $\{\vec{u}_2\vec{u}_3\} \cap (\mathbf{fn}\,(t_1) \cup \{\vec{z}\}) = \emptyset$
3. $|\vec{u}_2| = r_2$
4. $\{\vec{u}_3\} \cap (\mathbf{fn}\,(t_2) \cup \{\vec{v}_2\}) = \emptyset$

$$\frac{\langle\vec{z}\rangle \vdash t_1 \xrightarrow{(\vec{u}_1)\vec{\alpha}_1\langle\vec{v}_1\rangle} t_1' \quad \dfrac{\langle\vec{v}_1\rangle \vdash t_2 \xrightarrow{(\vec{u}_2)\vec{\alpha}_2\langle\vec{v}_2\rangle} t_2' \quad \langle\vec{v}_2\rangle \vdash t_3 \xrightarrow{(\vec{u}_3)\vec{\alpha}_2\langle\vec{v}_3\rangle} t_3'}{\langle\vec{z}\rangle \vdash t_2 \cdot t_3 \xrightarrow{(\vec{u}_2\vec{u}_3)\vec{\alpha}_2\vec{\alpha}_3\langle\vec{v}_3\rangle} (t_2' \otimes \mathbf{id}_{k_3}) \cdot (\mathbf{id}_{l_2} \otimes \mathbf{ab}_{\vec{u}_2}t_3') \cdot (\mathbf{id}_{l_2} \otimes \mathbf{p}_{r_2,l_3} \otimes \mathbf{id})} \circ}{\langle\vec{z}\rangle \vdash t_1 \cdot (t_2 \cdot t_3) \xrightarrow{(\vec{u})\vec{\alpha}\langle\vec{v}\rangle} t_L'} \circ$$

where $t'_L \equiv (t'_1 \otimes \mathbf{id}_{k_2 \otimes k_3}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}_1} t''_L) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1, l_2 \otimes l_3} \otimes \mathbf{id})$
and $t''_L \equiv (t'_2 \otimes \mathbf{id}_{k_3}) \cdot (\mathbf{id}_{l_2} \otimes \mathbf{ab}_{\vec{u}_2} t'_3) \cdot (\mathbf{id}_{l_2} \otimes \mathbf{p}_{r_2, l_3} \otimes \mathbf{id})$.

Right term t_R: In the following derivation, we also have side conditions

1. $|\vec{u}_1| = r_1$
2. $\{\vec{u}_2\} \cap (\mathbf{fn}\,(t_1) \cup \{\vec{z}\}) = \emptyset$
3. $|\vec{u}_1 \vec{u}_2| = r_1 \otimes r_2$
4. $\{\vec{u}_3\} \cap (\mathbf{fn}\,(t_1) \cup \mathbf{fn}\,(t_2) \cup \{\vec{v}_2\}) = \emptyset$

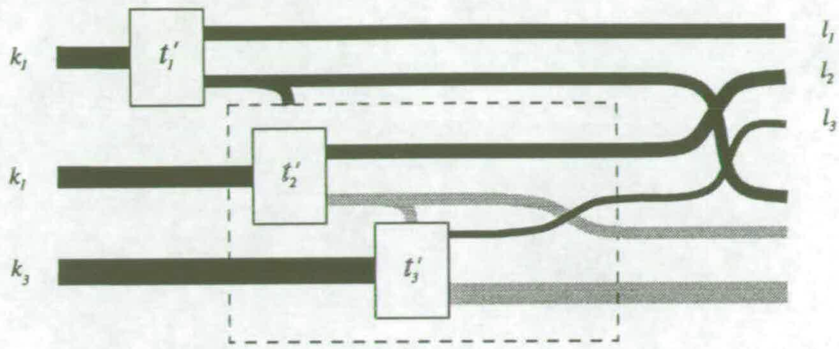
$$\dfrac{\dfrac{\langle \vec{z} \rangle \vdash t_1 \xrightarrow{(\vec{u}_1)\vec{\alpha}_1\langle \vec{v}_1 \rangle} t'_1 \quad \langle \vec{v}_1 \rangle \vdash t_2 \xrightarrow{(\vec{u}_2)\vec{\alpha}_2\langle \vec{v}_2 \rangle} t'_2}{\langle \vec{z} \rangle \vdash t_1 \cdot t_2 \xrightarrow{(\vec{u}_1\vec{u}_2)\vec{\alpha}_1\vec{\alpha}_2\langle \vec{v}_2 \rangle} (t'_1 \otimes \mathrm{id}_{k_2}) \cdot (\mathrm{id}_{l_1} \otimes \mathrm{ab}_{\vec{u}_1} t'_2) \cdot (\mathrm{id}_{l_1} \otimes \mathrm{p}_{r_1, l_2} \otimes \mathrm{id})} \circ \quad \langle \vec{v}_2 \rangle \vdash t_3 \xrightarrow{(\vec{u}_3)\vec{\alpha}_3\langle \vec{v}_3 \rangle} t'_3}{\langle \vec{z} \rangle \vdash (t_1 \cdot t_2) \cdot t_3 \xrightarrow{(\vec{u})\vec{\alpha}\langle \vec{v} \rangle} t'_R} \circ$$

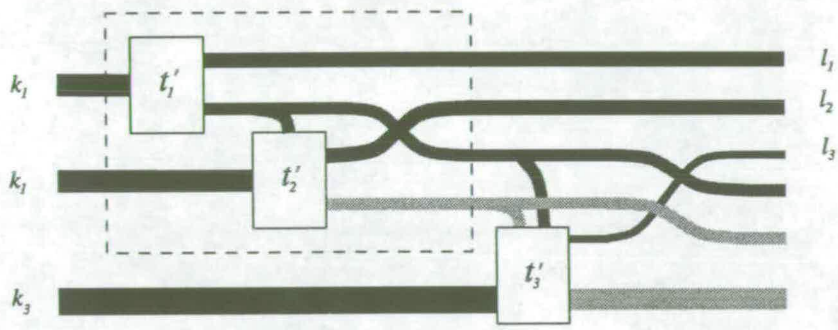
where $t'_R \equiv (t''_R \otimes \mathbf{id}_{k_3}) \cdot (\mathbf{id}_{l_1 \otimes l_2} \otimes \mathbf{ab}_{\vec{u}_1 \vec{u}_2} t'_3) \cdot (\mathbf{id}_{l_1 \otimes l_2} \otimes \mathbf{p}_{r_1 \otimes r_2, l_3} \otimes \mathbf{id})$
and $t''_R \equiv (t'_1 \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}_1} t'_2) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1, l_2} \otimes \mathbf{id})$.

It is easy to see that the side conditions in each derivation are equivalent. We must now show that $t'_L = t'_R$.

$$\begin{aligned}
t'_L &= (t'_1 \otimes \mathrm{id}_{k_2 \otimes k_3}) \cdot (\mathrm{id}_{l_1} \otimes \mathrm{ab}_{\vec{u}_1}((t'_2 \otimes \mathrm{id}_{k_3}) \cdot (\mathrm{id}_{l_2} \otimes \mathrm{ab}_{\vec{u}_2} t'_3) \\
&\qquad \cdot (\mathrm{id}_{l_2} \otimes \mathrm{p}_{r_2, l_3} \otimes \mathrm{id}))) \cdot (\mathrm{id}_{l_1} \otimes \mathrm{p}_{r_1, l_2 \otimes l_3} \otimes \mathrm{id}) \\
&= (t'_1 \otimes \mathrm{id}_{k_2 \otimes k_3}) \cdot (\mathrm{id}_{l_1} \otimes \mathrm{ab}_{\vec{u}_1}((t'_2 \otimes \mathrm{id}_{k_3}) \cdot (\mathrm{id}_{l_2} \otimes \mathrm{ab}_{\vec{u}_2} t'_3))) \\
&\qquad \cdot (\mathrm{id}_{l_1 \otimes r_1 \otimes l_2} \otimes \mathrm{p}_{r_2, l_3} \otimes \mathrm{id}) \cdot (\mathrm{id}_{l_1} \otimes \mathrm{p}_{r_1, l_2 \otimes l_3} \otimes \mathrm{id}) \\
&= (t'_1 \otimes \mathrm{id}_{k_2 \otimes k_3}) \cdot (\mathrm{id}_{l_1} \otimes \mathrm{ab}_{\vec{u}_1}((t'_2 \otimes \mathrm{id}_{k_3}) \cdot (\mathrm{id}_{l_2} \otimes \mathrm{ab}_{\vec{u}_2} t'_3))) \\
&\qquad \cdot (\mathrm{id}_{l_1} \otimes \mathrm{p}_{r_1, l_2} \otimes \mathrm{id}) \cdot (\mathrm{id}_{l_1 \otimes l_2} \otimes \mathrm{p}_{r_1 \otimes r_2, l_3} \otimes \mathrm{id}) \\
&= (t'_1 \otimes \mathrm{id}_{k_2 \otimes k_3}) \cdot (\mathrm{id}_{l_1} \otimes ((\mathrm{ab}_{\vec{u}_1} t'_2 \otimes \mathrm{id}_{k_3}) \cdot (\mathrm{ab}_{\vec{u}_1}(\mathrm{id}_{l_2} \otimes \mathrm{ab}_{\vec{u}_2} t'_3)))) \\
&\qquad \cdot (\mathrm{id}_{l_1} \otimes \mathrm{p}_{r_1, l_2} \otimes \mathrm{id}) \cdot (\mathrm{id}_{l_1 \otimes l_2} \otimes \mathrm{p}_{r_1 \otimes r_2, l_3} \otimes \mathrm{id}) \\
&= (t'_1 \otimes \mathrm{id}_{k_2 \otimes k_3}) \cdot (\mathrm{id}_{l_1} \otimes ((\mathrm{ab}_{\vec{u}_1} t'_2 \otimes \mathrm{id}_{k_3}) \cdot (\mathrm{ab}_{\vec{u}_1}(\mathrm{id}_{l_2} \otimes \mathrm{ab}_{\vec{u}_2} t'_3)) \\
&\qquad \cdot (\mathrm{p}_{r_1, l_2} \otimes \mathrm{id}))) \cdot (\mathrm{id}_{l_1 \otimes l_2} \otimes \mathrm{p}_{r_1 \otimes r_2, l_3} \otimes \mathrm{id}) \\
&= (t'_1 \otimes \mathrm{id}_{k_2 \otimes k_3}) \cdot (\mathrm{id}_{l_1} \otimes ((\mathrm{ab}_{\vec{u}_1} t'_2 \otimes \mathrm{id}_{k_3}) \cdot (\mathrm{p}_{r_1, l_2} \otimes \mathrm{id}) \cdot (\mathrm{id}_{l_2} \otimes \mathrm{ab}_{\vec{u}_1 \vec{u}_2} t'_3)) \\
&\qquad \cdot (\mathrm{id}_{l_1 \otimes l_2} \otimes \mathrm{p}_{r_1 \otimes r_2, l_3} \otimes \mathrm{id}) \\
&= (((t'_1 \otimes \mathrm{id}_{k_2}) \cdot (\mathrm{id}_{l_1} \otimes \mathrm{ab}_{\vec{u}_1} t'_2) \cdot (\mathrm{id}_{l_1} \otimes \mathrm{p}_{r_1, l_2} \otimes \mathrm{id})) \otimes \mathrm{id}_{k_3}) \cdot \\
&\qquad (\mathrm{id}_{l_1 \otimes l_2} \otimes \mathrm{ab}_{\vec{u}_1 \vec{u}_2} t'_3) \cdot (\mathrm{id}_{l_1 \otimes l_2} \otimes \mathrm{p}_{r_1 \otimes r_2, l_3} \otimes \mathrm{id}) \\
&= t'_R
\end{aligned}$$

The equational proofs involve substantial tedious but routine calculations. The reader may find it useful to construct a diagrammatic equivalent, which, while not formal, provides an intuition of the equality of the terms. Such diagrams representing the terms t'_L and t'_R respectively are given below:





Axiom $\mathbf{P}_1$: $t \otimes \mathrm{id}_\epsilon = t = \mathrm{id}_\epsilon \cdot t$ Straightforward.

Axiom P_2: $t_1 \otimes (t_2 \otimes t_3) = (t_1 \otimes t_2) \otimes t_3$ We shall write the last part of the derivation in each case until subderivations with principal terms t_1, t_2 and t_3. Let $(\vec{u})\vec{\alpha}\langle\vec{v}\rangle = (\vec{u}_1\vec{u}_2\vec{u}_3)\vec{\alpha}_1\vec{\alpha}_2\vec{\alpha}_3\langle\vec{v}_1\vec{v}_2\vec{v}_3\rangle$ and $\vec{z} = \vec{z}_1\vec{z}_2\vec{z}_3$.

Left term t_L: In the following derivation, we also have side conditions

1. $\{\vec{u}_1\} \cap (\mathbf{fn}\,(t_2) \cup \mathbf{fn}\,(t_3) \cup \{\vec{z}_2\vec{z}_3\}) = \emptyset$
2. $\{\vec{u}_2\vec{u}_3\} \cap (\mathbf{fn}\,(t_1) \cup \{\vec{z}_1\}) = \emptyset$
3. $|\vec{u}_1| = r_1$
4. $\{\vec{u}_3\} \cap (\mathbf{fn}\,(t_2) \cup \{\vec{z}_2\}) = \emptyset$
5. $\{\vec{u}_2\} \cap (\mathbf{fn}\,(t_3) \cup \{\vec{z}_3\}) = \emptyset$

6. $|\vec{u}_2| = r_2$

$$\dfrac{\langle\vec{z}_1\rangle \vdash t_1 \xrightarrow{(\vec{u}_1)\vec{\alpha}_1\langle\vec{v}_1\rangle} t_1' \quad \dfrac{\langle\vec{z}_2\rangle \vdash t_2 \xrightarrow{(\vec{u}_2)\vec{\alpha}_2\langle\vec{v}_2\rangle} t_2' \quad \langle\vec{z}_3\rangle \vdash t_3 \xrightarrow{(\vec{u}_3)\vec{\alpha}_3\langle\vec{v}_3\rangle} t_3'}{\langle\vec{z}_2\vec{z}_3\rangle \vdash t_2 \otimes t_3 \xrightarrow{(\vec{u}_2\vec{u}_3)\vec{\alpha}_2\vec{\alpha}_3\langle\vec{v}_2\vec{v}_3\rangle} (t_2' \otimes t_3') \cdot (\mathbf{id}_{l_2} \otimes \mathbf{p}_{r_2,l_3} \otimes \mathbf{id})}\,\otimes}{\langle\vec{z}\rangle \vdash t_1 \otimes (t_2 \otimes t_3) \xrightarrow{(\vec{u})\vec{\alpha}\langle\vec{v}\rangle} t_L'}\,\otimes$$

where $t_L' \equiv (t_1' \otimes t_L'') \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1,l_2\otimes l_3} \otimes \mathbf{id})$
and $t_L'' \equiv (t_2' \otimes t_3') \cdot (\mathbf{id}_{l_2} \otimes \mathbf{p}_{r_2,l_3} \otimes \mathbf{id})$.

Right term t_R: In the following derivation, we also have side conditions

1. $\{\vec{u}_1\} \cap (\mathbf{fn}\,(t_2) \cup \{\vec{z}_2\}) = \emptyset$
2. $\{\vec{u}_2\} \cap (\mathbf{fn}\,(t_1) \cup \{\vec{z}_1\}) = \emptyset$
3. $|\vec{u}_1| = r_1$
4. $\{\vec{u}_3\} \cap (\mathbf{fn}\,(t_1) \cup \mathbf{fn}\,(t_2) \cup \{\vec{z}_1\vec{z}_2\}) = \emptyset$
5. $\{\vec{u}_1\vec{u}_2\} \cap (\mathbf{fn}\,(t_3) \cup \{\vec{z}_3\}) = \emptyset$
6. $|\vec{u}_1\vec{u}_2| = r_1 \otimes r_2$

$$\dfrac{\dfrac{\langle\vec{z}_1\rangle \vdash t_1 \xrightarrow{(\vec{u}_1)\vec{\alpha}_1\langle\vec{v}_1\rangle} t_1' \quad \langle\vec{z}_2\rangle \vdash t_2 \xrightarrow{(\vec{u}_2)\vec{\alpha}_2\langle\vec{v}_2\rangle} t_2'}{\langle\vec{z}_1\vec{z}_2\rangle \vdash t_1 \otimes t_2 \xrightarrow{(\vec{u}_1\vec{u}_2)\vec{\alpha}_1\vec{\alpha}_2\langle\vec{v}_1\vec{v}_2\rangle} (t_1' \otimes t_2') \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1,l_2} \otimes \mathbf{id})}\,\otimes \quad \langle\vec{z}_3\rangle \vdash t_3 \xrightarrow{(\vec{u}_3)\vec{\alpha}_3\langle\vec{v}_3\rangle} t_3'}{\langle\vec{z}\rangle \vdash (t_1 \otimes t_2) \otimes t_3 \xrightarrow{(\vec{u})\vec{\alpha}\langle\vec{v}\rangle} t_R'}\,\otimes$$

where $t_R' \equiv (t_R'' \otimes t_3') \cdot (\mathbf{id}_{l_1\otimes l_2} \otimes \mathbf{p}_{r_1\otimes r_2,l_3} \otimes \mathbf{id})$
and $t_R'' \equiv (t_1' \otimes t_2') \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1,l_2} \otimes \mathbf{id})$.

It is easy to see that the side conditions in each derivation are equivalent. We must now show that $t_L' = t_R'$.

$$\begin{aligned}
t_L' &= (t_1' \otimes ((t_2' \otimes t_3') \cdot (\mathbf{id}_{l_2} \otimes \mathbf{p}_{r_2,l_3} \otimes \mathbf{id}))) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1,l_2\otimes l_3} \otimes \mathbf{id}) \\
&= (t_1' \otimes t_2' \otimes t_3') \cdot (\mathbf{id}_{l_1} \otimes \mathbf{id}_{r_1} \otimes \mathbf{id}l_2 \otimes \mathbf{p}_{r_2,l_3} \otimes \mathbf{id}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1,l_2\otimes l_3} \otimes \mathbf{id}) \\
&= (t_1' \otimes t_2' \otimes t_3') \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1,l_2} \otimes \mathbf{id}) \cdot (\mathbf{id}_{l_1\otimes l_2} \otimes \mathbf{p}_{r_1\otimes r_2,l_3} \otimes \mathbf{id}) \\
&= (((t_1' \otimes t_2') \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1,l_2} \otimes \mathbf{id})) \otimes t_3') \cdot (\mathbf{id}_{l_1\otimes l_2} \otimes \mathbf{p}_{r_1\otimes r_2,l_3} \otimes \mathbf{id}) \\
&= t_R'
\end{aligned}$$

<u>**Axiom PF$_1$: $\mathbf{id} \otimes \mathbf{id} = \mathbf{id}$**</u> Straightforward.

<u>**Axiom PF$_2$: $(s_1 \otimes s_2) \cdot (t_1 \otimes t_2) = (s_1 \cdot t_1) \otimes (s_2 \cdot t_2)$**</u> We shall write the last part of the derivation in each case until subderivations with principal terms s_1, s_2,t_1 and t_2. Let $\vec{z} = \vec{z}_1\vec{z}_2$ and $(\vec{u})\vec{\alpha}\langle\vec{v}\rangle = (\vec{u}_1\vec{u}_2\vec{x}_1\vec{x}_2)\vec{\alpha}_1\vec{\alpha}_2\vec{\beta}_1\vec{\beta}_2\langle\vec{y}_1\vec{y}_2\rangle$ and $(\vec{u'})\vec{\alpha}'\langle\vec{v}\rangle = (\vec{u}_1\vec{x}_1\vec{u}_2\vec{x}_2)\vec{\alpha}_1\vec{\beta}_1\vec{\alpha}_2\vec{\beta}_2\langle\vec{y}_1\vec{y}_2\rangle$.

Left term t_L: In the following derivation, we also have side conditions

1. $\{\vec{u}_1\} \cap (\mathbf{fn}\,(s_2) \cup \{\vec{z}_2\}) = \emptyset$
2. $\{\vec{u}_2\} \cap (\mathbf{fn}\,(s_1) \cup \{\vec{z}_1\}) = \emptyset$
3. $|\vec{u}_i| = r_i$
4. $\{\vec{x}_1\} \cap (\mathbf{fn}\,(t_2) \cup \{\vec{v}_2\}) = \emptyset$
5. $\{\vec{x}_2\} \cap (\mathbf{fn}\,(t_1) \cup \{\vec{v}_1\}) = \emptyset$
6. $|\vec{x}_i| = s_i$
7. $\beta_i : m_i \rightarrow n_i$.

$$\dfrac{\dfrac{\langle\vec{z}_1\rangle \vdash s_1 \xrightarrow{(\vec{u}_1)\vec{\alpha}_1\langle\vec{v}_1\rangle} s_1' \quad \langle\vec{z}_2\rangle \vdash s_2 \xrightarrow{(\vec{u}_2)\vec{\alpha}_2\langle\vec{v}_2\rangle} s_2'}{\langle\vec{z}_1\vec{z}_2\rangle \vdash s_1 \otimes s_2 \xrightarrow{(\vec{u}_1\vec{u}_2)\vec{\alpha}_1\vec{\alpha}_2\langle\vec{v}_1\vec{v}_2\rangle} s_L''} \otimes \quad \dfrac{\langle\vec{v}_1\rangle \vdash t_1 \xrightarrow{(\vec{x}_1)\vec{\beta}_1\langle\vec{y}_1\rangle} t_1' \quad \langle\vec{v}_2\rangle \vdash t_2 \xrightarrow{(\vec{x}_2)\vec{\beta}_2\langle\vec{y}_2\rangle} t_2'}{\langle\vec{v}_1\vec{v}_2\rangle \vdash t_1 \otimes t_2 \xrightarrow{(\vec{x}_1\vec{x}_2)\vec{\beta}_1\vec{\beta}_2\langle\vec{y}_1\vec{y}_2\rangle} t_L''} \otimes}{\langle\vec{z}\rangle \vdash (s_1 \otimes s_2) \cdot (t_1 \otimes t_2) \xrightarrow{(\vec{u})\vec{\alpha}\langle\vec{v}\rangle} t_L'} \circ$$

where $t_L' \equiv (s_L'' \otimes \mathbf{id}_{m_1 \otimes m_2}) \cdot (\mathbf{id}_{l_1 \otimes l_2} \otimes \mathbf{ab}_{\vec{u}_1\vec{u}_2} t_L'') \cdot (\mathbf{id}_{l_1 \otimes l_2} \otimes \mathbf{p}_{r_1 \otimes r_2, n_1 \otimes n_2} \otimes \mathbf{id})$ with $s_L'' \equiv (s_1' \otimes s_2') \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1, l_2} \otimes \mathbf{id})$ and $t_L'' \equiv (t_1' \otimes t_2') \cdot (\mathbf{id}_{n_1} \otimes \mathbf{p}_{s_1, n_2} \otimes \mathbf{id})$.

Right term t_R: In the following derivation, we also have side conditions

1. $\{\vec{x}_1\} \cap (\mathbf{fn}\,(s_1) \cup \{\vec{z}_1\}) = \emptyset$
2. $\{\vec{x}_2\} \cap (\mathbf{fn}\,(s_2) \cup \{\vec{z}_2\}) = \emptyset$
3. $|\vec{u}_i| = r_i$
4. $\{\vec{u}_1\vec{x}_1\} \cap (\mathbf{fn}\,(s_2) \cup \mathbf{fn}\,(t_2) \cup \{\vec{z}_2\}) = \emptyset$
5. $\{\vec{u}_2\vec{x}_2\} \cap (\mathbf{fn}\,(s_1) \cup \mathbf{fn}\,(t_1) \cup \{\vec{z}_1\}) = \emptyset$
6. $|\vec{x}_i| = s_i$

7. $\beta_i : m_i \to n_i$.

$$\dfrac{\dfrac{\langle\vec{z}_1\rangle \vdash s_1 \xrightarrow{(\vec{u}_1)\vec{\alpha}_1\langle\vec{v}_1\rangle} s_1' \quad \langle\vec{v}_1\rangle \vdash t_1 \xrightarrow{(\vec{x}_1)\vec{\beta}_1\langle\vec{y}_1\rangle} t_1'}{\langle\vec{z}_1\rangle \vdash s_1 \cdot t_1 \xrightarrow{(\vec{u}_1\vec{x}_1)\vec{\alpha}_1\vec{\beta}_1\langle\vec{y}_1\rangle} t_a}\circ \quad \dfrac{\langle\vec{z}_2\rangle \vdash s_2 \xrightarrow{(\vec{u}_2)\vec{\alpha}_2\langle\vec{v}_2\rangle} s_2' \quad \langle\vec{v}_2\rangle \vdash t_2 \xrightarrow{(\vec{x}_2)\vec{\beta}_2\langle\vec{y}_2\rangle} t_2'}{\langle\vec{z}_2\rangle \vdash s_2 \cdot t_2 \xrightarrow{(\vec{u}_2\vec{x}_2)\vec{\alpha}_2\vec{\beta}_2\langle\vec{y}_2\rangle} t_b}\circ}{\langle\vec{z}\rangle \vdash (s_1 \cdot t_1) \otimes (s_2 \cdot t_2) \xrightarrow{(\vec{u}')\vec{\alpha}'\langle\vec{v}\rangle} t_R'}\otimes$$

where $t_R' \equiv (t_a \otimes t_b) \cdot (\mathbf{id}_{l_1 \otimes n_1} \otimes \mathbf{p}_{r_1 \otimes s_1, l_2 \otimes n_2} \otimes \mathbf{id})$
with $t_a \equiv (s_1' \otimes \mathbf{id}_{m_1}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}_1} t_1') \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1, n_1} \otimes \mathbf{id})$
and $t_b \equiv (s_2' \otimes \mathbf{id}_{m_2}) \cdot (\mathbf{id}_{l_2} \otimes \mathbf{ab}_{\vec{u}_2} t_2') \cdot (\mathbf{id}_{l_2} \otimes \mathbf{p}_{r_2, n_2} \otimes \mathbf{id})$.

Note that the two derivations do not derive transitions with identical labels. The labels differ by permutations of the binding vectors and the vector of particles constituting their bodies. From each derivation one can construct a derivation for a transition which matches the other. We will just show one of the cases.

$$\dfrac{\dfrac{\langle\vec{z}\rangle \vdash (s_1 \cdot t_1) \otimes (s_2 \cdot t_2) \xrightarrow{(\vec{u}')\vec{\alpha}'\langle\vec{v}\rangle} t_R'}{\langle\vec{z}\rangle \vdash (s_1 \cdot t_1) \otimes (s_2 \cdot t_2) \xrightarrow{(\vec{u})\vec{\alpha}'\langle\vec{v}\rangle} t_R' \cdot (\mathbf{id}_{l_1 \otimes n_1 \otimes l_2 \otimes n_2 \otimes r_1} \otimes \mathbf{p}_{r_2, s_1} \otimes \mathbf{id})}\text{PERM}_1}{\langle\vec{z}\rangle \vdash (s_1 \cdot t_1) \otimes (s_2 \cdot t_2) \xrightarrow{(\vec{u})\vec{\alpha}\langle\vec{v}\rangle} t_R''}\text{PERM}_2$$

where $t_R'' \equiv (\mathbf{id}_{k_1} \otimes \mathbf{p}_{k_2, m_1} \otimes \mathbf{id}) \cdot t_R' \cdot (\mathbf{id}_{l_1 \otimes n_1 \otimes l_2 \otimes n_2 \otimes r_1} \otimes \mathbf{p}_{r_2, s_1} \otimes \mathbf{id}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{n_1, l_2} \otimes \mathbf{id})$.

It is easy to see that the side conditions in each derivation are equivalent. We must now show that $t_L' = t_R''$. We shall do this in several stages. Essentially, the proof involves permuting the subterms s_1', s_2', t_1' and t_2' and simplifying the (often large) terms representing the isomorphisms. For the proofs concerning the rewriting of terms representing isomorphisms we shall not give details: the simplest way to demonstrate these term transformations is through diagrammatic means in the style of Joyal et al.

(1) $(\mathbf{id}_{l_1 \otimes r_1} \otimes \mathbf{p}_{l_2 \otimes r_2, m_1} \otimes \mathbf{id}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1 \otimes m_1, l_2} \otimes \mathbf{id}) \cdot (\mathbf{id}_{l_1 \otimes l_2 \otimes r_1} \otimes \mathbf{p}_{m_1, r_2} \otimes \mathbf{id})$

$$= \quad \mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1, l_2} \otimes \mathbf{id}$$

$$
\begin{aligned}
(2)\ & (\mathbf{id}_{l_1\otimes l_2\otimes r_1}\otimes \mathbf{p}_{r_2,n_1\otimes s_1}\otimes \mathbf{id})\cdot(\mathbf{id}_{l_1}\otimes \mathbf{p}_{l_2,r_1\otimes n_1\otimes s_1}\otimes \mathbf{id})\\
&\cdot(\mathbf{id}_{l_1}\otimes \mathbf{p}_{r_1,n_1}\otimes \mathbf{id}_{s_1\otimes l_2}\otimes \mathbf{p}_{r_2,n_2}\otimes \mathbf{id})\cdot(\mathbf{id}_{l_1\otimes n_1}\otimes \mathbf{p}_{r_1\otimes s_1,l_2\otimes n_2}\otimes \mathbf{id})\\
&\cdot(\mathbf{id}_{l_1\otimes n_1\otimes l_2\otimes n_2\otimes r_2}\otimes \mathbf{p}_{r_2,s_2}\otimes \mathbf{id})\cdot(\mathbf{id}_{l_1}\otimes \mathbf{p}_{n_1,l_2}\otimes \mathbf{id})\\
=\ & (\mathbf{id}_{l_1\otimes l_2\otimes r_1\otimes r_2\otimes n_1}\otimes \mathbf{p}_{s_1,n_2}\otimes \mathbf{id})\cdot(\mathbf{id}_{l_1\otimes l_2}\otimes \mathbf{p}_{r_1\otimes r_2,n_1\otimes n_2}\otimes \mathbf{id})
\end{aligned}
$$

$$
\begin{aligned}
(3)\ & \mathbf{id}_{l_1}\otimes \mathbf{ab}_{\vec{u}_1}t_1'\otimes \mathbf{id}_{l_2}\otimes \mathbf{ab}_{\vec{u}_2}t_2'\\
=\ & (\mathbf{id}_{l_1}\otimes \mathbf{p}_{r_1\otimes m_1,l_2}\otimes \mathbf{id})\cdot(\mathbf{id}_{l_1\otimes l_2}\otimes \mathbf{ab}_{\vec{u}_1}t_1'\otimes \mathbf{ab}_{\vec{u}_2}t_2')\\
&\cdot(\mathbf{id}_{l_1}\otimes \mathbf{p}_{l_2,r_1\otimes n_1\otimes s_1}\otimes \mathbf{id})\\
=\ & (\mathbf{id}_{l_1}\otimes \mathbf{p}_{r_1\otimes m_1,l_2}\otimes \mathbf{id})\cdot(\mathbf{id}_{l_1\otimes l_2\otimes r_1}\otimes \mathbf{p}_{m_1,r_2}\otimes \mathbf{id})\\
&\cdot(\mathbf{id}_{l_1\otimes l_2}\otimes \mathbf{ab}_{\vec{u}_1\vec{u}_2}(t_1'\otimes t_2'))\cdot(\mathbf{id}_{l_1\otimes l_2\otimes r_1}\otimes \mathbf{p}_{r_2,n_1\otimes s_1}\otimes \mathbf{id})\\
&\cdot(\mathbf{id}_{l_1}\otimes \mathbf{p}_{l_2,r_1\otimes n_1\otimes s_1}\otimes \mathbf{id})
\end{aligned}
$$

$$
\begin{aligned}
(4)\ & (\mathbf{id}_{k_1}\otimes \mathbf{p}_{k_2,m_1}\otimes \mathbf{id})\cdot(t_a\otimes t_b)\\
=\ & (\mathbf{id}_{k_1}\otimes \mathbf{p}_{k_2,m_1}\otimes \mathbf{id})\cdot(s_1'\otimes \mathbf{id}_{m_1}\otimes s_2'\otimes \mathbf{id}_{m_2})\\
&\cdot(\mathbf{id}_{l_1}\otimes \mathbf{ab}_{\vec{u}_1}t_1'\otimes \mathbf{id}_{l_2}\otimes \mathbf{ab}_{\vec{u}_2}t_2')\cdot(\mathbf{id}_{l_1}\otimes \mathbf{p}_{r_1,n_1}\otimes \mathbf{id}_{s_1\otimes l_2}\otimes \mathbf{p}_{r_2,n_2}\otimes \mathbf{id})\\
=\ & (\mathbf{id}_{l_1\otimes r_1}\otimes \mathbf{p}_{l_2\otimes r_2,m_1}\otimes \mathbf{id})\cdot(s_1'\otimes s_2'\otimes \mathbf{id}_{m_1\otimes m_2})\\
&\cdot(\mathbf{id}_{l_1}\otimes \mathbf{ab}_{\vec{u}_1}t_1'\otimes \mathbf{id}_{l_2}\otimes \mathbf{ab}_{\vec{u}_2}t_2')\cdot(\mathbf{id}_{l_1}\otimes \mathbf{p}_{r_1,n_1}\otimes \mathbf{id}_{s_1\otimes l_2}\otimes \mathbf{p}_{r_2,n_2}\otimes \mathbf{id})
\end{aligned}
$$

$$
\begin{aligned}
(5)\ & t_R''\\
=\ & (\mathbf{id}_{k_1}\otimes \mathbf{p}_{k_2,m_1}\otimes \mathbf{id})\cdot t_R'\cdot(\mathbf{id}_{l_1\otimes n_1\otimes l_2\otimes n_2\otimes r_1}\otimes \mathbf{p}_{r_2,s_1}\otimes \mathbf{id})\\
&\cdot(\mathbf{id}_{l_1}\otimes \mathbf{p}_{n_1,l_2}\otimes \mathbf{id})\\
=\ & (\mathbf{id}_{k_1}\otimes \mathbf{p}_{k_2,m_1}\otimes \mathbf{id})\cdot(t_a\otimes t_b)\cdot(\mathbf{id}_{l_1\otimes n_1}\otimes \mathbf{p}_{r_1\otimes s_1,l_2\otimes n_2}\otimes \mathbf{id})\\
&\cdot(\mathbf{id}_{l_1\otimes n_1\otimes l_2\otimes n_2\otimes r_1}\otimes \mathbf{p}_{r_2,s_1}\otimes \mathbf{id})\cdot(\mathbf{id}_{l_1}\otimes \mathbf{p}_{n_1,l_2}\otimes \mathbf{id})\\
=\ & (s_1'\otimes s_2'\otimes \mathbf{id})\cdot(\mathbf{id}_{l_1}\otimes \mathbf{p}_{r_1,l_2}\otimes \mathbf{id})\cdot(\mathbf{id}_{l_1\otimes l_2}\otimes \mathbf{ab}_{\vec{u}_1\vec{u}_2}(t_1'\otimes t_2'))\\
&\cdot(\mathbf{id}_{l_1\otimes l_2\otimes r_1\otimes r_2\otimes n_1}\otimes \mathbf{p}_{s_1,n_2}\otimes \mathbf{id})\cdot(\mathbf{id}_{l_1\otimes l_2}\otimes \mathbf{p}_{r_1\otimes r_2,n_1\otimes n_2}\otimes \mathbf{id})\\
=\ & (s_L''\otimes \mathbf{id}_{m_1\otimes m_2})\cdot(\mathbf{id}_{l_1\otimes l_2}\otimes \mathbf{ab}_{\vec{u}_1\vec{u}_2}t_L'')\cdot(\mathbf{id}_{l_1\otimes l_2}\otimes \mathbf{p}_{r_1\otimes r_2,n_1\otimes n_2}\otimes \mathbf{id})\\
=\ & t_L'
\end{aligned}
$$

Axiom AF$_1$: $\mathbf{ab}_x\mathbf{id} = \mathbf{id}$ Straightforward.

Axiom AF$_2$: $\mathbf{ab}_x(t_1\cdot t_2) = \mathbf{ab}_x t_1\cdot \mathbf{ab}_x t_2$ We shall write the last part of the derivation in each case until subderivations with principal terms t_1 and t_2. Let $(\vec{u})\vec{\alpha}\langle\vec{v}\rangle = (\vec{u}_1\vec{u}_2)\vec{\alpha}_1\vec{\alpha}_2\langle y\vec{v}_2\rangle$.

Left term t_L: In the following derivation, we also have side conditions

1. $y \notin \{\vec{u}_1\vec{u}_2\}$
2. $\{\vec{u}_2\} \cap (\mathbf{fn}(\{y/x\}t_1) \cup \{\vec{z}\}) = \emptyset$
3. $|\vec{u}_1| = r$

$$\dfrac{\dfrac{\langle\vec{z}\rangle \vdash \{y/x\}t_1 \xrightarrow{(\vec{u}_1)\vec{\alpha}_1\langle\vec{v}_1\rangle} t_1' \quad \langle\vec{v}_1\rangle \vdash \{y/x\}t_2 \xrightarrow{(\vec{u}_2)\vec{\alpha}_2\langle\vec{v}_2\rangle} t_2'}{\langle\vec{z}\rangle \vdash \{y/x\}t_1 \cdot \{y/x\}t_2 \xrightarrow{(\vec{u}_1\vec{u}_2)\vec{\alpha}_1\vec{\alpha}_2\langle\vec{v}_2\rangle} (t_1' \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}_1} t_2') \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r,l_2} \otimes \mathbf{id})}\,\circ}{\langle y\vec{z}\rangle \vdash \mathbf{ab}_x(t_1 \cdot t_2) \xrightarrow{(\vec{u})\vec{\alpha}\langle\vec{v}\rangle} (t_1' \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}_1} t_2') \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r,l_2} \otimes \mathbf{id})}\,\mathbf{ab}_x$$

Right term t_R: In the following derivation, we also have side conditions

1. $y \notin \{\vec{u}_1\}$
2. $y \notin \{\vec{u}_2\}$
3. $\{\vec{u}_2\} \cap (\mathbf{fn}(\{y/x\}t_1) \cup \{\vec{z}\}) = \emptyset$
4. $|\vec{u}_1| = r$

$$\dfrac{\dfrac{\langle\vec{z}\rangle \vdash \{y/x\}t_1 \xrightarrow{(\vec{u}_1)\vec{\alpha}_1\langle\vec{v}_1\rangle} t_1'}{\langle y\vec{z}\rangle \vdash \mathbf{ab}_x t_1 \xrightarrow{(\vec{u}_1)\vec{\alpha}_1\langle y\vec{v}_1\rangle} t_1'}\,\mathbf{ab}_x \quad \dfrac{\langle\vec{v}_1\rangle \vdash \{y/x\}t_2 \xrightarrow{(\vec{u}_2)\vec{\alpha}_2\langle\vec{v}_2\rangle} t_2'}{\langle y\vec{v}_1\rangle \vdash \mathbf{ab}_x t_2 \xrightarrow{(\vec{u}_2)\vec{\alpha}_2\langle y\vec{v}_2\rangle} t_2'}}{\langle y\vec{z}\rangle \vdash \mathbf{ab}_x t_1 \cdot \mathbf{ab}_x t_2 \xrightarrow{(\vec{u})\vec{\alpha}\langle\vec{v}\rangle} (t_1' \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}_1} t_2') \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r,l_2} \otimes \mathbf{id})}\,\circ$$

It is easy to see that the side conditions in each derivation are equivalent.

Axiom γ: $(x)t = \omega \otimes t \quad (x \notin \mathbf{fn}(t))$ Straightforward.

Axiom δ: $(x)(\langle x\rangle \otimes \mathbf{id}) = \mathsf{id}$ Straightforward.

Axiom ζ: $(t_1 \otimes t_2) \cdot \mathbf{p}_{n_1,n_2} = \mathbf{p}_{m_1,m_2} \cdot (t_2 \otimes t_1) \quad (t_i : m_i \rightarrow n_i)$ We shall write the last part of the derivation in each case until subderivations with principal terms t_1 and t_2. Let $(\vec{u})\vec{\alpha}\langle\vec{v}\rangle = (\vec{u}_1\vec{u}_2)\vec{\alpha}_1\vec{\alpha}_2\langle\vec{v}_1\vec{v}_2\rangle$, $(\vec{u})\vec{\alpha}\langle\vec{v}'\rangle = (\vec{u}_2\vec{u}_1)\vec{\alpha}_2\vec{\alpha}_1\langle\vec{v}_2\vec{v}_1\rangle$, and $\vec{z} = \vec{z}_1\vec{z}_2$.

Left term t_L: In the following derivation, we also have side conditions

1. $\{\vec{u}_1\} \cap (\mathbf{fn}\,(t_2) \cup \{\vec{z}_2\}) = \emptyset$
2. $\{\vec{u}_2\} \cap (\mathbf{fn}\,(t_1) \cup \{\vec{z}_1\}) = \emptyset$
3. $|\vec{u}_2| = r_2$

$$\dfrac{\dfrac{\langle \vec{z}_1 \rangle \vdash t_1 \xrightarrow{(\vec{u}_1)\vec{\alpha}_1\langle\vec{v}_1\rangle} t_1' \quad \langle \vec{z}_2 \rangle \vdash t_2 \xrightarrow{(\vec{u}_2)\vec{\alpha}_2\langle\vec{v}_2\rangle} t_2'}{\langle \vec{z}_1\vec{z}_2 \rangle \vdash t_1 \otimes t_2 \xrightarrow{(\vec{u}_1\vec{u}_2)\vec{\alpha}_1\vec{\alpha}_2\langle\vec{v}_1\vec{v}_2\rangle} (t_1' \otimes t_2') \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1,l_2} \otimes \mathbf{id})}\otimes \quad \langle \vec{v}_1\vec{v}_2 \rangle \vdash \mathbf{p}_{n_1,n_2} \xrightarrow{\langle\vec{v}_2\vec{v}_1\rangle}_{=} \mathbf{id}_\epsilon}{\langle \vec{z} \rangle \vdash (t_1 \otimes t_2) \cdot \mathbf{p}_{n_1,n_2} \xrightarrow{(\vec{u})\vec{\alpha}\langle\vec{v}'\rangle}_{=} (t_1' \otimes t_2') \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1,l_2} \otimes \mathbf{id})}\circ$$

Right term t_R: In the following derivation, we also have side conditions

1. $\{\vec{u}_1\} \cap (\mathbf{fn}\,(t_2) \cup \{\vec{z}_2\}) = \emptyset$
2. $\{\vec{u}_2\} \cap (\mathbf{fn}\,(t_1) \cup \{\vec{z}_1\}) = \emptyset$
3. $|\vec{u}_2| = r_2$

$$\dfrac{\langle \vec{z}_1\vec{z}_2 \rangle \vdash \mathbf{p}_{m_2,m_1} \xrightarrow{\langle\vec{z}_2\vec{z}_1\rangle}_{=} \mathbf{id}_\epsilon \quad \dfrac{\langle \vec{z}_2 \rangle \vdash t_2 \xrightarrow{(\vec{u}_2)\vec{\alpha}_2\langle\vec{v}_2\rangle} t_2' \quad \langle \vec{z}_1 \rangle \vdash t_1 \xrightarrow{(\vec{u}_1)\vec{\alpha}_1\langle\vec{v}_1\rangle} t_1'}{\langle \vec{z}_1\vec{z}_2 \rangle \vdash t_2 \otimes t_1 \xrightarrow{(\vec{u}_2\vec{u}_1)\vec{\alpha}_2\vec{\alpha}_1\langle\vec{v}_2\vec{v}_1\rangle} (t_2' \otimes t_1') \cdot (\mathbf{id}_{l_2} \otimes \mathbf{p}_{r_2,l_1} \otimes \mathbf{id})}\otimes}{\langle \vec{z} \rangle \vdash \mathbf{p}_{m_2,m_1} \cdot (t_2 \otimes t_1) \xrightarrow{(\vec{u}')\vec{\alpha}'\langle\vec{v}'\rangle}_{=} (t_2' \otimes t_1') \cdot (\mathbf{id}_{l_2} \otimes \mathbf{p}_{r_2,l_1} \otimes \mathbf{id})}\circ$$

Note that the two derivations do not derive transitions with identical labels. The labels differ by permutations of the binding vectors and the vector of particles constituting their bodies. From each derivation one can construct a derivation for a transition which matches the other. We will just show one of the cases.

$$\dfrac{\dfrac{\langle \vec{z} \rangle \vdash \mathbf{p}_{m_2,m_1} \cdot (t_2 \otimes t_1) \xrightarrow{(\vec{u}')\vec{\alpha}'\langle\vec{v}'\rangle} (t_2' \otimes t_1') \cdot (\mathbf{id}_{l_2} \otimes \mathbf{p}_{r_2,l_1} \otimes \mathbf{id})}{\langle \vec{z} \rangle \vdash \mathbf{p}_{m_2,m_1} \cdot (t_2 \otimes t_1) \xrightarrow{(\vec{u}_1\vec{u}_2)\vec{\alpha}_2\vec{\alpha}_1\langle\vec{v}_2\vec{v}_1\rangle} (t_2' \otimes t_1') \cdot (\mathbf{id}_{l_2} \otimes \mathbf{p}_{r_2,l_1} \otimes \mathbf{id}) \cdot (\mathbf{id}_{l_2 \otimes l_1} \otimes \mathbf{p}_{r_1,r_2} \otimes \mathbf{id})}\,\text{PERM}_1}{\langle \vec{z} \rangle \vdash \mathbf{p}_{m_2,m_1} \cdot (t_2 \otimes t_1) \xrightarrow{(\vec{u})\vec{\alpha}\langle\vec{v}'\rangle} t_R'}\,\text{PERM}_2$$

where $t_R' \equiv (\mathbf{p}_{k_1,k_2} \otimes \mathbf{id}_\epsilon) \cdot t_R'' \cdot (\mathbf{id}_\epsilon \otimes \mathbf{p}_{l_2,l_1} \otimes \mathbf{id})$ and $t_R'' = (t_2' \otimes t_1') \cdot (\mathbf{id}_{l_2} \otimes \mathbf{p}_{r_2,l_1} \otimes \mathbf{id})$.

It is easy to see that the side conditions in each derivation are equivalent. We must now show that $t_L' = t_R'$.

$$
\begin{aligned}
t'_R &= \mathbf{p}_{k_1,k_2} \cdot (t'_2 \otimes t'_1) \cdot (\mathbf{id}_{l_2} \otimes \mathbf{p}_{r_2,l_1} \otimes \mathbf{id}) \cdot (\mathbf{p}_{l_2,l_1} \otimes \mathbf{id}) \\
&= (t'_1 \otimes t'_2) \cdot \mathbf{p}_{l_1 \otimes r_1, l_2 \otimes r_2} \cdot (\mathbf{id}_{l_2} \otimes \mathbf{p}_{r_2,l_1} \otimes \mathbf{id}) \cdot (\mathbf{p}_{l_2,l_1} \otimes \mathbf{id}) \\
&= (t'_1 \otimes t'_2) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1,l_2} \otimes \mathbf{id}) \\
&= t'_L
\end{aligned}
$$

Axiom σ: $(\langle y \rangle \otimes \mathbf{id}) \cdot (x)t = \{y/x\}t$ Straightforward.

Axiom ρ_1: $\mathbf{id}_p = \uparrow_p \mathbf{p}_{p,p}$ Straightforward.

Axiom ρ_2: $\uparrow_p t \otimes \mathbf{id} = \uparrow_p (t \otimes \mathbf{id})$ Straightforward.

Axiom ρ_3: $\uparrow_p t_1 \cdot t_2 = \uparrow_p (t_1 \cdot (\mathbf{id}_p \otimes t_2))$ Let $(\vec{u})\vec{\alpha}\langle\vec{v}\rangle = (\vec{u}_1\vec{u}_2)\vec{\alpha}_1\vec{\alpha}_2\langle\vec{v}_2\rangle$.

Case For $y \notin (\mathbf{fn}\,(t_1) \cup \{\vec{z}\})$ we have the derivation $\langle y\vec{z}\rangle \vdash t_1 \xrightarrow{(\vec{u}_1)\vec{\alpha}_1\langle y\vec{v}_1\rangle} t'_1$:

Left term t_L: In the following derivation, we also have side conditions

1. $\{y\vec{u}_1\} \cap \mathbf{fn}\,(t_2) = \emptyset$
2. $\{\vec{u}_2\} \cap (\mathbf{fn}\,(t_1) \cup \{\vec{z}\}) = \emptyset$
3. $|y\vec{u}_1| = p \otimes r_1$
4. $y \notin \mathbf{fn}\,(t_1) \cup \{\vec{z}\vec{u}_1\}$.

$$
\dfrac{\dfrac{\langle y\vec{z}\rangle \vdash t_1 \xrightarrow{(\vec{u}_1)\vec{\alpha}_1\langle y\vec{v}_1\rangle} t'_1}{\langle\vec{z}\rangle \vdash \uparrow_p t_1 \xrightarrow{(y\vec{u}_1)\vec{\alpha}_1\langle\vec{v}_1\rangle} (\nu y)(t'_1 \cdot (\mathbf{id}_{l_1} \otimes \langle y\rangle \otimes \mathbf{id}))}\uparrow_1 \qquad \langle\vec{v}_1\rangle \vdash t_2 \xrightarrow{(\vec{u}_2)\vec{\alpha}_2\langle\vec{v}_2\rangle} t'_2}{\langle\vec{z}\rangle \vdash \uparrow_p t_1 \cdot t_2 \xrightarrow{(y\vec{u})\vec{\alpha}\langle\vec{v}_2\rangle} t'_L}\circ
$$

where $t'_L \equiv (t''_L \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{ab}_{y\vec{u}_1} t'_2) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{p\otimes r_1, l_2} \otimes \mathbf{id})$
and $t''_L \equiv (\nu y)(t'_1 \cdot (\mathbf{id}_{l_1} \otimes \langle y\rangle \otimes \mathbf{id}))$.

Right term t_R : In the following derivation, we also have side conditions

1. $\{\vec{u}_1\} \cap \mathbf{fn}\,(t_2) = \emptyset$
2. $\{\vec{u}_2\} \cap (\mathbf{fn}\,(t_1) \cup \{\vec{z}\}) = \emptyset$

3. $|\vec{u}_1| = r_1$

4. $y' \notin \mathbf{fn}(t_1) \cup \mathbf{fn}(t_2) \cup \{\vec{z}\vec{u}_1\vec{u}_2\}$.

Note that one of the side conditions requires that $y' \notin \mathbf{fn}(t_2)$. Therefore, we cannot simply rely on the name y used in the derivation (involving t_L) above. Instead, we shall choose such a $y' \notin \mathbf{fn}(t_1) \cup \mathbf{fn}(t_2) \cup \{\vec{u}_1\vec{u}_2\vec{z}\}$. We shall then use the substitution lemma to establish the required correspondence between the subderivations involving y and y'. In what follows, let $\sigma = \{y'/y\}$.

$$\dfrac{\dfrac{\dfrac{\langle y\vec{z}\rangle \vdash t_1 \xrightarrow{(\vec{u}_1)\vec{\alpha}_1\langle y\vec{v}_1\rangle} t_1'}{\langle y'\vec{z}\rangle \vdash t_1 \xrightarrow{(\vec{u}_1)\sigma\vec{\alpha}_1\langle y'\sigma\vec{v}_1\rangle}_{=} \sigma t_1'} \qquad \dfrac{\langle y'\rangle \vdash \mathbf{id}_p \xrightarrow{\langle y'\rangle} \mathbf{id}_\epsilon \quad \dfrac{\langle \vec{v}_1\rangle \vdash t_2 \xrightarrow{(\vec{u}_2)\vec{\alpha}_2\langle \vec{v}_2\rangle} t_2'}{\langle \sigma\vec{v}_1\rangle \vdash \sigma t_2 \xrightarrow{(\vec{u}_2)\sigma\vec{\alpha}_2\langle \sigma\vec{v}_2\rangle}_{=} \sigma t_2'}}{\langle y'\sigma\vec{v}_1\rangle \vdash \mathbf{id}_p \otimes t_2 \xrightarrow{(\vec{u}_2)\sigma\vec{\alpha}_2\langle y'\sigma\vec{v}_2\rangle}_{=} \sigma t_2'}\otimes}{\langle y'\vec{z}\rangle \vdash t_1 \cdot (\mathbf{id}_p \otimes t_2) \xrightarrow{(\vec{u})\sigma\vec{\alpha}\langle y'\sigma\vec{v}_2\rangle}_{=} (\sigma t_1' \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}_1}\sigma t_2') \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1,l_2} \otimes \mathbf{id})}\circ}{\langle \vec{z}\rangle \vdash \uparrow_p(t_1 \cdot (\mathbf{id}_p \otimes t_2)) \xrightarrow{(y'\vec{u})\sigma\vec{\alpha}\langle \sigma\vec{v}_2\rangle}_{=} t_R'}\uparrow_1$$

where $t_R' = (\nu y')(t_R'' \cdot (\mathbf{id}_{l_1\otimes l_2} \otimes \langle y'\rangle \otimes \mathbf{id}))$ and $t_R'' \equiv (\sigma t_1' \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}_1}\sigma t_2') \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1,l_2} \otimes \mathbf{id})$.

First, note that the labels of the derived transitions for t_L and t_R are indistinguishable up to alphaconversion. We shall now prove the equality of the residuals $t_L' = t_R'$.

$$\begin{aligned}
t_R' &= (\nu y')((\sigma t_1' \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}_1}\sigma t_2') \\
&\qquad\qquad \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1,l_2} \otimes \mathbf{id}) \cdot (\mathbf{id}_{l_1\otimes l_2} \otimes \langle y'\rangle \otimes \mathbf{id})) \\
&= (\nu y)((t_1' \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}_1} t_2') \\
&\qquad\qquad \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1,l_2} \otimes \mathbf{id}) \cdot (\mathbf{id}_{l_1\otimes l_2} \otimes \langle y\rangle \otimes \mathbf{id})) \\
&= (\nu y)((t_1' \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}_1} t_2') \\
&\qquad\qquad \cdot (\mathbf{id}_{l_1} \otimes \langle y\rangle \otimes \mathbf{id}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{p\otimes r_1,l_2} \otimes \mathbf{id})) \\
&= (\nu y)((t_1' \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes \langle y\rangle \otimes \mathbf{ab}_{\vec{u}_1} t_2') \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{p\otimes r_1,l_2} \otimes \mathbf{id}))
\end{aligned}$$

$$
\begin{aligned}
&= (\nu y)((t_1' \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes (((y) \otimes \mathbf{id}) \cdot (y)((y) \otimes \mathbf{ab}_{\vec{u}_1} t_2'))) \\
&\qquad\qquad \cdot (\mathbf{id}_{l_1} \otimes \langle y \rangle \otimes \mathbf{id}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{p \otimes r_1, l_2} \otimes \mathbf{id})) \\
&= (\nu y)((t_1' \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes \langle y \rangle \otimes \mathbf{id}) \\
&\qquad\qquad \cdot (\mathbf{id}_{l_1} \otimes \mathbf{ab}_{y\vec{u}_1} t_2') \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{p \otimes r_1, l_2} \otimes \mathbf{id})) \\
&= (\nu y)((t_1' \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes \langle y \rangle \otimes \mathbf{id})) \\
&\qquad\qquad \cdot (\mathbf{id}_{l_1} \otimes \mathbf{ab}_{y\vec{u}_1} t_2') \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{p \otimes r_1, l_2} \otimes \mathbf{id}) \\
&= (\nu y)((t_1' \cdot (\mathbf{id}_{l_1} \otimes \langle y \rangle)) \otimes \mathbf{id}_{k_2}) \otimes \mathbf{id})) \\
&\qquad\qquad \cdot (\mathbf{id}_{l_1} \otimes \mathbf{ab}_{y\vec{u}_1} t_2') \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{p \otimes r_1, l_2} \otimes \mathbf{id}) \\
&= t_L'
\end{aligned}
$$

The above derivation shows how a matching derivation for t_R can be obtained from a derivation for t_L. We argue that obtaining a matching derivation for t_L from a derivation for t_R is simpler since the side conditions in the derivation for t_R (involving some $y \notin \mathbf{fn}(t_1) \cup \mathbf{fn}(t_2) \cup \{\vec{z}\}$) are stronger than those required for t_L.

Case For $y \notin \mathbf{fn}(t_1) \cup \{\vec{z}\vec{u}_1\}$ we have the derivation $\langle y\vec{z} \rangle \vdash t_1 \xrightarrow{(\vec{u}_1)\vec{\alpha}_1\langle w\vec{v}_1 \rangle} t_1'$ with $w \neq y$:

Left term t_L: Let $\sigma' = \{{}^{w}/_{y}\}$. In the following derivation, we also have side conditions

1. $\{\vec{u}_1\} \cap \mathbf{fn}(t_2) = \emptyset$
2. $\{\vec{u}_2\} \cap (\mathbf{fn}(t_1) \cup \{\vec{z}\}) = \emptyset$
3. $|\vec{u}_1| = r_1$
4. $y \notin \mathbf{fn}(t_1) \cup \{\vec{z}\vec{u}_1\}$
5. $w \neq y$.

$$
\dfrac{\dfrac{\langle y\vec{z} \rangle \vdash t_1 \xrightarrow{(\vec{u}_1)\vec{\alpha}_1\langle w\vec{v}_1 \rangle} t_1'}{\langle \vec{z} \rangle \vdash \uparrow_p t_1 \xrightarrow{(\vec{u}_1)\vec{\sigma}'\alpha_1\langle \vec{v}_1 \rangle} \uparrow(y)(t_1' \cdot (\mathbf{id}_{l_1} \otimes (\vec{u}_1)\langle w\vec{u}_1 \rangle)) \cdot (\mathbf{p}_{l_1,p} \otimes \mathbf{id}))} \uparrow_2 \quad \langle \sigma'\vec{v}_1 \rangle \vdash t_2 \xrightarrow{(\vec{u}_2)\vec{\alpha}_2\langle \vec{v}_2 \rangle} t_2'}{\langle \vec{z} \rangle \vdash \uparrow_p t_1 \cdot t_2 \xrightarrow{(\vec{u})(\sigma'\vec{\alpha}_1)\vec{\alpha}_2\langle \vec{v}_2 \rangle} t_L'} \circ
$$

where $t_L' \equiv (t_L'' \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}_1} t_2') \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1,l_2} \otimes \mathbf{id})$
and $t_L'' \equiv \uparrow(y)(t_1' \cdot (\mathbf{id}_{l_1} \otimes (\vec{u}_1)\langle w\vec{u}_1 \rangle)) \cdot (\mathbf{p}_{l_1,p} \otimes \mathbf{id}))$.

Right term t_R: In the following derivation, we also have side conditions

1. $\{y'\vec{u}_1\} \cap \mathbf{fn}\,(t_2) = \emptyset$
2. $\{\vec{u}_2\} \cap (\mathbf{fn}\,(t_1) \cup \{y'\vec{z}\}) = \emptyset$
3. $|\vec{u}_1| = r_1$
4. $y' \notin \mathbf{fn}\,(t_1) \cup \mathbf{fn}\,(t_2) \cup \{\vec{z}\vec{u}_1\vec{u}_2\}$
5. $w \notin \{\vec{u}_2\}$
6. $y' \neq w$.

Note that one of the side conditions requires that $y' \notin \mathbf{fn}\,(t_2) \cup \{\vec{u}_1\}$. Therefore, we cannot simply rely on the name y used in the derivation (involving t_L) above. Instead, we shall choose such a $y' \notin \mathbf{fn}\,(t_1) \cup \mathbf{fn}\,(t_2) \cup \{\vec{u}_1\vec{u}_2\vec{z}\}$. Note that by the free names lemma, $\{\vec{v}_1\vec{v}_2\} \subseteq \mathbf{fn}\,(t_1) \cup \mathbf{fn}\,(t_2) \cup \{vecz\}$ and hence $y' \notin \{\vec{v}_1\vec{v}_2\}$. We shall then use the substitution lemma to establish the required correspondence between the subderivations involving y and y'. In what follows, let $\sigma = \{{}^{y'}\!/\!_{y}\}$.

Now, by the above conditions, $\{{}^{w}\!/\!_{y}\}\vec{v}_1 = \{{}^{w}\!/\!_{y'}\}(\{{}^{y'}\!/\!_{y}\}\vec{v}_1)$ and $\{{}^{w}\!/\!_{y'}\}t_2 \equiv t_2$. Hence, by the substitution lemma, there exist $\vec{\beta}_2$, $\vec{w}_2$ and t_2'' such that

$$\langle \{{}^{w}\!/\!_{y'}\}(\{{}^{y'}\!/\!_{y}\}\vec{v}_1)\rangle \vdash \{{}^{w}\!/\!_{y'}\}t_2 \xrightarrow{(\vec{u}_2)\vec{\alpha}_2\langle\vec{v}_2\rangle} t_2' \quad \Rightarrow \quad \langle \{{}^{y'}\!/\!_{y}\}\vec{v}_1\rangle \vdash t_2 \xrightarrow{(\vec{u}_2)\vec{\beta}_2\langle\vec{w}_2\rangle} t_2''$$

where $\{{}^{w}\!/\!_{y'}\}\vec{\beta}_2 = \vec{\alpha}_2$, $\{{}^{w}\!/\!_{y'}\}\vec{w}_2 = \vec{v}_2$ and $\{{}^{w}\!/\!_{y'}\}t_2'' = t_2'$.

$$\dfrac{\dfrac{\dfrac{\langle y\vec{z}\rangle \vdash t_1 \xrightarrow{(\vec{u}_1)\vec{\alpha}_1\langle w\vec{v}_1\rangle} t_1'}{\langle y'\vec{z}\rangle \vdash t_1 \xrightarrow{(\vec{u}_1)\sigma\vec{\alpha}_1\langle w\sigma\vec{v}_1\rangle}{}_{=}\sigma t_1'} \qquad \dfrac{\langle w\rangle \vdash \mathbf{id}_p \xrightarrow{\langle w\rangle} \mathbf{id}_\epsilon \quad \dfrac{\langle \sigma'\vec{v}_1\rangle \vdash t_2 \xrightarrow{(\vec{u}_2)\vec{\alpha}_2\langle\vec{v}_2\rangle} t_2'}{\langle \sigma\vec{v}_1\rangle \vdash t_2 \xrightarrow{(\vec{u}_2)\vec{\beta}_2\langle\vec{w}_2\rangle}{}_{=} t_2''}}{\langle w\sigma\vec{v}_1\rangle \vdash \mathbf{id}_p \otimes t_2 \xrightarrow{(\vec{u}_2)\vec{\beta}_2\langle w\vec{w}_2\rangle}{}_{=} t_2''}\otimes}{\langle y'\vec{z}\rangle \vdash t_1 \cdot (\mathbf{id}_p \otimes t_2) \xrightarrow{(\vec{u})(\sigma\vec{\alpha}_1)\vec{\beta}_2\langle w\vec{w}_2\rangle}{}_{=} (\sigma t_1' \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}_1} t_2'') \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1,l_2} \otimes \mathbf{id})}\circ}{\langle\vec{z}\rangle \vdash \uparrow_p(t_1 \cdot (\mathbf{id}_p \otimes t_2)) \xrightarrow{(\vec{u})(\sigma'\vec{\alpha}_1)\vec{\alpha}_2\langle\vec{v}_2\rangle}{}_{=} t_R'}\uparrow_2$$

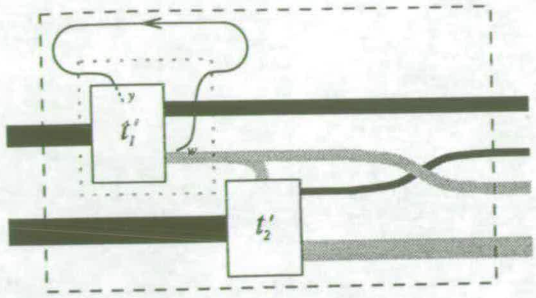
where $t_R' = \uparrow(y')(t_R'' \cdot (\mathbf{id}_{l_1 \otimes l_2} \otimes (\vec{u}_1\vec{u}_2)\langle w\vec{u}_1\vec{u}_2\rangle) \cdot (\mathbf{p}_{l_1 \otimes l_2, p} \otimes \mathbf{id}))$
and $t_R'' = (\sigma t_1' \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}_1}\sigma t_2'') \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1,l_2} \otimes \mathbf{id})$.

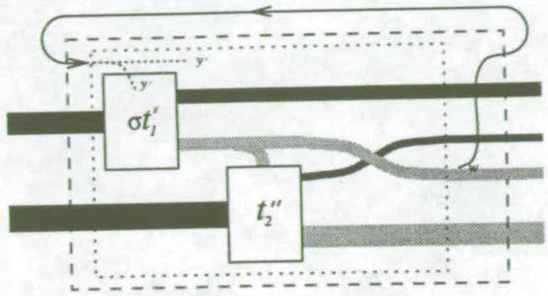
It is easy to see that the side conditions in each derivation are equivalent. We must now show that $t_L' = t_R'$.

$$
\begin{aligned}
t'_L = \;& (\uparrow(y)(t'_1 \cdot (\mathrm{id}_{l_1} \otimes (\vec{u}_1)\langle w\vec{u}_1\rangle) \cdot (\mathrm{p}_{l_1,p} \otimes \mathrm{id})) \otimes \mathrm{id}_{k_2}) \cdot (\mathrm{id}_{l_1} \otimes \mathrm{ab}_{\vec{u}_1} t'_2) \\
& \cdot(\mathrm{id}_{l_1} \otimes \mathrm{p}_{r_1,l_2} \otimes \mathrm{id}) \\
= \;& (\uparrow(y')(\{v'/v\}t'_1 \cdot (\mathrm{id}_{l_1} \otimes (\vec{u}_1)\langle w\vec{u}_1\rangle) \cdot (\mathrm{p}_{l_1,p} \otimes \mathrm{id})) \otimes \mathrm{id}_{k_2}) \cdot (\mathrm{id}_{l_1} \otimes \mathrm{ab}_{\vec{u}_1} t'_2) \\
& \cdot(\mathrm{id}_{l_1} \otimes \mathrm{p}_{r_1,l_2} \otimes \mathrm{id}) \\
= \;& (\uparrow(y')((\{v'/v\}t'_1 \otimes \mathrm{id}_{k_2}) \cdot (\mathrm{id}_{l_1} \otimes (\vec{u}_1)\langle w\vec{u}_1\rangle \otimes \mathrm{id}_{k_2}) \\
& \cdot(\mathrm{p}_{l_1,p} \otimes \mathrm{id}))) \cdot (\mathrm{id}_{l_1} \otimes \mathrm{ab}_{\vec{u}_1} t'_2) \cdot (\mathrm{id}_{l_1} \otimes \mathrm{p}_{r_1,l_2} \otimes \mathrm{id}) \\
= \;& \uparrow(y')((\{v'/v\}t'_1 \otimes \mathrm{id}_{k_2}) \cdot (\mathrm{id}_{l_1} \otimes (\vec{u}_1)\langle w\vec{u}_1\rangle \otimes \mathrm{id}_{k_2}) \\
& \cdot(\mathrm{p}_{l_1,p} \otimes \mathrm{id}) \cdot (\mathrm{id}_{p\otimes l_1} \otimes \mathrm{ab}_{\vec{u}_1} t'_2) \cdot (\mathrm{id}_{p\otimes l_1} \otimes \mathrm{p}_{r_1,l_2} \otimes \mathrm{id}))
\end{aligned}
$$

$$
\begin{aligned}
= \;& \uparrow(y')((\{v'/v\}t'_1 \otimes \mathrm{id}_{k_2}) \cdot (\mathrm{p}_{l_1,r_1} \otimes \mathrm{id}) \cdot (\vec{u}_1)((\mathrm{id}_{l_1} \otimes \langle w\vec{u}_1\rangle \otimes \mathrm{id}_{k_2}) \\
& \cdot(\mathrm{p}_{l_1,p} \otimes \mathrm{id}) \cdot (\mathrm{id}_{p\otimes l_1} \otimes \mathrm{ab}_{\vec{u}_1} t'_2) \cdot (\mathrm{id}_{p\otimes l_1} \otimes \mathrm{p}_{r_1,l_2} \otimes \mathrm{id}))) \\
= \;& \uparrow(y')((\{v'/v\}t'_1 \otimes \mathrm{id}_{k_2}) \cdot (\mathrm{p}_{l_1,r_1} \otimes \mathrm{id}) \cdot (\vec{u}_1)(((\langle w\rangle \otimes \mathrm{id}_{l_1} \otimes \langle \vec{u}_1\rangle \otimes \mathrm{id}_{k_2}) \\
& \cdot(\mathrm{id}_{p\otimes l_1} \otimes \mathrm{ab}_{\vec{u}_1} t'_2) \cdot (\mathrm{id}_{p\otimes l_1} \otimes \mathrm{p}_{r_1,l_2} \otimes \mathrm{id}))) \\
= \;& \uparrow(y')((\{v'/v\}t'_1 \otimes \mathrm{id}_{k_2}) \cdot (\mathrm{p}_{l_1,r_1} \otimes \mathrm{id}) \\
& \cdot(\vec{u}_1)(\langle w\rangle \otimes \mathrm{id}_{l_1} \otimes (((\langle \vec{u}_1\rangle \otimes t'_2) \cdot (\mathrm{p}_{r_1,l_2} \otimes \mathrm{id}))) \\
= \;& \uparrow(y')((\{v'/v\}t'_1 \otimes \mathrm{id}_{k_2}) \cdot (\mathrm{p}_{l_1,r_1} \otimes \mathrm{id}) \\
& \cdot(\vec{u}_1)(\langle w\rangle \otimes (\mathrm{id}_{l_1} \otimes (t'_2 \cdot (\mathrm{id}_{l_2} \otimes \langle \vec{u}_1\rangle \otimes \mathrm{id}))))) \\
= \;& \uparrow(y')((\{v'/v\}t'_1 \otimes \mathrm{id}_{k_2}) \cdot (\mathrm{p}_{l_1,r_1} \otimes \mathrm{id}) \\
& \cdot(\vec{u}_1)(\langle w\rangle \otimes (\mathrm{id}_{l_1} \otimes (\{w/v'\}t'_2 \cdot (\mathrm{id}_{l_2} \otimes \langle \vec{u}_1\rangle \otimes \mathrm{id}))))) && \text{A.1} \\
= \;& \uparrow(y')((\{v'/v\}t'_1 \otimes \mathrm{id}_{k_2}) \cdot (\mathrm{p}_{l_1,r_1} \otimes \mathrm{id}) \\
& \cdot(\vec{u}_1)(\langle w\rangle \otimes (\mathrm{id}_{l_1} \otimes (\{w/v'\}\{w/v'\}t''_2 \cdot (\mathrm{id}_{l_2} \otimes \langle \vec{u}_1\rangle \otimes \mathrm{id}))))) \\
= \;& \uparrow(y')((\{v'/v\}t'_1 \otimes \mathrm{id}_{k_2}) \cdot (\mathrm{p}_{l_1,r_1} \otimes \mathrm{id}) \\
& \cdot(\vec{u}_1)(\langle w\rangle \otimes (\mathrm{id}_{l_1} \otimes (\{w/v'\}t''_2 \cdot (\mathrm{id}_{l_2} \otimes \langle \vec{u}_1\rangle \otimes \mathrm{id}))))) \\
= \;& \uparrow(y')((\{v'/v\}t'_1 \otimes \mathrm{id}_{k_2}) \cdot (\mathrm{p}_{l_1,r_1} \otimes \mathrm{id}) \\
& \cdot(\vec{u}_1)(\langle w\rangle \otimes (\mathrm{id}_{l_1} \otimes (t''_2 \cdot (\mathrm{id}_{l_2} \otimes \langle \vec{u}_1\rangle \otimes \mathrm{id}))))) && \text{A.1} \\
= \;& \uparrow(y')((\{v'/v\}t'_1 \otimes \mathrm{id}_{k_2}) \cdot (\mathrm{p}_{l_1,r_1} \otimes \mathrm{id}) \cdot (\vec{u}_1)((\mathrm{id}_{l_1} \otimes \langle \vec{u}_1\rangle \otimes t''_2) \\
& \cdot(\mathrm{id}_{l_1} \otimes \mathrm{p}_{r_1,l_2} \otimes \mathrm{id}) \cdot (\mathrm{id}_{l_1\otimes l_2} \otimes (\vec{u}_1\vec{u}_2)\langle w\vec{u}_1\vec{u}_2\rangle) \cdot (\mathrm{p}_{l_1\otimes l_2} \otimes \mathrm{id}))) \\
= \;& \uparrow(y')((\{v'/v\}t'_1 \otimes \mathrm{id}_{k_2}) \cdot (\mathrm{id}_{l_1} \otimes \mathrm{ab}_{\vec{u}_1} t''_2) \\
& \cdot(\mathrm{id}_{l_1} \otimes \mathrm{p}_{r_1,l_2} \otimes \mathrm{id}) \cdot (\mathrm{id}_{l_1\otimes l_2} \otimes (\vec{u}_1\vec{u}_2)\langle w\vec{u}_1\vec{u}_2\rangle) \cdot (\mathrm{p}_{l_1\otimes l_2} \otimes \mathrm{id})) \\
= \;& t'_R
\end{aligned}
$$

A graphic representation of the two terms may be of assistance in following the above proof; diagrams representing t'_L and t'_R respectively are included below:





<u>**Axiom ρ_4:** $t_1 \cdot \uparrow_p t_2 = \uparrow_p((\mathbf{id}_p \otimes t_1) \cdot t_2)$</u> Let $(\vec{u})\vec{\alpha}\langle\vec{v}\rangle = (\vec{u}_1\vec{u}_2)\vec{\alpha}_1\vec{\alpha}_2\langle\vec{v}_2\rangle$.

Case For $y \notin \mathbf{fn}\,(t_2) \cup \{\vec{v}_1\}$ we have the derivation $\langle y\vec{v}_1\rangle \vdash t_2 \xrightarrow{(\vec{u}_2)\vec{\alpha}_2\langle y\vec{v}_2\rangle} t_2'$:

Left term t_L: In the following derivation, we also have side conditions

1. $\{\vec{u}_1\} \cap \mathbf{fn}\,(t_2) = \emptyset$
2. $\{y\vec{u}_2\} \cap (\mathbf{fn}\,(t_1) \cup \{\vec{z}\}) = \emptyset$
3. $|\vec{u}_1| = r_1$
4. $y \notin \mathbf{fn}\,(t_2) \cup \{\vec{v}_1\}$.

$$\dfrac{\langle\vec{z}\rangle \vdash t_1 \xrightarrow{(\vec{u}_1)\vec{\alpha}_1\langle\vec{v}_1\rangle} t_1' \qquad \dfrac{\langle y\vec{v}_1\rangle \vdash t_2 \xrightarrow{(\vec{u}_2)\vec{\alpha}_2\langle y\vec{v}_2\rangle} t_2'}{\langle\vec{v}_1\rangle \vdash \uparrow_p t_2 \xrightarrow{(y\vec{u}_2)\vec{\alpha}_2\langle\vec{v}_2\rangle} (\boldsymbol{\nu} y)(t_2' \cdot (\mathbf{id}_{l_2} \otimes \langle y\rangle \otimes \mathbf{id}))}\,\uparrow_1}{\langle\vec{z}\rangle \vdash t_1 \cdot \uparrow_p t_2 \xrightarrow{(\vec{u}_1 y\vec{u}_2)\vec{\alpha}\langle\vec{v}_2\rangle} t_L'}\,\circ$$

where $t_L' \equiv (t_1' \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}_1} t_L'') \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1,l_2} \otimes \mathbf{id})$
and $t_L'' \equiv (\boldsymbol{\nu} y)(t_2' \cdot (\mathbf{id}_{l_2} \otimes \langle y\rangle \otimes \mathbf{id}))$.

Right term t_R: In the following derivation, we also have side conditions

1. $\{y\vec{u}_1\} \cap \mathbf{fn}\,(t_2) = \emptyset$

2. $\{\vec{u}_2\} \cap (\mathbf{fn}\,(t_1) \cup \{\vec{z}\}) = \emptyset$

3. $|y\vec{u}_1| = p \otimes r_1$

4. $y \notin \mathbf{fn}\,(t_1) \cup \mathbf{fn}\,(t_2) \cup \{\vec{z}\vec{u}_1\vec{u}_2\}$.

$$\dfrac{\dfrac{\dfrac{\langle y\rangle \vdash \mathbf{id}_p \xrightarrow{\langle y\rangle} \mathbf{id}_\epsilon \quad \langle\vec{z}\rangle \vdash t_1 \xrightarrow{(\vec{u}_1)\vec{\alpha}_1\langle\vec{v}_1\rangle} t_1'}{\langle y\vec{z}\rangle \vdash \mathbf{id}_p \otimes t_1 \xrightarrow{(\vec{u}_1)\vec{\alpha}_1\langle y\vec{v}_1\rangle}{}_{=}\sigma t_1'}\otimes \qquad \langle y\vec{v}_1\rangle \vdash t_2 \xrightarrow{(\vec{u}_2)\vec{\alpha}_2\langle y\vec{v}_2\rangle} t_2'}{\langle y\vec{z}\rangle \vdash (\mathbf{id}_p \otimes t_1)\cdot t_2 \xrightarrow{(\vec{u})\vec{\alpha}\langle y\vec{v}_2\rangle}{}_{=}(t_1' \otimes \mathbf{id}_{k_2})\cdot(\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}_1}t_2')\cdot(\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1,l_2} \otimes \mathbf{id})}\circ}{\langle\vec{z}\rangle \vdash \uparrow_p((\mathbf{id}_p \otimes t_1)\cdot t_2) \xrightarrow{(y\vec{u})\vec{\alpha}\langle\vec{v}_2\rangle}{}_{=}t_R'}\uparrow_1$$

where $t_R' \equiv (\nu y)(t_R'' \cdot (\mathbf{id}_{l_1 \otimes l_2} \otimes \langle y\rangle \otimes \mathbf{id}))$ and $t_R'' \equiv (t_1' \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}_1}t_2') \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1,l_2} \otimes \mathbf{id})$.

Note that the two derivations do not derive transitions with identical labels. The labels differ by permutations of the binding vectors. From each derivation one can construct a derivation for a transition which matches the other. We will just show one of the cases.

$$\dfrac{\langle\vec{z}\rangle t_1 \cdot \uparrow_p t_2 (\vec{u}_1 y \vec{u}_2)\vec{\alpha}\langle\vec{v}_2\rangle t_L'}{\langle\vec{z}\rangle \vdash t_1 \cdot \uparrow_p t_2 \xrightarrow{(y\vec{u})\vec{\alpha}\langle\vec{v}_2\rangle} t_L' \cdot (\mathbf{id}_{l_1 \otimes l_2} \otimes \mathbf{p}_{r_1,p} \otimes \mathbf{id})}\ \text{PERM}_1$$

It is easy to see that the side conditions in each derivation are equivalent. We must now show that $t_L' \cdot (\mathbf{id}_{l_1 \otimes l_2} \otimes \mathbf{p}_{r_1,p} \otimes \mathbf{id}) = t_R'$.

$$\begin{aligned}
t'_R &= (\nu y)((t'_1 \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}_1} t'_2) \\
&\qquad \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1, l_2} \otimes \mathbf{id}) \cdot (\mathbf{id}_{l_1 \otimes l_2} \otimes \langle y \rangle \otimes \mathbf{id})) \\
&= (t'_1 \otimes \mathbf{id}_{k_2}) \cdot (\nu y)((\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}_1} t'_2) \\
&\qquad \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1, l_2} \otimes \mathbf{id}) \cdot (\mathbf{id}_{l_1 \otimes l_2} \otimes \langle y \rangle \otimes \mathbf{id}))) \\
&= (t'_1 \otimes \mathbf{id}_{k_2}) \cdot (\nu y)((\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}_1} t'_2) \\
&\qquad \cdot (\mathbf{id}_{l_1 \otimes r_1 \otimes l_2} \otimes \langle y \rangle \otimes \mathbf{id}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1, l_2 \otimes p} \otimes \mathbf{id})) \\
&= (t'_1 \otimes \mathbf{id}_{k_2}) \cdot (\nu y)((\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}_1} t'_2) \\
&\qquad \cdot (\mathbf{id}_{l_1 \otimes r_1 \otimes l_2} \otimes \langle y \rangle \otimes \mathbf{id})) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1, l_2 \otimes p} \otimes \mathbf{id}) \\
&= (t'_1 \otimes \mathbf{id}_{k_2}) \cdot (\nu y)(\mathbf{id}_{l_1} \otimes (\mathbf{ab}_{\vec{u}_1} t'_2 \cdot (\mathbf{id}_{r_1 \otimes l_2} \otimes \langle y \rangle \otimes \mathbf{id}))) \\
&\qquad \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1, l_2 \otimes p} \otimes \mathbf{id}) \\
&= (t'_1 \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes (\nu y)(\mathbf{ab}_{\vec{u}_1} t'_2 \cdot (\mathbf{id}_{r_1 \otimes l_2} \otimes \langle y \rangle \otimes \mathbf{id}))) \\
&\qquad \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1, l_2 \otimes p} \otimes \mathbf{id}) \\
&= (t'_1 \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes (\nu y)\mathbf{ab}_{\vec{u}_1} (t'_2 \cdot (\mathbf{id}_{r_1 \otimes l_2} \otimes \langle y \rangle \otimes \mathbf{id}))) \\
&\qquad \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1, l_2 \otimes p} \otimes \mathbf{id}) \\
&= (t'_1 \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes (\nu y)\mathbf{ab}_{\vec{u}_1} (t'_2 \cdot (\mathbf{id}_{r_1 \otimes l_2} \otimes \langle y \rangle \otimes \mathbf{id}))) \\
&\qquad \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1, l_2} \otimes \mathbf{id}) \cdot (\mathbf{id}_{l_1 \otimes l_2} \otimes \mathbf{p}_{r_1, p} \otimes \mathbf{id}) \\
&= t'_L \cdot (\mathbf{id}_{l_1 \otimes l_2} \otimes \mathbf{p}_{r_1, p} \otimes \mathbf{id})
\end{aligned}$$

Case For $y \notin (\mathbf{fn}(t_2) \cup \{\vec{z}\})$ we have the derivation $\langle y\vec{v}_1 \rangle \vdash t_2 \xrightarrow{(\vec{u}_2)\vec{\alpha}_2\langle w\vec{v}_2\rangle} t'_2$ with $w \neq y$:

Left term t_L: Let $\sigma' = \{{}^{w}\!/\!_{y}\}$. In the following derivation, we also have side conditions

1. $\{\vec{u}_1\} \cap \mathbf{fn}(t_2) = \emptyset$
2. $\{\vec{u}_2\} \cap (\mathbf{fn}(t_1) \cup \{\vec{z}\}) = \emptyset$
3. $|\vec{u}_1| = r_1$
4. $y \notin \mathbf{fn}(t_2) \cup \{\vec{v}_1 \vec{u}_2\}$
5. $w \neq y$.

$$\dfrac{\langle \vec{z} \rangle \vdash t_1 \xrightarrow{(\vec{u}_1)\vec{\alpha}_1\langle \vec{v}_1\rangle} t'_1 \qquad \dfrac{\langle y\vec{v}_1 \rangle \vdash t_2 \xrightarrow{(\vec{u}_2)\vec{\alpha}_2\langle w\vec{v}_2\rangle} t'_2}{\langle \vec{v}_1 \rangle \vdash \uparrow_p t_2 \xrightarrow{(\vec{u}_2)\vec{\alpha}_2\langle \vec{v}_2\rangle} \uparrow(y)(t'_2 \cdot (\mathbf{id}_{l_2} \otimes (\vec{u}_2)\langle w\vec{u}_2\rangle) \cdot (\mathbf{p}_{l_2,p} \otimes \mathbf{id}))}\uparrow_2}{\langle \vec{z} \rangle \vdash t_1 \cdot \uparrow_p t_2 \xrightarrow{(\vec{u})\vec{\alpha}_1(\sigma'\vec{\alpha}_2)\langle \sigma'\vec{v}_2\rangle} t'_L}\circ$$

where $t'_L \equiv (t'_1 \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}_1} t''_L) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1,l_2} \otimes \mathbf{id})$
and $t''_L \equiv \uparrow(y)(t'_2 \cdot (\mathbf{id}_{l_2} \otimes (\vec{u}_2)\langle w\vec{u}_2\rangle) \cdot (\mathbf{p}_{l_2,p} \otimes \mathbf{id}))$.

Right term t_R: In the following derivation, we also have side conditions

1. $\{\vec{u}_1\} \cap \mathbf{fn}\,(t_2) = \emptyset$
2. $\{\vec{u}_2\} \cap (\mathbf{fn}\,(t_1) \cup \{y'\vec{z}\}) = \emptyset$
3. $|\vec{u}_1| = r_1$
4. $y' \notin \mathbf{fn}\,(t_1) \cup \mathbf{fn}\,(t_2) \cup \{\vec{z}\vec{u}_1\vec{u}_2\}$
5. $w \notin \{\vec{u}_2\}$
6. $y' \neq w$.

Note that one of the side conditions requires that $y' \notin \mathbf{fn}\,(t_2) \cup \{\vec{u}_2\}$. Therefore, we cannot simply rely on the name y used in the derivation (involving t_L) above. Instead, we shall choose such a $y' \notin \mathbf{fn}\,(t_1) \cup \mathbf{fn}\,(t_2) \cup \{\vec{u}_1\vec{u}_2\vec{z}\}$. We shall then use the substitution lemma to establish the required correspondence between the subderivations involving y and y'. In what follows, let $\sigma = \{y'/y\}$.

$$\dfrac{\dfrac{\langle y'\rangle \vdash \mathbf{id}_p \xrightarrow{\langle y'\rangle} \mathbf{id}_\epsilon \quad \langle\vec{z}\rangle \vdash t_1 \xrightarrow{(\vec{u}_1)\vec{\alpha}_1\langle\vec{v}_1\rangle} t'_1}{\langle y'\vec{z}\rangle \vdash \mathbf{id}_p \otimes t_1 \xrightarrow{(\vec{u}_1)\vec{\alpha}_1\langle y'\vec{v}_1\rangle}_{=} t'_1} \otimes \quad \dfrac{\langle y\vec{v}_1\rangle \vdash t_2 \xrightarrow{(\vec{u}_2)\vec{\alpha}_2\langle w\vec{v}_2\rangle} t'_2}{\langle y'\sigma\vec{v}_1\rangle \vdash \sigma t_2 \xrightarrow{(\vec{u}_2)\sigma\vec{\alpha}_2\langle w\sigma\vec{v}_2\rangle}_{=} \sigma t'_2}}{\dfrac{\langle y'\vec{z}\rangle \vdash (\mathbf{id}_p \otimes t_1) \cdot t_2 \xrightarrow{(\vec{u})\vec{\alpha}_1(\sigma\vec{\alpha}_2)\langle w\sigma\vec{v}_2\rangle}_{=} (t'_1 \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}_1}\sigma t'_2) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1,l_2} \otimes \mathbf{id})}{\langle\vec{z}\rangle \vdash \uparrow_p((\mathbf{id}_p \otimes t_1) \cdot t_2) \xrightarrow{(\vec{u})\vec{\alpha}_1(\sigma'\vec{\alpha}_2)\langle\sigma'\vec{v}_2\rangle}_{=} t'_R} \uparrow_2} \circ$$

We shall now prove the equality $t'_L = t'_R$.

$$\begin{aligned}
t'_L &= (t'_1 \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}_1} \uparrow(y)(t'_2 \cdot (\mathbf{id}_{l_2} \otimes (\vec{u}_2)\langle w\vec{u}_2\rangle) \cdot (\mathbf{p}_{l_2,p} \otimes \mathbf{id}))) \\
&\qquad \cdot(\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1,l_2} \otimes \mathbf{id}) \\
&= (t'_1 \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}_1} \uparrow(y')(\{y'/y\}t'_2 \cdot (\mathbf{id}_{l_2} \otimes (\vec{u}_2)\langle w\vec{u}_2\rangle) \cdot (\mathbf{p}_{l_2,p} \otimes \mathbf{id}))) \\
&\qquad \cdot(\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1,l_2} \otimes \mathbf{id}) \\
&= (t'_1 \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes \uparrow(y')(\mathbf{ab}_{\vec{u}_1}\{y'/y\}t'_2 \cdot (\mathbf{id}_{r_1 \otimes l_2} \otimes (\vec{u}_2)\langle w\vec{u}_2\rangle) \\
&\qquad \cdot(\mathbf{id}_{r_1}\mathbf{p}_{l_2,p} \otimes \mathbf{id}) \cdot (\mathbf{p}_{r_1,p} \otimes \mathbf{id}))) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1,l_2} \otimes \mathbf{id})
\end{aligned}$$

$$\begin{aligned}
&= (t_1' \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes \uparrow(y')(\mathbf{ab}_{\vec{u}_1}\{y'/y\}t_2' \cdot (\mathbf{id}_{r_1 \otimes l_2} \otimes (\vec{u}_2)\langle w\vec{u}_2\rangle) \\
&\qquad \cdot(\mathbf{id}_{r_1}\mathbf{p}_{l_2,p} \otimes \mathbf{id}) \cdot (\mathbf{p}_{r_1,p} \otimes \mathbf{id}) \cdot (\mathbf{id}_p \otimes \mathbf{p}_{r_1,l_2} \otimes \mathbf{id}))) \\
&= (t_1' \otimes \mathbf{id}_{k_2}) \cdot \uparrow(y')((\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}_1}\{y'/y\}t_2') \\
&\qquad \cdot(\mathbf{id}_{l_1 \otimes r_1 \otimes l_2} \otimes (\vec{u}_2)\langle w\vec{u}_2\rangle) \cdot (\mathbf{id}_{l_1 \otimes r_1} \otimes \mathbf{p}_{l_2,p} \otimes \mathbf{id}) \\
&\qquad\qquad \cdot(\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1,p} \otimes \mathbf{id}) \cdot (\mathbf{id}_{l_1 \otimes p} \otimes \mathbf{p}_{r_1,l_2} \otimes \mathbf{id}) \cdot (\mathbf{p}_{l_1,p} \otimes \mathbf{id})) \\
&= \uparrow(y')((t_1' \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}_1}\{y'/y\}t_2') \\
&\qquad \cdot(\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1,l_2} \otimes (\vec{u}_2)\langle w\vec{u}_2\rangle) \cdot (\mathbf{p}_{l_1 \otimes l_2 \otimes r_1,p} \otimes \mathbf{id})) \\
&= \uparrow(y')((t_1' \otimes \mathbf{id}_{k_2}) \cdot (\mathbf{id}_{l_1} \otimes \mathbf{ab}_{\vec{u}_1}\{y'/y\}t_2') \cdot (\mathbf{id}_{l_1} \otimes \mathbf{p}_{r_1,l_2} \otimes \mathbf{id}) \\
&\qquad \cdot(\mathbf{id}_{l_1 \otimes l_2 \otimes r_1} \otimes (\vec{u}_2)\langle w\vec{u}_2\rangle) \cdot (\mathbf{p}_{l_1 \otimes l_2 \otimes r_1,p} \otimes \mathbf{id})) \\
&= t_R'
\end{aligned}$$

The above derivation shows how a matching derivation for t_R can be obtained from a derivation for t_L. We argue that obtaining a matching derivation for t_L from a derivation for t_R is simpler since the side conditions in the derivation for t_R (involving some $y \notin \mathbf{fn}\,(t_1) \cup \mathbf{fn}\,(t_2) \cup \{\vec{z}\}$) are stronger than those required for t_L.

Axiom ρ_5: $(x)\uparrow_p t = \uparrow_p((\mathbf{p}_{p,q} \otimes \mathbf{id}) \cdot (x)t) \quad (x : q)$ Straightforward.

Axiom ρ_6: $\uparrow_q\uparrow_p t = \uparrow_p\uparrow_q((\mathbf{p}_{q,p} \otimes \mathbf{id}) \cdot t \cdot (\mathbf{p}_{p,q} \otimes \mathbf{id}))$ For $y_1, y_2 \notin (\mathbf{fn}\,(t) \cup \{\vec{z}\})$ we have the derivation $\langle y_1 y_2 \vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\vec{\alpha}\langle w_1 w_2 \vec{v}\rangle} t'$. By alphaconvertibility of labels, we can assume, without loss of generality, that $\{y_1 y_2 \vec{z}\} \cap \{\vec{u}\} = \emptyset$.

Case $w_1 = y_1, w_2 = y_2$:

Left term t_L: In the following derivation, we also have side conditions

1. $y_1 \notin \mathbf{fn}\,(t) \cup \{y_2, \vec{z}\}$
2. $y_2 \notin \mathbf{fn}\,(t) \cup \{y_1, \vec{z}\}$.

$$\dfrac{\dfrac{\langle y_1 y_2 \vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\vec{\alpha}\langle y_1 y_2 \vec{v}\rangle} t'}{\langle y_2 \vec{z}\rangle \vdash \uparrow_p t \xrightarrow{(y_1\vec{u})\vec{\alpha}\langle y_2 \vec{v}\rangle} (\nu y_1)(t' \cdot (\mathbf{id}_l \otimes \langle y_1\rangle \otimes \mathbf{id}))}\uparrow_1}{\langle \vec{z}\rangle \vdash \uparrow_q\uparrow_p t \xrightarrow{(y_2 y_1 \vec{u})\vec{\alpha}\langle \vec{v}\rangle} (\nu y_2)((\nu y_1)(t' \cdot (\mathbf{id}_l \otimes \langle y_1\rangle \otimes \mathbf{id})) \cdot (\mathbf{id}_l \otimes \langle y_2\rangle \otimes \mathbf{id}))}\uparrow_1$$

Right term t_R: In the following derivation, we also have side conditions

1. $\{y_1 y_2 \vec{z}\} \cap \{\vec{u}\} = \emptyset$
2. $y_1 \notin \mathbf{fn}(t) \cup \{y_2, \vec{z}\}$
3. $y_2 \notin \mathbf{fn}(t) \cup \{y_1, \vec{z}\}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\langle y_2 y_1 \vec{z}\rangle \vdash \mathbf{p}_{q,p} \otimes \mathbf{id} \xrightarrow{\langle y_1 y_2 \vec{z}\rangle}_= \mathbf{id}_\epsilon \quad \langle y_1 y_2 \vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\vec{\alpha}\langle y_1 y_2 \vec{v}\rangle} t'}{\langle y_2 y_1 \vec{z}\rangle \vdash (\mathbf{p}_{q,p} \otimes \mathbf{id}) \cdot t \xrightarrow{(\vec{u})\vec{\alpha}\langle y_1 y_2 \vec{v}\rangle}_= t'} \circ \quad \langle y_1 y_2 \vec{v}\rangle \vdash \mathbf{p}_{p,q} \otimes \mathbf{id} \xrightarrow{\langle y_2 y_1 \vec{v}\rangle}_= \mathbf{id}_\epsilon}{\langle y_2 y_1 \vec{z}\rangle \vdash (\mathbf{p}_{q,p} \otimes \mathbf{id}) \cdot t \cdot (\mathbf{p}_{p,q} \otimes \mathbf{id}) \xrightarrow{(\vec{u})\vec{\alpha}\langle y_2 y_1 \vec{v}\rangle}_= t'} \circ}{\langle y_1 \vec{z}\rangle \vdash \uparrow_q((\mathbf{p}_{q,p} \otimes \mathbf{id}) \cdot t \cdot (\mathbf{p}_{p,q} \otimes \mathbf{id})) \xrightarrow{(y_2 \vec{u})\vec{\alpha}\langle y_1 \vec{v}\rangle}_= (\boldsymbol{\nu} y_2)(t' \cdot (\mathbf{id}_l \otimes \langle y_2\rangle \otimes \mathbf{id}))} \uparrow_1}{\langle \vec{z}\rangle \vdash \uparrow_p\uparrow_q((\mathbf{p}_{q,p} \otimes \mathbf{id}) \cdot t \cdot (\mathbf{p}_{p,q} \otimes \mathbf{id})) \xrightarrow{(y_1 y_2 \vec{u})\vec{\alpha}\langle \vec{v}\rangle}_= (\boldsymbol{\nu} y_1) t''_R \cdot (\mathbf{id}_l \otimes \langle y_1\rangle \otimes \mathbf{id}))} \uparrow_1$$

where $t''_R \equiv (\boldsymbol{\nu} y_2)(t' \cdot (\mathbf{id}_l \otimes \langle y_2\rangle \otimes \mathbf{id})$.

Note that the two derivations do not derive transitions with identical labels. The labels differ by permutations of the binding vectors. From each derivation one can construct a derivation for a transition which matches the other. We will just show one of the cases.

$$\dfrac{\langle \vec{z}\rangle \vdash \uparrow_q\uparrow_p t \xrightarrow{(y_2 y_1 \vec{u})\vec{\alpha}\langle \vec{v}\rangle} t'_L}{\langle \vec{z}\rangle \vdash \uparrow_q\uparrow_p t \xrightarrow{(y_1 y_2 \vec{u})\vec{\alpha}\langle \vec{v}\rangle} t'_L \cdot (\mathbf{id}_l \otimes \mathbf{p}_{q,p} \otimes \mathbf{id})} \text{PERM}_1$$

It is easy to see that the side conditions in each derivation are equivalent. We must now show that $t'_L \cdot (\mathbf{id}_l \otimes \mathbf{p}_{q,p} \otimes \mathbf{id}) = t'_R$.

$$\begin{aligned}
t'_L &= (\boldsymbol{\nu} y_2)((\boldsymbol{\nu} y_1)(t' \cdot (\mathbf{id}_l \otimes \langle y_1\rangle \otimes \mathbf{id})) \\
&\qquad \cdot (\mathbf{id}_l \otimes \langle y_2\rangle \otimes \mathbf{id})) \cdot (\mathbf{id}_l \otimes \mathbf{p}_{q,p} \otimes \mathbf{id}) \\
&= (\boldsymbol{\nu} y_2)((\boldsymbol{\nu} y_1)(t' \cdot (\mathbf{id}_l \otimes \langle y_1\rangle \otimes \mathbf{id})) \\
&\qquad \cdot (\mathbf{id}_l \otimes \langle y_2\rangle \otimes \mathbf{id}) \cdot (\mathbf{id}_l \otimes \mathbf{p}_{q,p} \otimes \mathbf{id})) \\
&= (\boldsymbol{\nu} y_2)(\boldsymbol{\nu} y_1)(t' \cdot (\mathbf{id}_l \otimes \langle y_1\rangle \otimes \mathbf{id}) \\
&\qquad \cdot (\mathbf{id}_l \otimes \langle y_2\rangle \otimes \mathbf{id}) \cdot (\mathbf{id}_l \otimes \mathbf{p}_{q,p} \otimes \mathbf{id})) \\
&= (\boldsymbol{\nu} y_1)(\boldsymbol{\nu} y_2)(t' \cdot (\mathbf{id}_l \otimes \langle y_2 y_1\rangle \otimes \mathbf{id}) \cdot (\mathbf{id}_l \otimes \mathbf{p}_{q,p} \otimes \mathbf{id})) \\
&= (\boldsymbol{\nu} y_1)(\boldsymbol{\nu} y_2)(t' \cdot (\mathbf{id}_l \otimes \langle y_1 y_2\rangle \otimes \mathbf{id})) \\
&= (\boldsymbol{\nu} y_1)(\boldsymbol{\nu} y_2)(t' \cdot (\mathbf{id}_l \otimes \langle y_2\rangle \otimes \mathbf{id}) \cdot (\mathbf{id}_l \otimes \langle y_1\rangle \otimes \mathbf{id})) \\
&= (\boldsymbol{\nu} y_1)((\boldsymbol{\nu} y_2)(t' \cdot (\mathbf{id}_l \otimes \langle y_2\rangle \otimes \mathbf{id}))) \cdot (\mathbf{id}_l \otimes \langle y_1\rangle \otimes \mathbf{id}) \\
&= t'_R
\end{aligned}$$

Case $w_1 = y_1, w_2 \neq y_2$**:** Let $\sigma_2 = \{w_2/y_2\}$.

Left term t_L**:** In the following derivation, we also have side conditions

1. $w_2 \neq y_2$
2. $y_1 \notin \mathbf{fn}(t) \cup \{y_2, \vec{z}\}$
3. $y_2 \notin \mathbf{fn}(t) \cup \{y_1, \vec{z}\}$.

$$\dfrac{\dfrac{\langle y_1 y_2 \vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\vec{\alpha}\langle y_1 w_2 \vec{v}\rangle} t'}{\langle y_2 \vec{z}\rangle \vdash \Uparrow_p t \xrightarrow{(y_1\vec{u})\vec{\alpha}\langle w_2 \vec{v}\rangle} (\nu y_1)(t' \cdot (\mathbf{id}_l \otimes \langle y_1\rangle \otimes \mathbf{id}))}\ \Uparrow_1}{\langle \vec{z}\rangle \vdash \Uparrow_q \Uparrow_p t \xrightarrow{(y_1\vec{u})\sigma_2\vec{\alpha}\langle \sigma_2 \vec{v}\rangle} \Uparrow_q (y_2)(t''_L \cdot (\mathbf{id}_l \otimes (y_1\vec{u})\langle w_2 y_1 \vec{u}\rangle \otimes \mathbf{id}) \cdot (\mathbf{p}_{l,q} \otimes \mathbf{id}))}\ \Uparrow_2$$

where $t''_L \equiv (\nu y_1)(t' \cdot (\mathbf{id}_l \otimes \langle y_1\rangle \otimes \mathbf{id})$.

Right term t_R**:** In the following derivation, we also have side conditions

1. $w_2 \neq y_2$
2. $\{y_1 y_2 \vec{z}\} \cap \{\vec{u}\} = \emptyset$
3. $y_1 \notin \mathbf{fn}(t) \cup \{y_2, \vec{z}\}$
4. $y_2 \notin \mathbf{fn}(t) \cup \{y_1, \vec{z}\}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\langle y_2 y_1 \vec{z}\rangle \vdash \mathbf{p}_{q,p} \otimes \mathbf{id} \xrightarrow{\langle y_1 y_2 \vec{z}\rangle}_{=} \mathbf{id}_\epsilon \quad \langle y_1 y_2 \vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\vec{\alpha}\langle y_1 w_2 \vec{v}\rangle} t'}{\langle y_2 y_1 \vec{z}\rangle \vdash (\mathbf{p}_{q,p} \otimes \mathbf{id}) \cdot t \xrightarrow{(\vec{u})\vec{\alpha}\langle y_1 w_2 \vec{v}\rangle}_{=} t'}\ \circ \qquad \langle y_1 w_2 \vec{v}\rangle \vdash \mathbf{p}_{p,q} \otimes \mathbf{id} \xrightarrow{\langle w_2 y_1 \vec{v}\rangle}_{=} \mathbf{id}_\epsilon}{\langle y_2 y_1 \vec{z}\rangle \vdash (\mathbf{p}_{q,p} \otimes \mathbf{id}) \cdot t \cdot (\mathbf{p}_{p,q} \otimes \mathbf{id}) \xrightarrow{(\vec{u})\vec{\alpha}\langle w_2 y_1 \vec{v}\rangle}_{=} t'}\ \circ}{\langle y_1 \vec{z}\rangle \vdash \Uparrow_q((\mathbf{p}_{q,p} \otimes \mathbf{id}) \cdot t \cdot (\mathbf{p}_{p,q} \otimes \mathbf{id})) \xrightarrow{(\vec{u})\sigma_2\vec{\alpha}\langle y_1 \sigma_2 \vec{v}\rangle}_{=} \Uparrow_q (y_2)(t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle w_2 \vec{u}\rangle) \cdot (\mathbf{p}_{l,q} \otimes \mathbf{id}))}\ \Uparrow_2}{\langle \vec{z}\rangle \vdash \Uparrow_p \Uparrow_q((\mathbf{p}_{q,p} \otimes \mathbf{id}) \cdot t \cdot (\mathbf{p}_{p,q} \otimes \mathbf{id})) \xrightarrow{(y_1\vec{u})\sigma_2\vec{\alpha}\langle \sigma_2 \vec{v}\rangle}_{=} (\nu y_1)(t''_R \cdot (\mathbf{id}_l \otimes \langle y_1\rangle \otimes \mathbf{id}))}\ \Uparrow_1$$

where $t''_R \equiv \Uparrow_q (y_2)(t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle w_2 \vec{u}\rangle) \cdot (\mathbf{p}_{l,q} \otimes \mathbf{id}))$.

We shall now prove the equality $t'_L = t'_R$.

$$
\begin{aligned}
t'_R &= (\nu y_1)(\uparrow_q(y_2)(t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle w_2\vec{u}\rangle) \cdot (\mathbf{p}_{l,q} \otimes \mathbf{id})) \\
&\qquad \cdot(\mathbf{id}_l \otimes \langle y_1\rangle \otimes \mathbf{id})) \\
&= (\nu y_1)\uparrow_q(y_2)(t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle w_2\vec{u}\rangle) \cdot (\mathbf{p}_{l,q} \otimes \mathbf{id}) \\
&\qquad \cdot(\mathbf{id}_{q\otimes l} \otimes \langle y_1\rangle \otimes \mathbf{id})) \\
&= \uparrow_q(y_2)(\nu y_1)(t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle w_2\vec{u}\rangle) \cdot (\mathbf{p}_{l,q} \otimes \mathbf{id}) \\
&\qquad (\mathbf{id}_{q\otimes l} \otimes \langle y_1\rangle \otimes \mathbf{id})) \\
&= \uparrow_q(y_2)(\nu y_1)(t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle w_2 y_1\vec{u}\rangle) \cdot (\mathbf{p}_{l,q} \otimes \mathbf{id})) \\
&= \uparrow_q(y_2)(\nu y_1)(t' \cdot (\mathbf{id}_l \otimes \langle y_1\rangle \otimes \mathbf{id}) \cdot (\mathbf{id}_l \otimes (y_1\vec{u})\langle w_2 y_1\vec{u}\rangle) \\
&\qquad \cdot(\mathbf{p}_{l,q} \otimes \mathbf{id})) \\
&= \uparrow_q(y_2)((\nu y_1)(t' \cdot (\mathbf{id}_l \otimes \langle y_1\rangle \otimes \mathbf{id}) \cdot (\mathbf{id}_l \otimes (y_1\vec{u})\langle w_2 y_1\vec{u}\rangle) \\
&\qquad \cdot(\mathbf{p}_{l,q} \otimes \mathbf{id})) \\
&= t'_L
\end{aligned}
$$

Case $w_1 \neq y_1, w_2 = y_2$: Let $\sigma_1 = \{{}^{w_1}\!/\!{}_{y_1}\}$.

Left term t_L: In the following derivation, we also have side conditions

1. $w_1 \neq y_1$
2. $y_1 \notin \mathbf{fn}(t) \cup \{y_2, \vec{z}\}$
3. $y_2 \notin \mathbf{fn}(t) \cup \{y_1, \vec{z}\}$.

$$
\dfrac{\dfrac{\langle y_1 y_2 \vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\vec{\alpha}\langle w_1 y_2 \vec{v}\rangle} t'}{\langle y_2\vec{z}\rangle \vdash \uparrow_p t \xrightarrow{(\vec{u})\sigma_1\vec{\alpha}\langle y_2\sigma_1\vec{v}\rangle} \uparrow_p(y_1)(t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle w_1\vec{u}\rangle) \cdot (\mathbf{p}_{l,p} \otimes \mathbf{id}))}\uparrow_2}{\langle\vec{z}\rangle \vdash \uparrow_q\uparrow_p t \xrightarrow{(y_2\vec{u})\sigma_1\vec{\alpha}\langle\sigma_1\vec{v}\rangle} (\nu y_2)(t''_L \cdot (\mathbf{id}_l \otimes \langle y_2\rangle \otimes \mathbf{id}))}\uparrow_1
$$

where $t''_L \equiv \uparrow_p(y_1)(t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle w_1\vec{u}\rangle) \cdot (\mathbf{p}_{l,p} \otimes \mathbf{id}))$.

Right term t_R: In the following derivation, we also have side conditions

1. $w_1 \neq y_1$
2. $\{y_1 y_2 \vec{z}\} \cap \{\vec{u}\} = \emptyset$
3. $y_1 \notin \mathbf{fn}(t) \cup \{y_2, \vec{z}\}$
4. $y_2 \notin \mathbf{fn}(t) \cup \{y_1, \vec{z}\}$.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\langle y_2 y_1 \vec{z}\rangle \vdash \mathbf{p}_{q,p} \otimes \mathbf{id} \xrightarrow{\langle y_1 y_2 \vec{z}\rangle}_{=} \mathbf{id}_\epsilon \quad \langle y_1 y_2 \vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\vec{\alpha}\langle w_1 y_2 \vec{v}\rangle} t'}{\langle y_2 y_1 \vec{z}\rangle \vdash (\mathbf{p}_{q,p} \otimes \mathbf{id}) \cdot t \xrightarrow{(\vec{u})\vec{\alpha}\langle w_1 y_2 \vec{v}\rangle}_{=} t'} \circ \qquad \langle w_1 y_2 \vec{v}\rangle \vdash \mathbf{p}_{p,q} \otimes \mathbf{id} \xrightarrow{\langle y_2 w_1 \vec{v}\rangle}_{=} \mathbf{id}_\epsilon}{\langle y_2 y_1 \vec{z}\rangle \vdash (\mathbf{p}_{q,p} \otimes \mathbf{id}) \cdot t \cdot (\mathbf{p}_{p,q} \otimes \mathbf{id}) \xrightarrow{(\vec{u})\vec{\alpha}\langle w_1 y_2 \vec{v}\rangle}_{=} t'} \circ}{\langle y_1 \vec{z}\rangle \vdash \uparrow_q((\mathbf{p}_{q,p} \otimes \mathbf{id}) \cdot t \cdot (\mathbf{p}_{p,q} \otimes \mathbf{id})) \xrightarrow{(y_2\vec{u})\vec{\alpha}\langle w_1 \vec{v}\rangle}_{=} (\boldsymbol{\nu} y_2)(t' \cdot (\mathbf{id}_l \otimes \langle y_2\rangle \otimes \mathbf{id}))} \uparrow_1}{\langle \vec{z}\rangle \vdash \uparrow_p\uparrow_q((\mathbf{p}_{q,p} \otimes \mathbf{id}) \cdot t \cdot (\mathbf{p}_{p,q} \otimes \mathbf{id})) \xrightarrow{(y_2\vec{u})\sigma_1\vec{\alpha}\langle \sigma_1 \vec{v}\rangle}_{=} \uparrow_p(y_1)(t''_R \cdot (\mathbf{id}_l \otimes (y_2\vec{u})\langle w_1 y_2 \vec{u}\rangle) \cdot (\mathbf{p}_{l,p} \otimes \mathbf{id}))} \uparrow_2
$$

where $t''_R \equiv (\boldsymbol{\nu} y_2)(t' \cdot (\mathbf{id}_l \otimes \langle y_2\rangle \otimes \mathbf{id}))$.

We shall now prove the equality $t'_L = t'_R$.

$$
\begin{aligned}
t'_L &= (\boldsymbol{\nu} y_2)(\uparrow_p(y_1)(t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle w_1 \vec{u}\rangle) \cdot (\mathbf{p}_{l,p} \otimes \mathbf{id})) \\
&\qquad \cdot(\mathbf{id}_l \otimes \langle y_2\rangle \otimes \mathbf{id})) \\
&= (\boldsymbol{\nu} y_2)\uparrow_p(y_1)(t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle w_1 \vec{u}\rangle) \cdot (\mathbf{p}_{l,p} \otimes \mathbf{id}) \\
&\qquad \cdot(\mathbf{id}_{p\otimes l} \otimes \langle y_2\rangle \otimes \mathbf{id})) \\
&= \uparrow_p(y_1)(\boldsymbol{\nu} y_2)(t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle w_1 \vec{u}\rangle) \cdot (\mathbf{p}_{l,p} \otimes \mathbf{id}) \\
&\qquad \cdot(\mathbf{id}_{p\otimes l} \otimes \langle y_2\rangle \otimes \mathbf{id})) \\
&= \uparrow_p(y_1)(\boldsymbol{\nu} y_2)(t' \cdot (\mathbf{id}_l \otimes \langle y_2\rangle \otimes \mathbf{id}) \cdot (\mathbf{id}_l \otimes (y_2\vec{u})\langle w_1 y_2 \vec{u}\rangle) \\
&\qquad \cdot(\mathbf{p}_{l,p} \otimes \mathbf{id})) \\
&= \uparrow_p(y_1)((\boldsymbol{\nu} y_2)(t' \cdot (\mathbf{id}_l \otimes \langle y_2\rangle \otimes \mathbf{id})) \cdot (\mathbf{id}_l \otimes (y_2\vec{u})\langle w_1 y_2 \vec{u}\rangle) \\
&\qquad \cdot(\mathbf{p}_{l,p} \otimes \mathbf{id}) \\
&= t'_R
\end{aligned}
$$

Case $w_1 \neq y_1, w_2 \neq y_2$:

Subcase $w_1 = y_2, w_2 = y_1$: Let $\sigma_1 = \{y_2/y_1\}$ and $\sigma_2 = \{y_1/y_2\}$.

Left term t_L: In the following derivation, we also have side conditions

1. $y_1 \neq y_2$
2. $y_1 \notin \mathbf{fn}(t) \cup \{y_2, \vec{u}, \vec{z}\}$
3. $y_2 \notin \mathbf{fn}(t) \cup \{\vec{z}\}$.

$$
\dfrac{\dfrac{\langle y_1 y_2 \vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\vec{\alpha}\langle y_2 y_1 \vec{v}\rangle} t'}{\langle y_2 \vec{z}\rangle \vdash \uparrow_p t \xrightarrow{(\vec{u})\sigma_1\vec{\alpha}\langle y_2 \sigma_1 \vec{v}\rangle} \uparrow_p(y_1)(t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle y_2 \vec{u}\rangle) \cdot (\mathbf{p}_{l,p} \otimes \mathbf{id}))} \uparrow_2}{\langle \vec{z}\rangle \vdash \uparrow_p\uparrow_p t \xrightarrow{(y_2\vec{u})\sigma_1\vec{\alpha}\langle \sigma_1 \vec{v}\rangle} (\boldsymbol{\nu} y_2)(t''_L \cdot (\mathbf{id}_l \otimes \langle y_2\rangle \otimes \mathbf{id}))} \uparrow_1
$$

where $t''_L \equiv \uparrow_p(y_1)(t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle y_2\vec{u}\rangle) \cdot (\mathbf{p}_{l,p} \otimes \mathbf{id}))$.

Right term t_R: In the following derivation, we also have side conditions

1. $\{y_1y_2\vec{z}\} \cap \{\vec{u}\} = \emptyset$
2. $y_1 \neq y_2$
3. $y_1 \notin \mathbf{fn}\,(t) \cup \{\vec{z}\}$
4. $y_2 \notin \mathbf{fn}\,(t) \cup \{y_1, \vec{u}, \vec{z}\}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\langle y_2y_1\vec{z}\rangle \vdash \mathbf{p}_{p,p} \otimes \mathbf{id} \xrightarrow{\langle y_1y_2\vec{z}\rangle}_= \mathbf{id}_\epsilon \quad \langle y_1y_2\vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\vec{\alpha}\langle y_2y_1\vec{v}\rangle} t'}{\langle y_2y_1\vec{z}\rangle \vdash (\mathbf{p}_{p,p} \otimes \mathbf{id}) \cdot t \xrightarrow{(\vec{u})\vec{\alpha}\langle y_2y_1\vec{v}\rangle}_= t'}\circ \qquad \langle y_2y_1\vec{v}\rangle \vdash \mathbf{p}_{p,p} \otimes \mathbf{id} \xrightarrow{\langle y_1y_2\vec{v}\rangle}_= \mathbf{id}_\epsilon}{\langle y_2y_1\vec{z}\rangle \vdash (\mathbf{p}_{p,p} \otimes \mathbf{id}) \cdot t \cdot (\mathbf{p}_{p,p} \otimes \mathbf{id}) \xrightarrow{(\vec{u})\vec{\alpha}\langle y_1y_2\vec{v}\rangle}_= t'}\circ}{\langle y_1\vec{z}\rangle \vdash \uparrow_p((\mathbf{p}_{p,p} \otimes \mathbf{id}) \cdot t \cdot (\mathbf{p}_{p,p} \otimes \mathbf{id})) \xrightarrow{(y_1\vec{u})\vec{\alpha}\langle w_1\vec{v}\rangle}_= \uparrow_p(y_2)(t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle y_2\vec{u}\rangle) \cdot (\mathbf{p}_{l,p} \otimes \mathbf{id}))}\uparrow_2}{\langle\vec{z}\rangle \vdash \uparrow_p\uparrow_p((\mathbf{p}_{p,p} \otimes \mathbf{id}) \cdot t \cdot (\mathbf{p}_{p,p} \otimes \mathbf{id})) \xrightarrow{(y_1\vec{u})\sigma_2\vec{\alpha}\langle\sigma_2\vec{v}\rangle}_= (\nu y_1)(t''_R \cdot (\mathbf{id}_l \otimes \langle y_1\rangle \otimes \mathbf{id}))}\uparrow_1$$

where $t''_R \equiv \uparrow_p(y_2)(t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle y_1\vec{u}\rangle) \cdot (\mathbf{p}_{l,p} \otimes \mathbf{id}))$.

Equality of labels and residuals follows by alphaconversion.

Subcase $\neg(w_1 = y_2 \wedge w_2 = y_1)$: Let $\sigma_1 = \{w_1/y_1\}$, $\sigma_2 = \{w_2/y_2\}$, $\sigma_1' = \{(\sigma_1w_2)/y_2\}\sigma_1$ and $\sigma_2' = \{(\sigma_2w_1)/y_1\}\sigma_2$.

Left term t_L: In the following derivation, we also have side conditions

1. $w_1 \neq y_1$, $\sigma_1w_2 \neq y_2$
2. $y_1 \notin \mathbf{fn}\,(t) \cup \{y_2, \vec{u}, \vec{z}\}$
3. $y_2 \notin \mathbf{fn}\,(t) \cup \{\vec{u}, \vec{z}\}$.

$$\dfrac{\dfrac{\langle y_1y_2\vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\vec{\alpha}\langle w_1w_2\vec{v}\rangle} t'}{\langle y_2\vec{z}\rangle \vdash \uparrow_p t \xrightarrow{(\vec{u})\sigma_1\vec{\alpha}\langle\sigma_1w_2\vec{v}\rangle} \uparrow_p(y_1)(t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle w_1\vec{u}\rangle) \cdot (\mathbf{p}_{l,p} \otimes \mathbf{id}))}\uparrow_2}{\langle\vec{z}\rangle \vdash \uparrow_q\uparrow_p t \xrightarrow{(\vec{u})\sigma_1'\vec{\alpha}\langle\sigma_1'\vec{v}\rangle} \uparrow_q(y_2)(t''_L \cdot (\mathbf{id}_l \otimes (\vec{u})\langle(\sigma_1w_2)\vec{u}\rangle) \cdot (\mathbf{p}_{l,q} \otimes \mathbf{id}))}\uparrow_2$$

where $t''_L \equiv \uparrow_p(y_1)(t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle w_1\vec{u}\rangle) \cdot (\mathbf{p}_{l,p} \otimes \mathbf{id}))$.

Right term t_R: In the following derivation, we also have side conditions

1. $\{y_1 y_2 \vec{z}\} \cap \{\vec{u}\} = \emptyset$
2. $\sigma_2 w_1 \neq y_1,\ w_2 \neq y_2$
3. $y_1 \notin \mathbf{fn}\,(t) \cup \{\vec{u}, \vec{z}\}$
4. $y_2 \notin \mathbf{fn}\,(t) \cup \{y_1, \vec{u}, \vec{z}\}$.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\langle y_2 y_1 \vec{z}\rangle \vdash \mathbf{p}_{q,p} \otimes \mathbf{id} \xrightarrow{\langle y_1 y_2 \vec{z}\rangle}_{=} \mathbf{id}_\epsilon \quad \langle y_1 y_2 \vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\vec{\alpha}\langle w_1 w_2 \vec{v}\rangle} t'}{\langle y_2 y_1 \vec{z}\rangle \vdash (\mathbf{p}_{q,p} \otimes \mathbf{id}) \cdot t \xrightarrow{(\vec{u})\vec{\alpha}\langle w_1 w_2 \vec{v}\rangle}_{=} t'}\ \circ \qquad \langle w_1 w_2 \vec{v}\rangle \vdash \mathbf{p}_{p,q} \otimes \mathbf{id} \xrightarrow{\langle w_2 w_1 \vec{v}\rangle}_{=} \mathbf{id}_\epsilon}{\langle y_2 y_1 \vec{z}\rangle \vdash (\mathbf{p}_{q,p} \otimes \mathbf{id}) \cdot t \cdot (\mathbf{p}_{p,q} \otimes \mathbf{id}) \xrightarrow{(\vec{u})\vec{\alpha}\langle w_2 w_1 \vec{v}\rangle}_{=} t'}\ \circ}{\langle y_1 \vec{z}\rangle \vdash \uparrow_q((\mathbf{p}_{q,p} \otimes \mathbf{id}) \cdot t \cdot (\mathbf{p}_{p,q} \otimes \mathbf{id})) \xrightarrow{(\vec{u})\sigma_2\vec{\alpha}\langle \sigma_2 w_1 \vec{v}\rangle}_{=} \uparrow_q(y_2)(t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle w_2 \vec{u}\rangle) \cdot (\mathbf{p}_{l,q} \otimes \mathbf{id}))}\ \uparrow_2}{\langle \vec{z}\rangle \vdash \uparrow_p\uparrow_q((\mathbf{p}_{q,p} \otimes \mathbf{id}) \cdot t \cdot (\mathbf{p}_{p,q} \otimes \mathbf{id})) \xrightarrow{(\vec{u})\sigma_2'\vec{\alpha}\langle \sigma_2' \vec{v}\rangle}_{=} \uparrow_p(y_1)(t_R'' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle (\sigma_2 w_1)\vec{u}\rangle) \cdot (\mathbf{p}_{l,p} \otimes \mathbf{id}))}\ \uparrow_2
$$

where $t_R'' \equiv \uparrow_q(y_2)(t' \cdot (\mathbf{id}_l \otimes (\vec{u})\langle w_2 \vec{u}\rangle) \cdot (\mathbf{p}_{l,q} \otimes \mathbf{id}))$.

To show that the labels for left and right terms are equal, it suffices to show that $\sigma_1' = \sigma_2'$. We are working under the following assumptions:

1. $w_1 \neq y_2 \vee w_2 \neq y_1$
2. $y_1 \neq y_2$
3. $w_1 \neq y_1$
4. $w_1 \neq y_2$.

Case $w_1 \neq y_2, w_2 \neq y_1$ Then

$$
\begin{aligned}
\{{}^{\{w_2/y_2\}}w_1/y_1\}\{w_2/y_2\} &= \{w_1/y_1\}\{w_2/y_2\} \\
&= \{w_2/y_2\}\{w_1/y_1\} = \{{}^{\{w_1/y_1\}}w_2/y_2\}\{w_1/y_1\}
\end{aligned}
$$

Case $w_1 \neq y_2, w_2 = y_1$ Then

$$
\begin{aligned}
\{{}^{\{w_2/y_2\}}w_1/y_1\}\{w_2/y_2\} &= \{w_1/y_1\}\{y_1/y_2\} = \{w_1/y_1\}\{w_1/y_2\} \\
&= \{w_1/y_2\}\{w_1/y_1\} = \{{}^{\{w_1/y_1\}}w_2/y_2\}\{w_1/y_1\}
\end{aligned}
$$

Case $w_1 = y_2, w_2 \neq y_1$ Then

$$
\begin{aligned}
\{{}^{\{w_2/y_2\}}w_1/y_1\}\{w_2/y_2\} &= \{w_1/y_1\}\{y_1/y_2\} = \{w_1/y_1\}\{w_1/y_2\} \\
&= \{w_1/y_2\}\{w_1/y_1\} = \{{}^{\{w_1/y_1\}}w_2/y_2\}\{w_1/y_1\}
\end{aligned}
$$

We shall now prove the equality $t'_L = t'_R$.

$$\begin{aligned}
t'_R &= \uparrow_p(y_1)(\uparrow_q(y_2)(t' \cdot (\mathrm{id}_l \otimes (\vec{u})\langle w_2\vec{u}\rangle) \cdot (\mathrm{p}_{l,q} \otimes \mathrm{id})) \\
&\qquad \cdot(\mathrm{id}_l \otimes (\vec{u})\langle(\{^{w_2}/_{y_2}\}w_1)\vec{u}\rangle) \cdot (\mathrm{p}_{l,p} \otimes \mathrm{id})) \\
&= \uparrow_p(y_1)\uparrow_q(y_2)(t' \cdot (\mathrm{id}_l \otimes (\vec{u})\langle w_2\vec{u}\rangle) \cdot (\mathrm{p}_{l,q} \otimes \mathrm{id}) \\
&\qquad \cdot(\mathrm{id}_{q\otimes l} \otimes (\vec{u})\langle(\{^{w_2}/_{y_2}\}w_1)\vec{u}\rangle) \cdot (\mathrm{id}_q \otimes \mathrm{p}_{l,p} \otimes \mathrm{id})) \\
&= \uparrow_p(y_1)\uparrow_q(y_2)(t' \cdot (\mathrm{p}_{l,r} \otimes \mathrm{id}) \cdot (\vec{u})((id_l \otimes \langle w_2\vec{u}\rangle) \cdot (\mathrm{p}_{l,q} \otimes \mathrm{id}) \\
&\qquad \cdot(\mathrm{id}_{q\otimes l} \otimes (\vec{u})\langle(\{^{w_2}/_{y_2}\}w_1)\vec{u}\rangle) \cdot (\mathrm{id}_q \otimes \mathrm{p}_{l,p} \otimes \mathrm{id}))) \\
&= \uparrow_p(y_1)\uparrow_q(y_2)(t' \cdot (\mathrm{p}_{l,r} \otimes \mathrm{id}) \cdot (\vec{u})(\langle w_2\rangle \otimes \langle\{^{w_2}/_{y_2}\}w_1\rangle \otimes \mathrm{id}_l \otimes \langle\vec{u}\rangle)) \\
&= \uparrow_p(y_1)\uparrow_q(y_2)(t' \cdot (\mathrm{p}_{l,r} \otimes \mathrm{id}) \cdot (\vec{u})(\langle w_2\rangle \otimes \langle w_1\rangle \otimes \mathrm{id}_l \otimes \langle\vec{u}\rangle)) && \text{A.1} \\
&= \uparrow_q(y_2)\uparrow_p(y_1)(t' \cdot (\mathrm{p}_{l,r} \otimes \mathrm{id}) \cdot (\vec{u})(\langle w_1\rangle \otimes \langle w_2\rangle \otimes \mathrm{id}_l \otimes \langle\vec{u}\rangle)) \\
&= \uparrow_q(y_2)\uparrow_p(y_1)(t' \cdot (\mathrm{p}_{l,r} \otimes \mathrm{id}) \cdot (\vec{u})(\langle w_1\rangle \otimes \langle\{^{w_1}/_{y_1}\}w_2\rangle \otimes \mathrm{id}_l \otimes \langle\vec{u}\rangle)) && \text{A.1} \\
&= \uparrow_q(y_2)\uparrow_p(y_1)(t' \cdot (\mathrm{p}_{l,r} \otimes \mathrm{id}) \cdot (\vec{u})((id_l \otimes \langle w_1\vec{u}\rangle) \cdot (\mathrm{p}_{l,q} \otimes \mathrm{id}) \\
&\qquad \cdot(\mathrm{id}_{p\otimes l} \otimes (\vec{u})\langle(\{^{w_1}/_{y_1}\}w_2)\vec{u}\rangle) \cdot (\mathrm{id}_p \otimes \mathrm{p}_{l,p} \otimes \mathrm{id}))) \\
&= \uparrow_q(y_2)\uparrow_p(y_1)(t' \cdot (\mathrm{id}_l \otimes (\vec{u})\langle w_1\vec{u}\rangle) \cdot (\mathrm{p}_{l,p} \otimes \mathrm{id}) \\
&\qquad \cdot(\mathrm{id}_{p\otimes l} \otimes (\vec{u})\langle(\{^{w_1}/_{y_1}\}w_2)\vec{u}\rangle) \cdot (\mathrm{id}_p \otimes \mathrm{p}_{l,q} \otimes \mathrm{id})) \\
&= \uparrow_q(y_2)(\uparrow_p(y_1)(t' \cdot (\mathrm{id}_l \otimes (\vec{u})\langle w_1\vec{u}\rangle) \cdot (\mathrm{p}_{l,p} \otimes \mathrm{id})) \\
&\qquad \cdot(\mathrm{id}_l \otimes (\vec{u})\langle(\{^{w_1}/_{y_1}\}w_2)\vec{u}\rangle) \cdot (\mathrm{p}_{l,q} \otimes \mathrm{id})) \\
&= t'_L
\end{aligned}$$

By the standard derivation lemma, for any derivable $\langle\vec{z}\rangle \vdash t_1 \xrightarrow{\ell} t'_1$, there is a subderivation, for some δ and $t''_1 = t'_1$, of $\langle\vec{z}\rangle \vdash t_1 \xrightarrow{\delta} t''_1$ following which only permutation and SYNC rules are applied. The application of these rules does not depend on the structure of t_1 but only on the labels of the transitions. Moreover, the residual of these rules is obtained by introducing contructions around the residual of the premise which depend only on the labels. By the above, for some t''_2, of $\langle\vec{z}\rangle \vdash t_2 \xrightarrow{\delta} t''_2$ with $t''_1 = t''_2$. Applying the same sequence of permutation and SYNC rules to this derivation clearly gives a derivation of $\langle\vec{z}\rangle \vdash t_2 \xrightarrow{\ell} t'_2$ for some t'_2 which is equal to t'_1. ∎

A.3 Labelled Transitions

Proposition 5.13

1. $\langle\vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\langle\vec{v}\rangle}_= t' \iff [\![\langle\vec{z}\rangle \cdot t]\!] = [\vec{\mu}(\vec{u}')]\langle\vec{v}\rangle$ and $[\![t']\!] = [\vec{\mu}(\vec{u}')]\langle\vec{u}\rangle$;

2. $\langle\vec{z}\rangle \vdash t \xrightarrow{(\vec{u})\overline{x}\langle\vec{v}\rangle}_= t' \iff [\![\langle\vec{z}\rangle \cdot t]\!] = [\langle x\vec{w}\rangle\mathbf{out}, \vec{\mu}(\vec{u}')]\langle\vec{v}\rangle$ and $[\![t']\!] = [\vec{\mu}(\vec{u}')]\langle\vec{w}\vec{u}\rangle$;

3. $\langle\vec{z}\rangle \vdash t \xrightarrow{(\vec{u})x\langle\vec{v}\rangle}_= t' \iff [\![\langle\vec{z}\rangle \cdot t]\!] = [\langle x\rangle\mathbf{box}a(\vec{u}_1), \vec{\mu}(\vec{u}_2)]\langle\vec{v}\rangle$
and $[\![t']\!] = {}^{\bullet}\{\vec{z}_a/\vec{u}_1\}(\vec{x}_a)[\vec{\lambda}(\vec{y}_a), \vec{\mu}(\vec{u}_2)]\langle\vec{u}\rangle$.

with $a = (\vec{x}_a)\vec{\lambda}(\vec{y}_a)\langle\vec{z}_a\rangle$ and $\{\vec{u}\} = \{\vec{u}_1\vec{u}_2\} = \{\vec{u}'\}$.

Proof ($\Longleftarrow$) Let the (unique) inverse map of $[\![-]\!]$ be $\widehat{-}$. Then, by structural lemma and lemma 5.12, it suffices to give a derivation of $\vdash \widehat{[\![\langle\vec{z}\rangle \cdot t]\!]} \xrightarrow{\ell}_= \widehat{[\![t']\!]}$.

(1) Consider the inverse translations of the molecular forms of $\langle\vec{z}\rangle \cdot t$ and t', assuming $\vec{u}' : m$.

$$\begin{aligned} \widehat{[\![\langle\vec{z}\rangle \cdot t]\!]} &= \uparrow_m(\vec{u}')(\widehat{\vec{\mu}} \otimes \langle\vec{v}\rangle) \\ \widehat{[\![t']\!]} &= \uparrow_m(\vec{u}')(\widehat{\vec{\mu}} \otimes \langle\vec{u}\rangle) \end{aligned}$$

By lemma A.5, for any $\vec{y}, \vec{y}'$ such that $\{\vec{y}\} \cap \{\vec{v}\} = \emptyset$ and $\{\vec{y}\} = \{\vec{y}'\}$, we have:

$$\vdash \widehat{\vec{\mu}} \otimes \langle\vec{v}\rangle \xrightarrow{(\vec{y})\langle\vec{y}'\vec{v}\rangle} \widehat{\vec{\mu}} \cdot (\vec{y}')\langle\vec{y}\rangle$$

Choosing $\vec{y}, \vec{y}'$ such that $\{\vec{y}\} \cap \{\vec{u}\} = \emptyset$ and $(\vec{y}')\langle\vec{y}\rangle = (\vec{u}')\langle\vec{u}\rangle$, we can derive by lemma A.3, and the above transition:

$$\vdash \uparrow_m(\vec{u}')(\widehat{mu} \otimes \langle\vec{v}\rangle) \xrightarrow{(\vec{y})\langle\{\vec{y}'/\vec{u}'\}\vec{v}\rangle}_= \uparrow_m(\vec{u}')(\widehat{\vec{\mu}} \cdot (\vec{y}')\langle\vec{y}\rangle \cdot (\vec{y})\langle\vec{y}'\vec{y}\rangle)$$

But since $\{\vec{y}'/\vec{u}'\} = \{\vec{y}/\vec{u}\}$, we have $(\vec{y})\langle\{\vec{y}'/\vec{u}'\}\vec{v}\rangle = (\vec{y})\langle\{\vec{y}/\vec{u}\}\vec{v}\rangle = (\vec{u})\langle\vec{v}\rangle$. We shall now show that the residual is as expected:

$$
\begin{aligned}
&\Uparrow_m(\vec{u}')(\widehat{\mu} \otimes \langle\vec{u}\rangle) \\
&\quad = \Uparrow_m(\vec{u}')((\widehat{\mu} \cdot (\vec{y}')\langle\vec{y}'\rangle) \otimes \langle\vec{u}\rangle) && \text{2.16(5)} \\
&\quad = \Uparrow_m(\vec{u}')(\widehat{\mu} \cdot ((\vec{y}')\langle\vec{y}'\rangle \otimes \langle\vec{u}\rangle)) \\
&\quad = \Uparrow_m(\vec{u}')(\widehat{\mu} \cdot (\vec{y}')(\langle\vec{y}'\rangle \otimes \langle\vec{u}\rangle)) && \text{2.16(2)} \\
&\quad = \Uparrow_m(\vec{u}')(\widehat{\mu} \cdot (\vec{y}')(\langle\vec{y}'\rangle \otimes \{\vec{y}'/\vec{u}'\}\langle\vec{u}\rangle)) && \text{A.1} \\
&\quad = \Uparrow_m(\vec{u}')(\widehat{\mu} \cdot (\vec{y}')(\langle\vec{y}'\rangle \otimes \{\vec{y}/\vec{u}\}\langle\vec{u}\rangle)) \\
&\quad = \Uparrow_m(\vec{u}')(\widehat{\mu} \cdot (\vec{y}')\langle\vec{y}'\vec{y}\rangle)
\end{aligned}
$$

(2) The inverse translations of the molecular forms of $\langle\vec{z}\rangle \cdot t$ and t', assuming $\vec{u}' : m$ are given below:

$$
\begin{aligned}
\widehat{[\![\langle\vec{z}\rangle \cdot t]\!]} &= \Uparrow_m(\vec{u}')(\langle\vec{w}\rangle \cdot \mathbf{out}_x \otimes \widehat{\mu} \otimes \langle\vec{v}\rangle) \\
\widehat{[\![t']\!]} &= \Uparrow_m(\vec{u}')(\widehat{\mu} \otimes \langle\vec{w}\vec{u}\rangle)
\end{aligned}
$$

By lemma A.5, together with the tensor and $\mathbf{out}_2$ rules, for any $\vec{y}, \vec{y}'$ such that $\{\vec{y}\} \cap \{\vec{v}\} = \emptyset$ and $\{\vec{y}\} = \{\vec{y}'\}$, we have:

$$
\vdash \langle\vec{w}\rangle \cdot \mathbf{out}_x \otimes \widehat{\mu} \otimes \langle\vec{v}\rangle \xrightarrow{(\vec{y})\langle\vec{y}'\vec{v}\rangle} \langle\vec{w}\rangle \otimes (\widehat{\mu} \cdot (\vec{y}')\langle\vec{y}\rangle)
$$

where $\vec{y} : l$, and $\{\vec{y}\} \cap \{x\vec{w}\} = \emptyset$. Choosing $\vec{y}, \vec{y}'$ such that $\{\vec{y}\} \cap \{\vec{u}\} = \emptyset$ and $(\vec{y}')\langle\vec{y}\rangle = (\vec{u}')\langle\vec{u}\rangle$, we can derive by lemma A.3, and the above transition:

$$
\vdash \Uparrow_m(\vec{u}')(\langle\vec{w}\rangle \cdot \mathbf{out}_x \otimes \widehat{\mu} \otimes \langle\vec{v}\rangle) \xrightarrow{(\vec{y})\sigma\overline{x}\langle\sigma\vec{v}\rangle}{}_{=} t''
$$

where $t'' = \Uparrow_m(\vec{u}')(((\langle\vec{w}\rangle \otimes (\widehat{\mu} \cdot (\vec{y}')\langle\vec{y}\rangle)) \cdot (\mathbf{id}_l \otimes (\vec{y})\langle\vec{y}'\vec{y}\rangle) \cdot (\mathbf{p}_{l,m} \otimes \mathbf{id}))$ and $\sigma = \{\vec{y}'/\vec{y}'\}$. But since $\{\vec{y}'/\vec{u}'\} = \{\vec{y}/\vec{u}\}$, we have $(\vec{y})\{\vec{y}'/\vec{u}'\}\overline{x}\langle\{\vec{y}'/\vec{u}'\}\vec{v}\rangle = (\vec{y})\{\vec{y}/\vec{u}\}\overline{x}\langle\{\vec{y}/\vec{u}\}\vec{v}\rangle = (\vec{u})\overline{x}\langle\vec{v}\rangle$. We shall now show that the residual is as expected:

$$
\begin{aligned}
&\Uparrow_m(\vec{u}')(\widehat{\mu} \otimes \langle\vec{w}\vec{u}\rangle) \\
&\quad = \Uparrow_m(\vec{u}')((\widehat{\mu} \cdot (\vec{y}')\langle\vec{y}'\rangle) \otimes \langle\vec{w}\vec{u}\rangle) && \text{2.16(5)} \\
&\quad = \Uparrow_m(\vec{u}')(\widehat{\mu} \cdot ((\vec{y}')\langle\vec{y}'\rangle \otimes \langle\vec{w}\vec{u}\rangle)) \\
&\quad = \Uparrow_m(\vec{u}')(\widehat{\mu} \cdot (\vec{y}')(\langle\vec{y}'\rangle \otimes \langle\vec{w}\vec{u}\rangle)) && \text{2.16(2)}
\end{aligned}
$$

$$= \uparrow_m(\vec{u}')(\widehat{\mu}\cdot(\vec{y}')((\vec{y}')\otimes\{\vec{y}'/\vec{u}'\}\langle\vec{w}\vec{u}\rangle)) \qquad \text{A.1}$$
$$= \uparrow_m(\vec{u}')(\widehat{\mu}\cdot(\vec{y}')((\vec{y}')\otimes\{\vec{y}'/\vec{u}'\}(\langle\vec{w}\rangle\otimes\{\vec{y}/\vec{u}\}\langle\vec{u}\rangle)))$$
$$= \uparrow_m(\vec{u}')(\widehat{\mu}\cdot(\vec{y}')((\vec{y}')\otimes\{\vec{y}'/\vec{u}'\}(\langle\vec{w}\rangle\otimes\{\vec{y}/\vec{u}\}\langle\vec{u}\rangle)))$$
$$= \uparrow_m(\vec{u}')(\widehat{\mu}\cdot(\vec{y}')((\vec{y}')\otimes\{\vec{y}'/\vec{u}'\}\langle\vec{w}\vec{y}\rangle))$$
$$= \uparrow_m(\vec{u}')(\widehat{\mu}\cdot(\vec{y}')\langle\vec{y}'\vec{w}\vec{y}\rangle) \qquad \text{A.1}$$
$$= \uparrow_m(\vec{u}')(\widehat{\mu}\cdot(\vec{y}')\langle\vec{y}'\vec{y}\rangle\cdot(\mathbf{id}_m\otimes\langle\vec{w}\rangle\otimes\mathbf{id}))$$
$$= \uparrow_m(\vec{u}')(\widehat{\mu}\cdot(\vec{y}')\langle\vec{y}\rangle\cdot(\vec{y})\langle\vec{y}'\vec{y}\rangle\cdot(\mathbf{id}_m\otimes\langle\vec{w}\rangle\otimes\mathbf{id}))$$
$$= \uparrow_m(\vec{u}')(((\langle\vec{w}\rangle\otimes(\widehat{\mu}\cdot(\vec{y}')\langle\vec{y}\rangle))\cdot(\mathbf{id}_l\otimes(\vec{y})\langle\vec{y}'\vec{y}\rangle)\cdot(\mathbf{p}_{l,m}\otimes\mathbf{id}))$$

(3) Consider the inverse translations of the molecular forms of $\langle\vec{z}\rangle\cdot t$ and t', assuming $\vec{u}_i' : m_i$ and $\vec{y}_a : k$.

$$\widehat{[\![\langle\vec{z}\rangle\cdot t]\!]} = \uparrow_{m_1\otimes m_2}(\vec{u}_1\vec{u}_2)(\mathbf{box}_x\widehat{a}\otimes\widehat{\mu}\otimes\langle\vec{v}\rangle)$$
$$\widehat{[\![t']\!]} = \uparrow_{m_1}(\vec{u}_1)\uparrow_{k\otimes m_2}(\vec{y}_a\vec{u}_2)(\vec{x}_a)(\widehat{\widehat{\lambda}}\otimes\widehat{\mu}\otimes\langle\vec{z}_a\vec{u}\rangle)$$

By the $\mathbf{box}_2$ rule and lemma A.4, we have $\vdash \mathbf{box}_x\widehat{a} \xrightarrow{(\vec{y}_1)x\langle\vec{y}_1\rangle} \widehat{a}\otimes(\vec{y}_1)\langle\vec{y}_1'\rangle$ where $\{\vec{y}_1\} = \{\vec{y}_1'\}$ and $x \notin \{\vec{y}_1'\}$. Then, by lemma A.5, together with the tensor rule, for any $\vec{y},\vec{y}'$ such that $\{\vec{y}\}\cap\{\vec{v}\} = \emptyset$ $\{\vec{y}\} = \{\vec{y}'\}$, and $\{\}\cap(\mathbf{fn}\,(\mathbf{box}_x\widehat{a})\cup\mathbf{fn}\,(\widehat{\mu}\otimes\langle\vec{v}\rangle)) = \emptyset$ we have the following transition:

$$\vdash \mathbf{box}_x\widehat{a}\otimes\widehat{\mu}\otimes\langle\vec{v}\rangle \xrightarrow{(\vec{y})x\langle\vec{y}'\vec{v}\rangle}_= (\widehat{a}\otimes\widehat{\mu})\cdot(\vec{y}')\langle\vec{y}\rangle$$

Choosing $\vec{y},\vec{y}'$ such that $\{\vec{y}\}\cap\{\vec{u}\} = \emptyset$ and $(\vec{y}')\langle\vec{y}\rangle = (\vec{u}_1\vec{u}_2)\langle\vec{u}\rangle$, we can derive by lemma A.3, and the above transition:

$$\vdash \uparrow_{m_1\otimes m_2}(\vec{u}_1\vec{u}_2)(\mathbf{box}_x\widehat{a}\otimes\widehat{mu}\otimes\langle\vec{v}\rangle) \xrightarrow{(\vec{y})\sigma x\langle\sigma\vec{v}\rangle}_= t''$$

where $t'' = \uparrow_{m_1\otimes m_2}(\vec{u}_1\vec{u}_2)((\widehat{a}\otimes\widehat{\mu})\cdot(\vec{y}')\langle\vec{y}\rangle\cdot(\vec{y})\langle\vec{y}'\vec{y}\rangle)$ and $\sigma = \{\vec{y}'/\vec{u}_1\vec{u}_2\}$. But since $\{\vec{y}'/\vec{u}_1\vec{u}_2\} = \{\vec{y}/\vec{u}\}$, we have $(\vec{y})\sigma x\langle\sigma\vec{v}\rangle = (\vec{y})\{\vec{y}/\vec{u}\}x\langle\{\vec{y}/\vec{u}\}\vec{v}\rangle = (\vec{u})x\langle\vec{v}\rangle$. We shall now show that the residual is as expected:

$$
\begin{aligned}
&\uparrow_{m_1}(\vec{u}_1)\uparrow_{k\otimes m_2}(\vec{y}_a\vec{u}_2)(\vec{x}_a)(\hat{\vec{\lambda}}\otimes\hat{\mu}\otimes\langle\vec{z}_a\vec{u}\rangle)\\
&\quad= \uparrow_{m_1}(\vec{u}_1)\uparrow_{m_2}(\vec{u}_2)\uparrow_k(\vec{y}_a)(\vec{x}_a)(\hat{\vec{\lambda}}\otimes\hat{\mu}\otimes\langle\vec{z}_a\vec{u}\rangle) && 3.26^*\\
&\quad= \uparrow_{m_2}(\vec{u}_2)\uparrow_{m_1}(\vec{u}_1)(\uparrow_k(\vec{y}_a)(\vec{x}_a)(\hat{\vec{\lambda}}\otimes\hat{\vec{\mu}}\otimes\langle\vec{z}_a\vec{u}\rangle)\\
&\qquad\qquad\cdot(\mathbf{p}_{m_2,m_1}\otimes\mathbf{id})) && 3.26^*\\
&\quad= \uparrow_{m_2}(\vec{u}_2)\uparrow_{m_1}(\vec{u}_1)\uparrow_k(\vec{y}_a)(\vec{x}_a)((\hat{\vec{\lambda}}\otimes\hat{\vec{\mu}}\otimes\langle\vec{z}_a\vec{u}\rangle)\\
&\qquad\qquad\cdot(\mathbf{id}_k\otimes\mathbf{p}_{m_2,m_1}\otimes\mathbf{id})) && 2.16(1)^*,\ 3.9(4)\\
&\quad= \uparrow_{m_2}(\vec{u}_2)\uparrow_{m_1}(\vec{u}_1)\uparrow_k(\vec{y}_a)(\vec{x}_a)(\hat{\vec{\lambda}}\otimes\langle\vec{z}_a\rangle\otimes\hat{\vec{\mu}}\otimes\langle\vec{u}\rangle)\\
&\quad= \uparrow_{m_2}(\vec{u}_2)\uparrow_{m_1}(\vec{u}_1)\uparrow_k(\vec{y}_a)(\vec{x}_a)((\hat{\vec{\lambda}}\otimes\langle\vec{z}_a\rangle\otimes\hat{\vec{\mu}})\\
&\qquad\qquad\cdot(\mathbf{id}_k\otimes(\vec{y}')(\langle\vec{y}'\rangle\otimes\langle\vec{u}\rangle))) && 2.16(5,2)^*\\
&\quad= \uparrow_{m_2}(\vec{u}_2)\uparrow_{m_1}(\vec{u}_1)(\uparrow_k(\vec{y}_a)(\vec{x}_a)(\hat{\vec{\lambda}}\otimes\langle\vec{z}_a\rangle\otimes\hat{\vec{\mu}})\\
&\qquad\qquad\cdot(\vec{y}')(\langle\vec{y}'\rangle\otimes\langle\vec{u}\rangle)) && 3.9(4)\\
&\quad= \uparrow_{m_2}(\vec{u}_2)\uparrow_{m_1}(\vec{u}_1)((\hat{\vec{a}}\otimes\hat{\vec{\mu}})\cdot(\vec{y}')(\langle\vec{y}'\rangle\otimes\langle\vec{u}\rangle))\\
&\quad= \uparrow_{m_2}(\vec{u}_2)\uparrow_{m_1}(\vec{u}_1)((\hat{\vec{a}}\otimes\hat{\vec{\mu}})\cdot(\vec{y}')(\langle\vec{y}'\rangle\otimes\{\vec{y}'/\vec{u}_1\vec{u}_2\}\langle\vec{u}\rangle)) && A.1\\
&\quad= \uparrow_{m_2}(\vec{u}_2)\uparrow_{m_1}(\vec{u}_1)((\hat{\vec{a}}\otimes\hat{\vec{\mu}})\cdot(\vec{y}')(\langle\vec{y}'\rangle\otimes\{\vec{y}/\vec{u}\}\langle\vec{u}\rangle))\\
&\quad= \uparrow_{m_2}(\vec{u}_2)\uparrow_{m_1}(\vec{u}_1)((\hat{\vec{a}}\otimes\hat{\vec{\mu}})\cdot(\vec{y}')(\langle\vec{y}'y\rangle))\\
&\quad= \uparrow_{m_2}(\vec{u}_2)\uparrow_{m_1}(\vec{u}_1)((\hat{\vec{a}}\otimes\hat{\vec{\mu}})\cdot(\vec{y}')(\langle\vec{y}'y\rangle))\\
&\quad= \uparrow_{m_2}(\vec{u}_2)\uparrow_{m_1}(\vec{u}_1)((\hat{\vec{a}}\otimes\hat{\vec{\mu}})\cdot(\vec{y}')\langle\vec{y}\rangle\cdot(\vec{y})\langle\vec{y}'\vec{y}\rangle)\\
&\quad= \uparrow_{m_1\otimes m_2}(\vec{u}_1\vec{u}_2)((\hat{\vec{a}}\otimes\hat{\vec{\mu}})\cdot(\vec{y}')\langle\vec{y}\rangle\cdot(\vec{y})\langle\vec{y}'\vec{y}\rangle) && 3.26^*
\end{aligned}
$$

■

Index