Contributions to a Trace Theory beyond Mazurkiewicz Traces

HABILITATIONSSCHRIFT

zur Erlangung des akademischen Grades

Doctor rerum naturalium habilitatus
(Dr. rer. nat. habil.)

vorgelegt

der Fakultät Mathematik und Naturwissenschaften
der Technischen Universität Dresden

von

Dr. rer. nat. Dietrich Kuske
geboren am 8. Juni 1965 in Kühlungsborn

Gutachter: Prof. Dr. Volker Diekert
Prof. Dr. Manfred Droste
Prof. Dr. Wolfgang Thomas

Eingereicht am: 20. Dezember 1999

Tag der Verteidigung: 21. Dezember 2000

URN: urn:nbn:de:gbv:ilm1-2011200051

Contents

Introduction

To understand the behavior of contemporary computing devices, the concept of parallelism or concurrency is inevitable. There are several obvious reasons for an increasing use of these techniques: In an attempt to make programs faster one may distribute them over several executing machines. By duplicating memory or computation tasks, the reliability of systems can be increased. On a certain level of abstraction, a specification is inherently concurrent since the subsystems are thought to run independently from each other. Another aspect is that communication networks consist by definition of independent subsystems that are only loosely coupled. These observations call for a deeper understanding of the mechanisms involved.

For sequential systems, a mathematical foundation has proved fruitful. Already the consideration of formal systems in the first half of this century laid the ground for an distinction between (theoretically) computable and definitely not mechanizable tasks. Complexity theory sharpens this distinction further by the investigation of the frontier between tractable and nontractable computational tasks. Finite automata, although they are a very restricted model of a sequential computing device, have a rich theory as well as a wide-spread application. Their theory is closely related to algebraic theories. Furthermore, surprising connections between different logics and automata were found. These connections make it possible to automatize certain verification tasks in the development of software and hardware systems.

Aiming at similar benefits, attempts to develop a mathematical formalization of parallelism have a longstanding tradition in computer science. In the 60s, Petri introduced nets, now called Petri nets, as a model of concurrent systems. These Petri nets inspired many theoretical investigations and now have an extensive theory. But the semantics of these nets is technically quite complicated and a mathematical treatment in its full generality turns out to be cumbersome. Another line of research in this area is based on the idea of a process algebra introduced by Milner and Hoare in the 70s. This approach focuses more on at programming languages. Cellular automata can be traced back to ideas of v. Neumann but became widely known only in the 70s (in particular by Conway's "Game of Life"). Now they enjoy a well understood theory as well as several extensions.

Mazurkiewicz introduced traces, another model of concurrent behaviors, into computer science. They can be defined in two equivalent ways, either as dependence graphs or as equivalence classes of words. In both cases, one starts out from a finite set of elementary or atomic actions, called alphabet, and a binary dependence relation on the set of actions. Two such actions are dependent if they e.g. use a common resource. Hence, in a parallel computation of the system, independent actions can be performed in parallel, while dependent actions can be performed sequentially, only. A computation of such a system is modeled as a directed graph. The vertices of this graph correspond to events. Two such vertices are connected by an edge iff their labels are dependent. Since the computation is meant to run in time, the graph is assumed to be acyclic. Furthermore, we consider only finite computations and therefore finite graphs. A dependence graph is nothing but such a finite directed acyclic graph with edges between dependent vertices. Thus, a dependence graph describes the causal dependence in a computation.

In the alternative definition, one considers sequential observations of some parallel computation. The order in which independent actions are performed is regarded as irrelevant. In particular, if two observations differ only in the order of independent actions, they are identified. This defines an equivalence relation on words (over the alphabet of actions) and a trace is an equivalence class with respect to this equivalence.

It turns out that the linear extensions of a dependence graph form an equivalence class, i.e. a trace, and that any trace can be obtained from some dependence graph. In this sense, the two approaches are equivalent and "it is only a matter of taste which objects are chosen for representing concurrent processes: equivalence classes of strings or labelled graphs." [Maz95, page 14]. It seems that this dual nature of traces has contributed to a much to their success. This is not the right place to recall the amount of results on traces that have been obtained in the last two decades. For a in-depth surveys on the results on traces, the reader is referred to [DR95] that concentrates on the theoretical aspects in computer science as well as in mathematics, in particular in combinatorics.

Nonetheless, it turned out that certain limitations of traces made it necessary to extend the model into different directions. The probably most modest extension was that to infinite or real traces. These were introduced to model not only finite but also infinite computations. They can be defined in several equivalent ways: as directed and downward closed sets of finite traces [Maz87], via an equivalence relation on infinite words [Sta89, Kwi90] or as infinite dependence graphs where any event dominates only a finite number of vertices. Diekert introduced α- and δ-complex traces as metric completion of the set of finite traces with respect to two different metrics [Die91, Die93] and showed in particular that they can alternatively be defined as infinite dependence graphs with some alphabetic information. Most of the considerations on complex traces are based on this second characterization. Another similar extension of traces (approximating traces) is presented in the same spirit [DG98].

The generalizations mentioned so far have been introduced to model infinite behaviors of a parallel system. Differently, the aim of semi-commutations is to model some behaviors like the producer-consumer-example that do not fit into the setting of a symmetric independence relation. The idea is that the consumption of an item can be delayed after further productions, but conversely, the production cannot be postponed after the consumption. Here, we refer the reader to the survey [CLR95] and the literature cited therein.

Another limitation of Mazurkiewicz traces is the global and fixed independence relation. There are certainly systems where the answer to the question whether two particular actions can be performed concurrently depends on the situation, e.g. on the available resources that are produced by preceding events (cf. [KP92]). An automaton with concurrency relations [Dro90, Dro92] is a (finite) automaton whose states are equipped with independence relations, i.e. in this model the dependence of actions can change while the system evolves. Similarly to traces, one obtains an equivalence relation on the set of finite computation sequences by identifying those sequences that differ only in the order of independent actions. But now this independence refers to the state were the first of the two actions is performed. Thus, originally the behavior of an automaton with concurrency relations was defined via equivalence classes of sequential behaviors. In [BDK95, BDK97], representing these computations by dependence graphs, we presented a partial order semantics for these computations under some mild assumptions on the automaton with concurrency relations.

Another approach to incorporate changing independence relations into the model of traces is represented by context and generalized traces [BR94]. Here, two actions might be independent in one context and dependent in another where the context is (in the simplest form) given by the preceding action. Again, first an equivalence of words was constructed and context traces were defined as equivalence classes of words. An attempt to represent context traces by dependence graphs succeeded only partially [BR95].

Common to all generalizations listed so far is that the independence of actions is a binary relation. This limits their applicability since it is not possible to model a situation were two items of some resource are claimed by three actions. In such a situation, any two of the claiming actions might be performed concurrently and the third one afterwards. In addition, traces and their successors do not allow autoconcurrency. Local traces [HKT92, Hoo94] are an attempt to solve these problems. Here, sets or even multisets of actions are declared independent and this depends on the history of the system. A representation of such systems by local event structures was obtained in the same papers. The forthcoming work [KM00] aims at a representation of computations in this model by dependence graphs.

Note that in all the extensions mentioned so far, computations were first modeled as equivalence classes of sequential executions. Later (for some models much later) it was shown that these equivalence classes can be nicely represented by structures like dependence graphs. Differently, P-traces are by definition labeled partially ordered sets. Afterwards it is shown that they can also be obtained as equivalence classes of certain equivalence relations [Arn91].

Besides this duality, the different extensions of Mazurkiewicz traces have been considered under several aspects. Mazurkiewicz used traces to model the behavior of one-safe Petri nets. Categorical adjunctions were constructed between larger classes of Petri nets and trace structures [NRT90], step transition systems (i.e. local traces) [Muk92] and concurrent automata [DS93]. The order theoretic properties of the set of all trace-like objects was investigated for real traces [GR93, BCS93, Kus99], for several versions of complex traces [GP92, Teo93, DG98] and for the computations of an automaton with concurrency relations [Sta89, Dro90, Dro92, Kus94a, Kus94b, Sch98]. Metric and topological questions were dealt with for real traces [Kwi90, KK00], for complex and approximating traces [Die91, Die93, DG98] and for computations of automata with concurrency relations [KS98]. The recognizable sets of trace-like structures were studied thoroughly. The relation to rational sets was investigated for semicommutations, for real and for complex traces (cf. the corresponding surveys in [DR95]) and for computations of concurrent automata [Dro94, Dro95, Dro96]. The relation to logically axiomatizable sets can be found for finite and for real traces in [Tho90b, EM93, Ebi94], for computations of concurrent automata in [DK96, DK98] and for local traces in [KM00].

In the first part of the current work, we will define an extension of dependence graphs to so called Σ-dags where Σ is a finite set of actions. They generalize not only dependence graphs as defined above, but also CCI-sets [Arn91], dependence graphs of computations of concurrent automata [BDK95, BDK97], and (width-bounded) sp-pomsets [LW98b, LW98a, LW00]. Essentially, a Σ-dag is a Σ-labeled directed acyclic graph. The edges of this graph represent the causal dependency between the events that are modeled by the vertices. There are only two restrictions that we impose: First, we allow no autoconcurrency. Second, for any label a, an event can depend on and influence at most one a-labeled event directly.

As a computational model for these Σ-dags, we investigate asynchronous cellular automata. They were defined originally for dependence graphs as a truly parallel accepting device [Zie87].[1] Since then, they have been intensively studied, cf. [Zie95, DM95] for overviews. In [DG96], they were generalized in such a way that an asynchronous cellular automaton can accept labeled posets (pomsets) without autoconcurrency (cf. also [Kus98, DGK00]). Here, we extend them to the setting of Σ-dags. In the literature, infinite state systems are intensively studied [Mol96, BE97]. We extend asynchronous cellular automata furthermore by allowing them to have infinitely many states. To preserve some finiteness,

[1]The name might be misleading since these automata are not a generalization of v. Neumann's cellular automata mentioned above.

the set of states is endowed with a well-quasi ordering. Thus, loosely speaking, asynchronous cellular machines or Σ-ACMs are asynchronous cellular automata that run on Σ-dags, have possibly infinitely many states, and are equipped with a well-quasi ordering on these states.

The behavior of a Σ-ACM is the accepted language, i.e. a set of Σ-dags. Hence a Σ-ACM describes a property of Σ-dags. Since the intersection as well as the union of two acceptable sets can be accepted by a Σ-ACM, properties describable by Σ-ACMs can become quite complex. Then it is of interest whether the combined property is contradictory, or, equivalently, whether at least one Σ-dag satisfies it. Thus, one would like to know whether a Σ-ACM accepts at least one Σ-dag. Using a result from [FS98, FS01], we show that it is possible to gain this knowledge even automatically, i.e. we show that there exists an algorithm that on input of a Σ-ACM decides whether the Σ-ACM accepts at least one Σ-dag. For this to hold, we restrict the asynchronous cellular machines in two ways: The notion of "monotonicity" involves a connection between the well-quasi ordering and the transitions of the machine. The notion "effectiveness" requires that the machine is given in a certain finite way.

Another natural question is whether two properties are equivalent, i.e. whether two Σ-ACMs accept the same language. Since there is a Σ-ACM that accepts all Σ-dags, a special case of this equivalence problem is to ask whether a given Σ-ACM accepts all Σ-dags. The latter question, called universality, essentially asks whether the described property is always true. The corresponding question for sequential automata has a positive answer which is a consequence of the decidability of the emptiness: If one wants to know whether a sequential automaton accepts all words, one constructs the complementary automaton and checks whether its languages is empty. Thus, the crucial point for sequential automata is that they can effectively be complemented. But the set of acceptable Σ-dag-languages is not closed under complementation. Therefore, we cannot proceed as for sequential automata. On the contrary, we show that the universality is undecidable even for Σ-ACMs with only finitely many states. These finite Σ-ACMs are called asynchronous cellular automata or Σ-ACA. The undecidability of the universality implies that the equivalence of two Σ-ACAs, the complementability of a Σ-ACA as well as the existence of an equivalent deterministic Σ-ACA are undecidable, too. These results (restricted to pomsets) together with a sketch of proof were announced in [Kus98]. The proof we give here is based on ideas developed together with Paul Gastin.

The following chapter deals with the question which properties can be expressed by a Σ-ACA. For finite sequential automata, several answers are known to the question which properties can be checked by a finite sequential automaton: Kleene showed that these are precisely the rational properties. By the Myhill-Nerode Theorem, a property can be checked by a finite sequential automaton if its syntactic monoid is finite. Furthermore, Büchi and Elgot [Büc60, Elg61] showed that a property of words can be checked by a finite sequential automaton if it can be expressed in the monadic second order logic. This relation between a model of a computational device (finite sequential automata) and monadic second order logic is a paradigmatic result. It has been extended in several directions, e.g. to infinite words [Büc60], to trees [Rab69] (cf. also [Tho90a]), to finite [Tho90b] and to real [EM93, Ebi94] traces, and to computations of concurrent automata [DK96, DK98]. The celebrated theorem of Zielonka [Zie87, Zie95] together with the results from [Tho90b] states that for dependence graphs of traces, the expressive power of asynchronous cellular automata and monadic second order logic coincide. Aiming at a similar result for Σ-dags, in Chapter 5 we show that this is not possible in general. More precisely, we show that any recognizable set of Σ-dags can be axiomatized by a sentence of the monadic second order logic, but that the converse is false even for first-order logic. To overcome this, we restrict to a subclass of all Σ-dags, called (Σ, k)-dags. This restriction makes it possible to relabel a (Σ, k)-dag by an asynchronous cellular automaton in such a way that one obtains a dependence graph over a certain dependence alphabet. This is the crucial step in our proof that any monadically axiomatizable set of (Σ, k)-dags can be accepted by a (nondeterministic) asynchronous cellular automaton. But we show that it is necessary to allow nondeterminism in the automata since the expressive power of deterministic Σ-ACAs will be proved to be strictly weaker. Again, the restriction to pomsets of the results presented in this chapter can be found in [Kus98]. Here, we generalize the presentation in [DGK00].

The final chapter of the first part is devoted to the relation between our asynchronous cellular automata and other models of concurrent behavior. The covering relation of a pomset without autoconcurrency is a Σ-dag. This allows us to speak of the set of pomsets that is accepted by a Σ-ACA: A pomset $(V, \leq, \lambda)$ is accepted iff its Hasse-diagram $(V, -\!\!<, \lambda)$ admits a successful run. For pomsets, other automata models have been proposed in the literature. In particular, Arnold considered P-asynchronous automata [Arn91] and Lodaya and Weil dealt with branching automata [LW98a, LW98b, LW00]. We finish our consideration of Σ-dags and Σ-ACAs by a comparison of the expressive power of these automata with the expressive power of our Σ-ACAs. We show that branching automata when restricted to width-bounded languages have the same expressive power as Σ-ACAs when restricted to series-rational pomsets. Somewhat as a byproduct, this implies that the expressive power of branching automata on width bounded sp-pomsets coincides with the expressive power of the monadic second order logic. Finally, we show that any P-asynchronous automaton can be simulated by a Σ-ACA.

The Σ-dags considered in the first part of the current work are clearly labeled graphs. Above, I already cited A. Mazurkiewicz stating “it is only a matter of taste which objects are chosen for representing concurrent processes: equivalence classes of strings or labelled graphs.” [Maz95, page 14]. To satisfy those that prefer the algebraic approach (or at least appreciate it as the author), this is followed in the second part where left divisibility monoids are considered. These left divisibility monoids were introduced in [DK99, DK01]. As pointed out earlier, trace monoids are defined via a finite presentation (using a set of letters Σ together with a dependence relation on Σ). Later, algebraic properties where discovered that characterize trace monoids (up to isomorphism) [Dub86]. Differently, left divisibility monoids are defined in the language of monoids, i.e. via their algebraic properties. In particular, it is required that the prefix relation be a partial order and that for any monoid element, the set of prefixes forms a distributive lattice. Thus, divisibility monoids involve monoid theoretic as well as order theoretic concepts.

In Chapter 8, we show that divisibility monoids can be finitely presented. Not only will we show that this is possible in general, but we will give a concrete representation for any divisibility monoid. Finally, we give a decidable class of finite presentations that give rise to all divisibility monoids.

Kleene's theorem on recognizable languages of finite words has been generalized in several directions, e.g. to formal power series [Sch61], to infinite words [Büc60], and to infinite trees [Rab69]. More recently, rational monoids were investigated [Sak87], in which the recognizable languages coincide with the rational ones. Building on results from [CP85, CM88, Mét86], a complete characterization of the recognizable languages in a trace monoid by c-rational sets was obtained in [Och85]. A further generalization of Kleene's and Ochmański's results to concurrency monoids was given in [Dro95]. In Chapter 9, we derive such a result for divisibility monoids. The proofs by Ochmański [Och85] and by Droste [Dro95] rely on the *internal* structure of the elements of the monoids. Here, we do not use the internal representation of the monoid elements, but algebraic properties of the monoid itself. Thus, the considerations in Chapter 9 that appeared in [DK99] can be seen as an algebraic proof of Ochmański's Theorem.

The following chapter is devoted to the question when a divisibility monoid satisfies Kleene's Theorem, i.e. when the rational and the recognizable sets coincide. For trace monoids, this is only the case if the trace monoid is free. Our result for divisibility monoids states that they satisfy Kleene's Theorem iff they are rational. A defining property of divisibility monoids is that the sets of prefixes form a distributive lattice for any element of the monoid. We prove that this set of distributive lattices is width-bounded iff the monoid satisfies Kleene's Theorem. We obtain these characterizations applying the theory of rational functions (cf. [Ber79]) and a Foata normal form of monoid elements similar to that for traces.

Büchi showed that the monadic second order theory of the linearly ordered set $(\omega, \leq)$ is decidable. To achieve this goal, he used finite automata. In the course of these considerations he showed that a language in a free finitely generated monoid is recognizable iff it is monadically axiomatizable. In computer science, this latter result and its extension to infinite words are often referred to as "Büchi's Theorem" while in logic this term denotes the decidability of the monadic theory of ω. In the final chapter, I understand it in this second meaning. There, we show that certain monadic theories associated to a divisibility monoid are decidable. Let $\mathfrak{L}$ denote the set of distributive lattices associated to a given divisibility monoid. We show that the monadic theory of this class is decidable iff the monoid satisfies Kleene's Theorem. In general, this theory is undecidable, but the monadic theory of the join-irreducible elements of these lattices is still decidable. For trace monoids, this latter result just states that the monadic theory of all dependence graphs is decidable, a corollary from [EM93, Ebi94].

At the very end, we prove an order theoretic result that is inspired by the two decidabilities just mentioned: Together with a result from Chapter 10, we know that the monadic theory of $\mathfrak{L}$ is decidable if and only if $\mathfrak{L}$ is width-bounded. In a certain sense, we show that this does not depend on the special character of $\mathfrak{L}$ as the set of lattices associated with a divisibility monoid. Indeed, we show that any set of finite distributive lattices $\mathfrak{L}$ has a decidable monadic theory if and only if the monadic theory of the join-irreducible elements of these lattices is decidable and $\mathfrak{L}$ is width-bounded.

The present work shows that there are deep connections that arise from the theory of traces to different branches of mathematics. We finish the work with a list of problems that show up in the course of our considerations.

Chapter 1

Basic definitions

1.1 Order theoretic definitions

1.1.1 Well quasi orders

Let A be a set. A *quasi order* on A is a binary relation $\preceq \subseteq A \times A$ that is transitive and reflexive. The tuple $(A, \preceq)$ is called *quasi ordered set.* So let $(A, \preceq)$ be a quasi ordered set, $a \in A$ and $X \subseteq A$. Then we define $\uparrow a := \{b \in A \mid a \preceq b\}$ and $\uparrow X := \bigcup_{x \in X} \uparrow x$. A set $B \subseteq A$ is a *basis of* X if $\uparrow B = \uparrow X$. Note that any set X has a basis, namely itself or $\uparrow X$. In the literature, one often defines a basis for sets X with $X = \uparrow X$, only, but in our context, it is more convenient to extend the classical definition slightly.

We call a sequence $(a_i)_{i \in \mathbb{N}}$ in a quasi ordered set $(A, \preceq)$ *good* if there are $i < j$ with $a_i \preceq a_j$. If no such indices exist, the sequence is *bad.* A *well quasi order* is a quasi order $\preceq$ on a set A where any sequence in A is good. A *wqo* is a quasi ordered set $(A, \preceq)$ where $\preceq$ is a well quasi order. Occasionally, we use wqo as an abbreviation of well quasi order, too.

Let $(A, \preceq)$ be a wqo and $a_i \in A$ for $i \in \mathbb{N}$. Let M consist of all indices $i \in \mathbb{N}$ such that $x_i \not\preceq x_j$ for any $j > i$. Since $(A, \preceq)$ is a wqo, this set is finite. Choose $i_0 \in \mathbb{N}$ with $M \leq i_0$. Then, inductively, we find $i_{n+1} > i_n$ with $a_{i_n} \preceq a_{i_{n+1}}$, i.e. the sequence $(a_i)_{i \in \mathbb{N}}$ contains an infinite non-decreasing subsequence. Now let $X \subseteq A$. An element $x \in X$ is *minimal in* X if for any $y \in X$ with $y \preceq x$ we get $x \preceq y$. By $\min(X)$, we denote the set of minimal elements of X. Let $\sim = \preceq \cap \succeq$. Since $\preceq$ is transitive and reflexive, $\sim$ is an equivalence relation. Note that $\uparrow x = \uparrow y$ for $x, y \in A$ whenever $x \sim y$. Let $(x_i)_{i \in \alpha}$ be an enumeration of $\min(X)$ for some ordinal α. Furthermore, let $i_0 = 0$. Inductively, let $n \in \mathbb{N}$ and assume that $i_n \in \alpha$ is chosen. If there exists $i > i_n$ such that $x_i \not\sim x_{i_j}$ for $0 \leq j \leq n$, let i_{n+1} be the minimal such i. If this construction does not terminate, we get a sequence $(x_{i_n})_{n \in \mathbb{N}}$ with $x_{i_n} \not\sim x_{i_m}$ for $n < m$. Since $(A, \preceq)$ is a wqo, there is $n < m$ with $x_{i_n} \preceq x_{i_m}$. Since $x_{i_m} \in \min(X)$, this implies $x_{i_m} \preceq x_{i_n}$, contradicting the choice of i_m. Thus, there is $k \in \mathbb{N}$ such that we find for $x \in \min(X)$ an index $0 \leq j \leq k$

with $x \sim x_{i_j}$. Now let $y \in X$. Then there exists $x \in \min(X)$ with $x \preceq y$ for otherwise we found an infinite sequence $(y_i)_{i\in\mathbb{N}}$ with $y_i \succ y_{i+1}$, i.e. in particular with $y_i \not\preceq y_j$ for $i < j$. Thus, the set $\{x_{i_0}, x_{i_1}, \dots, x_{i_k}\}$ is a finite basis of X, i.e. we showed that any set $X \subseteq A$ contains a finite basis.

Next, we want to define from a wqo $(A, \preceq)$ a quasi order on the set of finite words over A. So let $v = a_1a_2\dots a_n$ and $w = b_1b_2\dots b_m$ be words over A. We define $v \preceq^\star w$ iff there exists a sequence $0 < i_1 < i_2 < \dots < i_n \leq m$ such that $a_j \preceq b_{i_j}$ for $1 \leq j \leq n$, i.e. if v is dominated by some subword of w letter by letter. Clearly, $\preceq^\star$ is transitive and reflexive, i.e. it is a quasi order on the set of words $A^\star$ over A.

Higman's Theorem [Hig52] $(A^\star, \preceq^\star)$ is a wqo.

Proof.[1] By contradiction suppose $\preceq^\star$ is no wqo. Then there exists a bad sequence in $A^\star$. Let v_0 be a word of minimal length such that there exists a bad sequence $(w_i)_{i\in\mathbb{N}}$ with $w_0 = v_0$. Inductively, assume we found $v_0, v_1, \dots, v_n \in A^\star$ such that there exists a bad sequence starting with these words. Then let $v_{n+1} \in A^\star$ be a word of minimal length such that $v_0, v_1, \dots, v_n, v_{n+1}$ can be extended to a bad sequence. Note that in particular $v_i \not\preceq^\star v_{n+1}$ for $0 \leq i \leq n$. This construction results in a bad sequence $(v_i)_{i\in\mathbb{N}}$ such that, for any $i \in \mathbb{N}$ and word $w \in A^\star$ shorter than v_i, the tuple $v_0, v_1, \dots, v_{i-1}, w$ cannot be extended to a bad sequence. Since the empty word is dominated by any word, in addition none of these words is empty. For $i \in \mathbb{N}$ let $a_i \in A$ be the first letter of v_i and let w_i be the remaining word, i.e. $a_iw_i = v_i$. Since $(A, \preceq)$ is a wqo, the sequence $(a_i)_{i\in\mathbb{N}}$ contains an infinite non-decreasing subsequence $(a_{i_j})_{j\in\mathbb{N}}$. Now consider the sequence

$$v_0, v_1, \dots, v_{i_0-1}, w_{i_0}, w_{i_1}, w_{i_2}, \dots$$

in $A^\star$. For $1 \leq i < j < i_0$, we have $v_i \not\preceq^\star v_j$ since the words v_n form a bad sequence. For $1 \leq i < i_0$ and $j \in \mathbb{N}$, we get $v_i \not\preceq^\star w_{i_j}$ for otherwise $v_i \preceq^\star a_{i_j}w_{i_j} = v_{i_j}$, contradicting that the words v_n form a bad sequence. Now let $i < j$ and assume $w_{i_i} \preceq^\star w_{i_j}$. Since $a_{i_i} \preceq a_{i_j}$, this implies $v_{i_i} = a_{i_i}w_{i_i} \preceq^\star a_{i_j}w_{i_j} = v_{i_j}$, again a contradiction. Hence the sequence above is bad. But this contradicts the fact that v_{i_0} is properly longer than w_{i_0} and that by our choice of v_{i_0}, the tuple $v_0, v_1, \dots, v_{i_0-1}, w_{i_0}$ cannot be extended to a bad sequence. Thus, indeed, $\preceq^\star$ is a wqo on the set of finite words over A. □

[1] This proof of Higman's theorem follows a proof given in [Die96] where the idea of a minimal bad subsequence is attributed to Nash-Williams [NW63].

1.1.2 Partial orders

Let A be a set. A quasi order $\leq$ on A is a *(partial) order* if it is antisymmetric. Then $(A, \leq)$ is a *partially ordered set* or *poset* for short. Two elements $a, b \in A$ are *incomparable* (denoted $a \parallel b$) if neither $a \leq b$ nor $b \leq a$. By $\lessgtr$, we denote the union of $<$ and $>$. Hence $a \not\lessgtr b$ iff $a \parallel b$ or $a = b$. An element $c \in A$ *covers* $a \in A$ iff $a < c$ and if $a < b \leq c$ implies $b = c$. We write $a \prec c$ whenever a is covered by c.

The set A is an *antichain* if any two distinct elements of A are incomparable. If, on the contrary, any two of its elements are comparable (i.e. not incomparable), then A is *linearly ordered* or a *chain*. An (anti-)chain X in $(A, \leq)$ is a subset $X \subseteq A$ such that $(X, \leq \cap X \times X)$ is an (anti-)chain. The set $X \subseteq A$ is *convex* if for any $x \leq y \leq z$ with $x, z \in X$ the element y belongs to X, too. A nonempty subset X of A is a *filter* if $x \in X$ and $x \leq y$ imply $y \in X$. Dually, a nonempty set $X \subseteq A$ is an *ideal* if $x \in X$ and $x \geq y$ imply $y \in X$. Since traditionally ideals were called heriditary set, the set of ideals of $(A, \leq)$ is denoted by $\mathbb{H}(A, \leq)$.

Recall that $\uparrow a = \{b \in A \mid a \leq b\}$. We call this set the *principal filter generated by* a. Dually, $\downarrow a = \{b \in A \mid a \geq b\}$ is the *principal ideal generated by* a. By $\updownarrow a$, we denote the union of $\uparrow a$ and $\downarrow a$, i.e. the set of elements of A that are comparable with a. The intersection of $\uparrow a$ and $\downarrow b$ is denoted by $[a, b]$. It is the interval with endpoints a and b. Note that this interval is nonempty iff $a \leq b$. For $X \subseteq A$, let $\uparrow X := \bigcup_{x \in X} \uparrow x$ and dually $\downarrow X := \bigcup_{x \in X} \downarrow x$ denote the generated filter and ideal, respectively. An ideal I is finitely generated if there exists a finite set X such that $I = \downarrow X$. The set of finitely generated ideals will be denoted by $\mathbb{H}_f(A, \leq)$.

For $X \subseteq A$ and $a \in A$, we write $X \leq a$ whenever $x \leq a$ for all $x \in X$. In this case a is an *upper bound* of X. It is a *minimal upper bound* if $X \leq x \leq a$ implies $x = a$. By $\text{mub}(X)$, we denote the set of minimal upper bounds of X. An upper bound a of X that is dominated by any upper bound of X is the *least upper bound*, *supremum* or *join* of X. It is denoted by $\sup(X)$ or $\bigvee X$. The supremum of a two-elements set $\{a, b\}$ is denoted by $a \vee b$. Dually, *lower bound*, *maximal lower bound*, *largest lower bound* or *infimum* or *meet* are defined. The infimum of $X \subseteq A$ is denoted by $\inf(X)$, $\bigwedge(X)$ or $a \wedge b$ if $X = \{a, b\}$. An element $a \in A$ is *join-irreducible* if $x \vee y = a$ implies $a \in \{x, y\}$ and $a \not\leq A$. By $\mathbb{J}(A, \leq)$, we denote the set of join-irreducible elements of A.

Let $(A, \leq)$ be a poset and $a \in A$. The *width* $w(A, \leq)$ *of* $(A, \leq)$ is the supremum of the sizes of all antichains in A. The *height of* a is the supremum of all sizes of chains $C < a$. We denote the height of a in $(A, \leq)$ by $h(a, (A, \leq))$ or shorter by $h(a, A)$ or just by $h(a)$. Note that the minimal elements of a poset have height 0. The *length of* $(A, \leq)$ is the supremum of the heights of the elements of A.

A partially ordered set $(A, \leq)$ is a *join-semilattice* iff any finite subset of A has a supremum. It is a *lattice* if in addition any finite subset of A has an infimum. Note that if $(A, \leq)$ is a lattice so is $(A, \geq)$. Two intervals $[a, b]$ and $[a', b']$ in a lattice $(A, \leq)$ are *transposed* iff $a = b \wedge a'$ and $b' = b \vee a'$.

A lattice of finite length is *semimodular* if $a \wedge b \longrightarrow\!\!< a$ implies $b \longrightarrow\!\!< a \vee b$. A lattice $(A, \leq)$ is *modular* if $a \leq c$ implies $a \vee (b \wedge c) = (a \vee b) \wedge c$. A lattice $(A, \leq)$ of finite length is modular iff both $(A, \leq)$ and $(A, \geq)$ are semimodular. Furthermore, in a modular lattice $(A, \leq)$, transposed intervals are isomorphic. More precisely, let $[b \wedge c, b]$ and $[c, b \vee c]$ be two transposed intervals and define $f(x) := x \vee c$ for $b \wedge c \leq x \leq b$. Then this mapping f is an isomorphism of the two intervals [Bir73, Theorem I.7.13]. A lattice $(A, \leq)$ is *distributive* if $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$ for any $a, b, c \in A$. Then one also has the dual identity $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$. Furthermore, any distributive lattice is modular.

Let $(A, \leq)$ be a poset. Then the set of ideals $X = \mathbb{H}(A, \leq)$ can be ordered by inclusion. The poset $(X, \subseteq)$ is a lattice, the supremum is given by union and the infimum by intersection. One can easily check that it is even a distributive lattice and that an ideal $I \in \mathbb{H}(A, \leq)$ is join-irreducible in this lattice if it is a principal ideal and no singleton. Note that an ideal $I \in \mathbb{H}(A, \leq)$ is join-irreducible iff it covers a unique element of $(\mathbb{H}(A, \leq), \subseteq)$.

Now let $(L, \leq)$ be a distributive lattice. Then $(A, \leq) := (\mathbb{J}(A, \leq), \leq)$ is a poset and $(\mathbb{H}(A, \leq), \subseteq)$ is a distributive lattice. If L is finite, this latter lattice is isomorphic to $(L, \leq)$ [Bir73, Theorem I.4.3].

1.2 Monoid theoretic definitions

A monoid is a set M equipped with a binary operation $\cdot : M^2 \to M$ that is associative and admits a neutral element 1. The *left divisibility relation* on a monoid $(M, \cdot, 1)$ is defined by $x \leq z$ iff there exists $y \in M$ with $x \cdot y = z$. Since the multiplication $\cdot$ is associative, this relation is transitive. It is in addition reflexive since a monoid contains a neutral element. Hence $(M, \leq)$ is a quasi ordered set. Since $1 \leq M$, the set $\{1\}$ is a basis of $(M, \leq)$. In general, $\leq$ is neither a partial order relation since it need not be antisymmetric (consider the reals with addition) nor a wqo (consider the nonnegative reals with addition).

An alphabet Σ is a nonempty finite set. The set $\Sigma^\star$ of words over Σ gets a monoid structure when equipped with the usual concatenation of words. The neutral element is the empty word, which is denoted by ε. The monoid $(\Sigma^\star, \cdot, \varepsilon)$ is called the *free monoid over* Σ.

Let $(M_i, \cdot_i, 1_i)$ be monoids for $i = 1, 2$ and let $f : M_1 \to M_2$ be a function. This function is a *homomorphism* if $f(x \cdot_1 y) = f(x) \cdot_2 f(y)$ for $x, y \in M_1$ and $f(1_1) = f(1_2)$. A *congruence* on the monoid M_1 is an equivalence relation $\sim$ such that $x_i \sim y_i$ for $i = 1, 2$ and $x_i, y_i \in M_1$ imply $x_1 \cdot_1 x_2 \sim y_1 \cdot_1 y_2$.

A *dependence alphabet* or *trace alphabet* is an alphabet Σ endowed with a binary relation D that is reflexive and symmetric. The relation D is called *dependence relation* and its complement $I = \Sigma^2 \setminus D$ is the *independence relation*. From a dependence alphabet (Σ, D), one defines the trace monoid $\mathbb{M}(\Sigma, D)$ as follows: First, let $\sim$ denote the least congruence on the free monoid $(\Sigma^\star, \cdot, \varepsilon)$ with $ab \sim ba$ for $(a, b) \in I$. Note that two equivalent words $v \sim w$ over Σ have the same length. Then $\mathbb{M}(\Sigma, D) = \Sigma^\star/\!\sim$ is a monoid whose elements are called *traces*. Thus, traces are equivalence classes of words. The length $|x|$ of a trace is the length of any of its representatives. Originally, these monoids where considered by Cartier & Foata [CF69] under the name free partially commutative monoids. The name *trace monoid* was coined by Mazurkiewicz [Maz77].

Besides this algebraic definition of trace monoids, there is another, equivalent, construction of them: Again, one starts with a dependence alphabet (Σ, D). A *dependence graph* is either empty or a triple $(V, \preceq, \lambda)$ where $(V, \preceq)$ is a finite poset and $\lambda : V \to \Sigma$ is a mapping such that for $x, y \in V$, one has

- $x \parallel y$ implies $(\lambda(x), \lambda(y)) \notin D$ and
- $x \multimap\!\!< y$ implies $(\lambda(x), \lambda(y)) \in D$.

As usual in mathematics, isomorphic dependence graphs are not differentiated. On the set of dependence graphs one defines a binary operation $\cdot$ by

$$(V_1, \preceq_1, \lambda_1) \cdot (V_2, \preceq_2, \lambda_2) = (V_1 \dot{\cup} V_2, \preceq_1 \cup \preceq_2 \cup (\preceq_1 \circ E \circ \preceq_2), \lambda_1 \cup \lambda_2)$$

where $E = \{(x, y) \in V_1 \times V_2 \mid (\lambda_1(x), \lambda_2(y)) \in D\}$. Then one can easily check that this operation is associative and that the empty dependence graph is a neutral element.

For $a \in \Sigma$, let $t_a = (\{a\}, \{(a, a)\}, \{(a, a)\})$ denote the dependence graph with one vertex that is labeled by the letter a. Since the monoid $\mathbb{M}(\Sigma, D)$ is generated by the elements $[a]$ for $a \in \Sigma$, the mapping $[a] \mapsto t_a$ can uniquely be extended to a homomorphism from the trace monoid $\mathbb{M}(\Sigma, D)$ to the monoid of dependence graphs. It turns out that this homomorphism is an isomorphism of the monoids. Hence traces can be considered as labeled partially ordered sets. The relation between traces, i.e. equivalence classes of words, and labeled posets can be seen in another light, too:

Recall that $x \leq z$ iff there exists $y \in \mathbb{M}(\Sigma, D)$ such that $x \cdot y = z$. Since $x < z$ implies $|x| < |y|$, on the trace monoid $\mathbb{M}(\Sigma, D)$, the left divisibility relation is a partial order. One can show that $({\downarrow}x, \leq)$ is a distributive lattice for any trace x. Let $(V, \leq, \lambda)$ be the dependence graph associated to x. Then the partial order of join-irreducibles of $({\downarrow}x, \leq)$ is isomorphic to $(V, \leq)$. Vice versa, $({\downarrow}x, \leq)$ is isomorphic to the set of ideals of $(V, \leq)$, i.e. to $(\mathbb{H}(V, \leq), \subseteq)$.

1.3 Logic

First, we introduce the monadic second order logic MSO that allows to reason on Σ-labeled graphs: So let Σ be an alphabet, i.e. a finite set. Let $V_e = \{x_i \mid i \in \mathbb{N}\}$ be a countable set of *elementary variables* and $V_s = \{X_i \mid i \in \mathbb{N}\}$ a countable set of *set variables*. There are three kinds of *atomic formulae*, namely $E(x_i, x_j)$, $X_j(x_i)$ and $\lambda(x_i) = a$ for $i, j \in \mathbb{N}$ and $a \in \Sigma$. *Formulas* are built up from these atomic formulae by the usual connectors $\wedge$ and $\neg$ and by existential quantification over elementary and over set variables. More precisely, if φ and ψ are formulae, then so are $\neg\varphi$, $\varphi \wedge \psi$, $\exists x_i \varphi$ and $\exists X_i \varphi$ where $i \in \mathbb{N}$. To define when a Σ-labeled graph (V, E, λ) satisfies a formula, let $f_e : V_e \to V$ and $f_s : V_s \to 2^V$ be mappings. Then

$$\begin{array}{lll}
(V,E,\lambda) \models_{f_e,f_s} E(x_i,x_j) & \text{iff} & (f_e(x_i), f_e(x_j)) \in E, \\
(V,E,\lambda) \models_{f_e,f_s} X_j(x_i) & \text{iff} & f_e(x_i) \in f_s(X_j), \\
(V,E,\lambda) \models_{f_e,f_s} \lambda(x_i) = a & \text{iff} & \lambda \circ f_e(x_i) = a, \\
(V,E,\lambda) \models_{f_e,f_s} \neg\varphi & \text{iff} & \text{not } (V,E,\lambda) \models_{f_e,f_s} \varphi, \text{ and} \\
(V,E,\lambda) \models_{f_e,f_s} \varphi \wedge \psi & \text{iff} & (V,E,\lambda) \models_{f_e,f_s} \varphi \text{ and } (V,E,\lambda) \models_{f_e,f_s} \psi.
\end{array}$$

Furthermore, $(V, E, \lambda) \models_{f_e,f_s} \exists x_i \varphi$ if there exists a function $g_e : V_e \to V$ such that $(V, E, \lambda) \models_{g_e,f_s} \varphi$ and this function differs from f_e at most in the value of x_i. Similarly, $(V, E, \lambda) \models_{f_e,f_s} \exists X_j \varphi$ if there exists a function $g_s : V_s \to 2^V$ such that $(V, E, \lambda) \models_{f_e,g_s} \varphi$ and this function differs from f_s at most in the value of X_j.

Let (V, E, λ) be a Σ-labeled graph and let φ be a formula whose free variables are among $\{x_0, x_1, \dots, x_k, X_0, X_1, \dots, X_\ell\}$. Let furthermore $f_e, g_e : V_e \to V$ and $f_s, g_s : V_s \to 2^V$ be mappings such that $f_e(x_i) = g_e(x_i)$ for $0 \le i \le k$ and $f_s(X_i) = g_s(X_i)$ for $0 \le i \le \ell$. Then it is an easy exercise to show that $(V, E, \lambda) \models_{f_e,f_s} \varphi$ iff $(V, E, \lambda) \models_{g_e,g_s} \varphi$. For this reason, one usually writes

$$(V,E,\lambda) \models \varphi[f_e(x_0), f_e(x_1), \dots, f_e(x_k), f_s(X_0), f_s(X_1), \dots, f_s(X_\ell)]$$

for $(V, E, \lambda) \models_{f_e,f_s} \varphi$.

A formula without free variables is called *sentence*. Since the satisfaction of a sentence by a graph does not depend on the functions f_e and f_s, we will in this case simply say that the sentence holds in the graph. A formula is an *elementary formula* if it does not contain any set variable. To stress that some formula is not elementary, we will speak of *monadic formulas*, too.

Let (V, E, λ) be some Σ-labeled graph. The *elementary theory* $\mathrm{Th}(V, E, \lambda)$ of this graph is the set of all elementary sentences that hold in (V, E, λ). Similarly, the *monadic theory* $\mathrm{MTh}(V, E, \lambda)$ is the set of all monadic sentences valid in the graph. We also define the elementary and monadic theory of classes of Σ-labeled graphs $\mathbb{C}$ by

$$\begin{aligned}
\mathrm{Th}(\mathbb{C}) &= \bigcap_{(V,E,\lambda)\in\mathbb{C}} \mathrm{Th}(V,E,\lambda), \text{ and} \\
\mathrm{MTh}(\mathbb{C}) &= \bigcap_{(V,E,\lambda)\in\mathbb{C}} \mathrm{MTh}(V,E,\lambda),
\end{aligned}$$

i.e. the elementary (monadic, respectively) theory of a class of graphs is the set of all elementary (monadic, respectively) sentences that hold in all graphs of this class.

Let $\mathbb{C}_1 \subseteq \mathbb{C}_2$ be two classes of Σ-labeled graphs. The class $\mathbb{C}_1$ is *monadically axiomatizable relative to* $\mathbb{C}_2$ iff there exists a monadic sentence φ such that for any $(V, E, \lambda) \in \mathbb{C}_2$ we have $(V, E, \lambda) \in \mathbb{C}_1$ iff $(V, E, \lambda) \models \varphi$. If φ is even an elementary sentence, the class $\mathbb{C}_1$ is *elementary axiomatizable*. Thus, the notion "axiomatizable" always refers to classes of graphs. Differently, the notion "definable" refers to relations inside some graph: Let $G = (V, E, \lambda)$ be a Σ-labeled graph, $n \in \mathbb{N}$ and φ be a monadic sentence whose free variables are among $\{x_0, x_1, \ldots, x_{n-1}\}$. Then

$$\varphi^G := \{(v_0, v_1, \ldots, v_{n-1}) \in V^n \mid G \models \varphi[v_0, v_1, \ldots, v_{n-1}]\}$$

is the *n-ary relation defined by φ*. An n-ary relation $R \subseteq V^n$ is *monadically definable inside G* if $R = \varphi^G$ for some monadic formula φ. *Elementary definable relations* are defined similarly.

Later, we will also use logical formulae to reason on unlabeled graphs. It should be clear that this just requires that atomic formulas of the form $\lambda(x) = a$ do not occur in the formula in question. The notions satisfaction, sentence, elementary and monadic theory etc. then are the obvious restrictions of the notions we defined above. In the last chapters of both parts, we will concentrate on (labeled) partially ordered sets which are special (labeled) graphs. To make the formulas more intuitive, we will occasionally use subformulas of the form $x \leq y$ as a substitute for $E(x, y)$ and $x \in X$ for $X(x)$. Recall that in the definition of the satisfaction of a monadic formula, monadic variables range over arbitrary sets. Therefore, we considered functions $f_s : V_s \to 2^V$. If $(V, \leq)$ is a partially ordered set, one can restrict the monadic variables to range over chains or antichains, only. This is done by considerting functions $f_s : V_s \to 2^V$ where $f_s(X)$ is an (anti-)chain for any $X \in V_s$. The resulting satisfaction relations are denoted by $\models_A$ if set variables range over antichains, and by $\models_C$ if the set variables range over chains, only. The *monadic chain theory* and the *monadic antichain theory* are defined canonically by

$$\begin{aligned}
\mathrm{MATh}(V, \leq) &= \{\varphi \text{ monadic sentence without } \lambda(x) = a \mid (V, \leq) \models_A \varphi\} \\
\mathrm{MCTh}(V, \leq) &= \{\varphi \text{ monadic sentence without } \lambda(x) = a \mid (V, \leq) \models_C \varphi\} \\
\mathrm{MATh}(\mathfrak{P}) &= \bigcap_{(V,E,\lambda) \in \mathfrak{P}} \mathrm{MATh}(V, E, \lambda), \text{ and} \\
\mathrm{MCTh}(\mathfrak{P}) &= \bigcap_{(V,E,\lambda) \in \mathfrak{P}} \mathrm{MCTh}(V, E, \lambda),
\end{aligned}$$

where $\mathfrak{P}$ is any set of posets.

As usual, we will use abbreviations like

$$\begin{aligned}
\varphi \vee \psi &\quad \text{for} \quad \neg\varphi \wedge \neg\psi, \\
\varphi \to \psi &\quad \text{for} \quad \neg\varphi \vee \psi, \text{ and} \\
\forall x \varphi &\quad \text{for} \quad \neg\exists x \neg\varphi.
\end{aligned}$$

Finally, for some properties that can obviously be expressed by a monadic formula, we will simply use their mathematical or English description as for instance "$\bigcup_{t\in T} X_t$ is everything" for "$\forall x \bigwedge_{t\in T} X_t(x)$" where T is a finite set or "X is a chain" for "$\forall x \forall y((X(x) \wedge X(y)) \to (E(x,y) \vee E(y,x)))$" .

1.4 Some notations

This very last part of the first chapter is devoted to some technical notions that will be used throughout this work. Most of them are standard in one or the other community but might be not so usual in another.

Let A, B be sets and $f : A \to B$. By $\operatorname{im} f$, we denote the image of f, i.e. the set $\{f(a) \mid a \in A\} \subseteq B$. The identity function $A \to A$ is denoted by id_A while the identity relation on A is $\Delta_A = \{(a,a) \mid a \in A\}$. For $A' \subseteq A$, a function $f : A' \to B$ is a partial function from A to B. The set A' is the domain $\mathrm{dom}(f)$ of the partial function f. By $\mathrm{part}(A,B)$, we denote the set of partial functions from A to B with nonempty domain. Already in the preceding section, I used the symbol 2^A for the powerset of A. By $\pi_1 : A \times B \to A$, we denote the projection to the first component of the direct product $A \times B$. Similarly, $\pi_2 : A \times B \to B$ is the projection to the second component. Finally, we write $[n] = \{1, 2, \ldots, n\}$ for the set of positive integers up to n while $\underline{n}$ denotes the set $\{0, 1, 2, \ldots, n-1\}$.

Part I

Asynchronous cellular machines

Chapter 2

Σ-dags and asynchronous cellular machines

In this chapter, we define Σ-dags and asynchronous cellular automata, the central topics of the first part of the present work. In addition, this chapter contains several examples that hopefully enable an intuition connected to the notions defined.

We start with the definition of a Σ-dag. These directed acyclic graphs generalise an aspect of dependence graphs known from trace theory: As defined in Section 1.2, a dependence graph is a labeled partially ordered set $(V, \leq, \lambda)$. Let $-\!\!<$ denote the associated covering relation. Then $(V, -\!\!<, \lambda)$, called Hasse-diagram of the dependence graph $(V, \leq, \lambda)$, is an example of a Σ-dag. In particular, $\lambda^{-1}(a)$, the set of a-labeled nodes, is linearly ordered by the reflexive and transitive closure of the edge relation $-\!\!<$, and for any node y and any label a, there is at most one a-labeled node x with $x -\!\!< y$ and at most one a-labeled node z with $y -\!\!< z$. These properties define a Σ-dag:

Definition 2.1.1 Let Σ be an alphabet. A Σ-*dag* is a triple (V, E, λ) where (V, E) is a finite directed acyclic graph and $\lambda : V \to \Sigma$ is a labeling function such that

1. $\lambda^{-1}(a)$ is linearly ordered with respect to $E^\star$ for any $a \in \Sigma$, and
2. for any $(x, y), (x', y') \in E$ with $\lambda(x) = \lambda(x')$, $\lambda(y) = \lambda(y')$, we have $(x, x') \in E^\star$ if and only if $(y, y') \in E^\star$.

By $\mathbb{D}$, we denote the set of all Σ-dags.

As usual, we will identify isomorphic Σ-dags. So let $(V, E, \lambda) \in \mathbb{D}$. Since (V, E) is acyclic, $E^\star$ is a partial order on V. By the first requirement, vertices that carry the same label are comparable with respect to the partial order $E^\star$. In computer science, this property is referred to as "no autoconcurrency". In particular, the width of the partially ordered set $(V, E^\star)$ is bounded by $|\Sigma|$ and

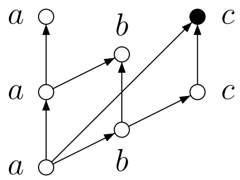


Figure 2.1: An $(\{a\},\{b\},\{c\})$-dag

there is a natural covering of a Σ-dag by the $|\Sigma|$ chains $\lambda^{-1}(a)$ for $a \in \Sigma$. Any edge connects two such chains. The second clause ensures in particular that two edges connecting the same chains (in the same direction) cannot "cross". More precisely, let $(x,y),(x',y') \in E$ with $\lambda(x) = \lambda(x')$ and $\lambda(y) = \lambda(y')$, i.e. these two edges connect the same chains in the same direction. Then, by the first requirement, x and x' are comparable with respect to $E^\star$, say $(x,x') \in E^\star$. Then the second requirement forces $(y,y') \in E^\star$. In particular, if $(x,y),(x,y') \in E$ with $\lambda(y) = \lambda(y')$, then $y = y'$ and dually if $(x,y),(x',y) \in E$ with $\lambda(x) = \lambda(x')$ then x and x' are forced to be equal.

Example 2.1.2 1. Let $\Sigma = \{a,b,c\}$. Then the labeled directed acyclic graph depicted in Figure 2.1 is a Σ-dag.

2. Let $(P,\leq)$ be a finite partial order and $\lambda : P \to \Sigma$ be a mapping. Then the triple $(P,\leq,\lambda)$ is a *pomset without autoconcurrency* if, for any $a \in \Sigma$, the set $\lambda^{-1}(a)$ is linearly ordered by $\leq$ (The left dag in Figure 2.2 is a pomset without autoconcurrency). Note that $(V,E^\star,\lambda)$ is a pomset without autoconcurrency for any Σ-dag (V,E,λ). Conversely, a pomset without autoconcurrency is not a Σ-dag for it may violate the second requirement. Now let $x \longrightarrow\!\!< y$ denote that $x < y$ and there is no element properly between x and y (We say that *x is covered by y*). The *Hasse-diagram* $\mathrm{Ha}(P,\leq,\lambda)$ of $(P,\leq,\lambda)$ is the labeled directed acyclic graph $(P,\longrightarrow\!\!<,\lambda)$. It is easily checked that this Hasse-diagram is a Σ-dag whenever $(P,\leq,\lambda)$ is a pomset without autoconcurrency (cf. the right dag in Figure 2.2).

Let $t = (V,E,\lambda) \in \mathbb{D}$ be a Σ-dag and $x \in V$. Then the *reading domain* $\mathrm{R}(x)$ of x is the set of all letters a from Σ that satisfy

$$\exists y \in V : \lambda(y) = a \text{ and } (y,x) \in E,$$

i.e. $R(x)$ is the set of labels of those nodes $y \in V$ that are connected with x by an edge (y,x). Informally, these nodes can be seen as the lower neighbors of

Figure 2.2: A pomset without autoconcurrency and its Hasse-diagram

x in the dag (V, E, λ) (but not necessarily in the partial order $(V, E^{\star}, \lambda)$). For $a \in \mathrm{R}(x)$ let $\partial_a(x)$ denote the (unique) element y of $\lambda^{-1}(a)$ with $(y, x) \in E$. Thus, $\{\partial_a(x) \mid a \in \mathrm{R}(x)\}$ is the set of lower neighbors of x in the Σ-dag t.

Example 2.1.2 (continued) Let x denote the element of the Σ-dag (P, E, λ) from Figure 2.1 depicted by a solid circle. Since x is the target of edges whose source is labeled by a and by c, respectively, the reading domain $\mathrm{R}(x)$ is $\{a, c\}$. Differently, the solid circle in the Σ-dag from Figure 2.2 is the target of only one edge whose source is labeled by c. Hence, for this Σ-dag, $\mathrm{R}(x) = \{c\}$.

Next we define asynchronous cellular machines, the model of parallel systems that we are going to investigate. An *asynchronous cellular machine over* Σ or Σ-*ACM* is a tuple

$$\mathcal{A} = ((Q_a, \sqsubseteq_a)_{a\in\Sigma}, (\delta_{a,J})_{a\in\Sigma, J\subseteq\Sigma}, F)$$

where

1. $(Q_a, \sqsubseteq_a)$ is an at most countable, well-quasi ordered set of local states for any $a \in \Sigma$,

2. $\delta_{a,J} : \prod_{b\in J} Q_b \to 2^{Q_a}$ is a nondeterministic transition function for any $a \in \Sigma$, $J \subseteq \Sigma$, and

3. $F \subseteq \bigcup_{J\subseteq\Sigma} \prod_{b\in J} Q_b$ is a finite set of accepting states.

One can think of a Σ-ACM as a Σ-tuple of sequential devices. The device with index a performs the a-labeled events of an execution. Doing so, it reads states from other consitutents of the Σ-ACM. But it changes its own state, only (see below for a formal definition of a run).

A Σ-ACM is *deterministic* if, for any $a \in \Sigma$, $J \subseteq \Sigma$ and $q_b \in Q_b$ for $b \in J$ the set $\delta_{a,J}((q_b)_{b\in J})$ is a singleton.[1] The set of all local states $\bigcup_{a\in\Sigma} Q_a$ will be denoted by Q.

Example 2.1.3 Let Σ be an alphabet. For $a \in \Sigma$ let $Q_a := \mathbb{N}^\Sigma$ be the set of all functions $\Sigma \to \{0, 1, 2, \dots\}$. The local wqos $\sqsubseteq_a$ are defined by $f \sqsubseteq_a g$ iff $f(b) \leq g(b)$ for any $b \in \Sigma$. Next, we define the transition function by

$$\delta_{a,J}((f_c)_{c\in J}) := \begin{cases} \emptyset & \text{if there exist } b, c \in J \text{ with } b \neq c \text{ and } f_b(c) \geq f_c(c) \\ \{g\} & \text{otherwise} \end{cases}$$

where the function $g : \Sigma \to \mathbb{N}$ is given by

$$g(b) := \begin{cases} \sup\{f_c(b) \mid c \in J\} & \text{if } a \neq b \\ 1 + \sup\{f_c(b) \mid c \in J\} & \text{if } a = b. \end{cases}$$

Furthermore, F is the set of all tuples $(f_c)_{c\in J}$ for $J \subseteq \Sigma$ with $f_c(b) \in \{0, 1\}$ for all $b \in \Sigma$ and $f_c(c) = 0$ for $c \in J$. The Σ-ACM $\mathcal{A} = ((Q_a, \sqsubseteq_a)_{a\in\Sigma}, (\delta_{a,J})_{a\in\Sigma,J\subseteq\Sigma}, F)$ is not deterministic since in some cases $\delta_{a,J}((f_c)_{c\in J})$ is the empty set. Defining $Q'_a := Q_a \dot\cup \{\bot\}$ with $\bot \sqsubset_a f$ for $f \in Q_a$ and $\delta'_{a,J}((f_c)_{c\in J}) = \{\bot\}$ in all cases where it was undefined so far, we obtain a deterministic Σ-ACM $\mathcal{A}'$. We will return to this example later and show that $\mathcal{A}$ accepts the set of all Hasse-diagrams of pomsets without autoconcurrency.

A Σ-ACM is called *asynchronous cellular automaton* over Σ (Σ-*ACA* for short) provided the sets of local states Q_c are finite for $c \in \Sigma$. Usually, for an ACA we will assume the wqos $\sqsubseteq_c$ to be the trivial reflexive relation Δ_{Q_c} on Q_c.

Next we define how a Σ-ACM can run on a Σ-dag and when it accepts a Σ-dag. Let $t = (V, E, \lambda)$ be a Σ-dag and $\mathcal{A}$ a Σ-ACM. Let $r : V \to \bigcup_{a\in\Sigma} Q_a$ be a mapping and $x \in V$ be a node of t. Then r *satisfies the run condition of* $\mathcal{A}$ *at* x *(relative to* t*)* if

$$r(x) \in \delta_{\lambda(x),\mathrm{R}(x)}((r\partial_b(x))_{b\in\mathrm{R}(x)}).$$

The mapping r is a *run of* $\mathcal{A}$ *on* t if it satisfies the run condition for any $x \in V$. Note that, for a run r and $x \in V$, we always have $r(x) \in Q_{\lambda(x)}$ since the image of $\delta_{\lambda(x),\mathrm{R}(x)}$ belongs to $Q_{\lambda(x)}$.

Although the transitions of a Σ-ACM $\mathcal{A}$ are modeled by functions $\delta_{a,J}$, we can think of them as tuples $(q, (p_b)_{b\in J})$ with $q \in \delta_{a,J}((p_b)_{b\in J})$. Such a tuple can be understood as a directed acyclic graph with node set $\{q, p_b \mid b \in J\}$ and edges from p_b to q for $b \in J$. Furthermore, the nodes are labeled by elements of $\Sigma \times Q$ where q gets the label (a, q) and p_b is labeled by (b, p_b). Note that

[1] Note that a deterministic Σ-ACM is "complete" since $\delta_{a,J}((q_b)_{b\in J}) \neq \emptyset$. As usual, this is no proper restriction since "incomplete" Σ-ACMs can be "completed" by the introduction of an additional state.

on the other side a run r on a Σ-dag $t = (V, E, \lambda)$ defines a (Σ, Q)-labeled dag by $t' = (V, E, \lambda \times r)$. Then r is a run iff for any $y \in V$, the restriction of t' to y and its lower neighbors is a transition, i.e. if t' can be "tiled" by the transitions. Differently from nondeterministic graph acceptors, considered e.g. in [Tho97b, Tho97a], here we take only into account the lower neighbors and not all neighbors of a node y. The reason for this restriction is that we understand a Σ-dag as a process that continues in time. Having this in mind, it is clear that the state reached by performing a node y can depend only on its history but not on the future.

Now let r be a run on the Σ-dag $t = (V, E, \lambda)$. It is *successful* provided there exists a tuple $(q_a)_{a \in \lambda(V)} \in F$ with

$$r(\max(\lambda^{-1}(a))) \sqsupseteq_a q_a \text{ for all } a \in \lambda(V),$$

i.e. if the "global final state" $(r(\max\{\lambda^{-1}(a)\}))_{a \in \lambda(V)}$ dominates some accepting state in the direct product of the wqos $\sqsubseteq_a$. Let $L(\mathcal{A}) \subseteq \mathbb{D}$ denote the set of all Σ-dags that admit a successful run of $\mathcal{A}$. Let M be a set of Σ-dags and $L \subseteq M$. Then we say that L *can be accepted by a Σ-ACM relative to M* if there is a Σ-ACM $\mathcal{A}$ with $L(\mathcal{A}) \cap M = L$. Sometimes we will omit the term "relative to M" if the set M is clear from the context or if M is the set of all Σ-dags. The word "recognize" is reserved for asynchronous cellular automata, i.e. a set of Σ-dags L is *recognizable relative to M* if there exists a Σ-ACA $\mathcal{A}$ with $L = L(\mathcal{A}) \cap M$.

Example 2.1.3 (continued) Let Σ be an alphabet and let Ha denote the set of Hasse-diagrams of pomsets without autoconcurrency. Then $\text{Ha} \subseteq \mathbb{D}$. Furthermore, let $\mathcal{A}$ denote the Σ-ACM defined above. We show that $L(\mathcal{A}) = \text{Ha}$: For a Hasse-diagram $(P, -\!\!<, \lambda) \in \text{Ha}$ let $r(x)(a)$ be the number of a-labeled elements below x (possibly including x, cf. Figure 2.3 for an example where a tuple (x, y, z) denotes the function $\{(a, x), (b, y), (c, z)\}$). For $x \in P$, the reading domain $\mathrm{R}(x)$ is the set of labels of vertices covered by x. Thus, the vertices $\partial_c(x)$ for $c \in \mathrm{R}(x)$ are mutually incomparable. Hence, for $c \in \mathrm{R}(x)$, the vertex $\partial_c(x)$ dominates the largest number of c-labeled vertices among $\{\partial_d(x) \mid d \in \mathrm{R}(x)\}$. Hence $r(\partial_c(x))(c) > r(\partial_d(x))(c)$ for $d \in \mathrm{R}(x) \setminus \{c\}$, i.e. r is a run of $\mathcal{A}$ on $(P, -\!\!<, \lambda)$. Since any tuple $(g_c)_{c \in J}$ dominates some state from F, it is accepting, i.e. $\text{Ha} \subseteq L(\mathcal{A})$. Conversely, let r be a successful run of $\mathcal{A}$ on the Σ-dag (V, E, λ). Then, for any $x \in V$, $c \in \mathrm{R}(x)$ and $a \in \Sigma$ we have $r(\partial_c(x))(a) \leq r(x)(a)$, i.e. $h_a : (V, E^\star) \to \mathbb{N}$ defined by $h_a(x) := r(x)(a)$ is monotone with respect to the partial order $(V, E^\star)$. Furthermore, by the definition of $\delta_{a,J}$, for any $x \in V$ and $c, d \in \mathrm{R}(x)$ with $c \neq d$ we have $h_c(\partial_c(x)) > h_c(\partial_b(x))$. Hence $\partial_c(x) \not\leq \partial_b(x)$. Since we can similarly infer $\partial_b(x) \not\leq \partial_c(x)$, the elements $\partial_c(x)$ and $\partial_b(x)$ are incomparable. Hence (V, E, λ) is a Hasse-diagram. Thus, the set of Hasse-diagrams can be accepted by a Σ-ACM with infinitely many states. It is not known whether finitely many states suffice. On the contrary, Lemma 4.1.3 below will show that

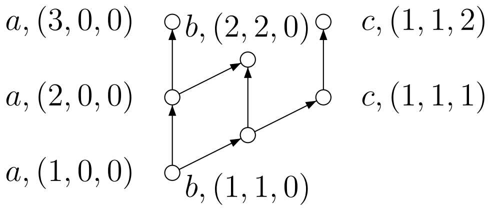


Figure 2.3: A run of $\mathcal{A}$

the set of *not*-Hasse-diagrams can be accepted by a Σ-ACA, i.e. by a Σ-ACM with only finitely many states.

Example 2.1.4 Let L be the set of all Σ-dags t satisfying

the number of d-labeled vertices of t is even for any $d \in \Sigma$.

This set can be accepted by a Σ-ACM that differs from the ACM $\mathcal{A}$ from Example 2.1.3 only in the wqos $\sqsubseteq_a$: Here, we define $f \sqsubseteq_a g$ iff $f(b) \leq g(b)$ for $b \in \Sigma$ and $f(a) \equiv g(a) \mod 2$. Then a tuple $(f_c)_{c\in\Sigma}$ dominates some accepting state iff $f_c(c)$ is even for all $c \in J$. Consider the run of $\mathcal{A}$ on the Σ-dag in Figure 2.3: The maximal a-labeled vertex carries a state f_a with $3 = f_a(a)$. Furthermore, let f_b and f_c denote the state at the maximal b-labeled and c-labeled vertex, respectively. Then the tuple (f_a, f_b, f_c) does not dominate (in the wqo $\sqsubseteq_a \times \sqsubseteq_b \times \sqsubseteq_c$) any state from F, i.e. the run r is not successful.

Recall that a Σ-ACM is an asynchronous cellular automaton if the sets of local states Q_a are finite. In this case, we usually consider the identity relation Δ_{Q_a} as the wqo $\sqsubseteq_a$. Then, as is easy to see, any asynchronous cellular automaton is monotone: A Σ-ACM is *monotone* if, for any $a \in \Sigma$, $J \subseteq \Sigma$, $p_b, p'_b \in Q_b$ for $b \in J$ and $q \in Q_a$, we have

$$q \in \delta_{a,J}((p_b)_{b\in J}) \text{ and } p_b \sqsubseteq_b p'_b \text{ for } b \in J \Longrightarrow \exists q' \in \delta_{a,J}((p'_b)_{b\in J}) : q \sqsubseteq_a q'.$$

Intuitively, this means that increasing the input of a transition does not disable the transition and increases its output.

We finish this chapter with some examples of the expressive power of monotone ACMs. In the first of these examples, we consider ACMs that run on words over Σ. To do this, we identify a word over Σ with the Hasse-diagram of a linearly ordered Σ-labeled poset. In this sense, we can show that the “word-language”

$\{a^m b^n \mid 1 \leq n < m\}$ can be accepted by a monotone ACM. This in particular implies that monotone ACMs are more powerful than finite automata since ACMs can have infinite sets of states. In addition, we will show that the set $\{b^n a^m \mid 1 \leq n < m\}$ cannot be accepted by a Σ-ACM. Thus, the set of languages acceptable by a monotone ACM is not closed under reversal. This might be surprising at first glance, but it is not really so since the notion of well quasi ordering as well as that of monotonicity are not symmetric.

Example 2.1.5 Let $\Sigma = \{a, b\}$ and $L = \{a^m b^n \mid 1 \leq n < m\} \subseteq \Sigma^+$. We consider the words in Σ^+ as Hasse-diagrams of linearly ordered sets so that $L \subseteq \mathbb{D}$. Let $\mathcal{A}$ denote the following Σ-ACM:
$Q_a = Q_b = \mathbb{N}$ with the usual linear order which is a wqo,

$$\delta_{a,J}((q_c)_{c\in J}) = \begin{cases} \emptyset & \text{if } b \in J \\ \{1\} & \text{if } J = \emptyset \\ \{q_a + 1\} & \text{otherwise, i.e. if } J = \{a\}, \text{ and} \end{cases}$$

$$\delta_{b,J}((q_c)_{c\in J}) = \begin{cases} \emptyset & \text{if } J = \{a, b\} \text{ or } J = \emptyset \\ \{\max(0, q_a - 1)\} & \text{if } J = \{a\} \\ \{\max(0, q_b - 1)\} & \text{otherwise, i.e. if } J = \{b\}. \end{cases}$$

The state $(1, 1)$ is the only accepting state from F. We show that the only Hasse-diagrams of linearly ordered sets that are accepted by $\mathcal{A}$ are those from the set L: So let $(V, E, \lambda) \in L$. It is of the form

$$a_1 \text{ --- } a_2 \text{ --- } \ldots \text{ --- } a_m \text{ --- } b_1 \text{ --- } b_2 \text{ --- } \ldots \text{ --- } b_n$$

with $1 \leq n < m$, $\lambda(a_i) = a$ and $\lambda(b_i) = b$ for all suitable i. Then the mapping $r : V \to \mathbb{N}$ with $r(a_i) = i$ and $r(b_i) = m - i$ is a run of $\mathcal{A}$ on (V, E, λ). It is successful since the final global state $(q_c)_{c\in\Sigma}$ equals $(m, m - n)$ and $m - n \geq 1$. If, on the contrary, (V, E, λ) is the Hasse-diagram of a linear order, but not from L, then
either it contains some a-labeled vertex that covers a b-labeled one,
or it contains some b-labeled vertex which is not the target of any edge,
or it is of the form $a^m b^n$ with $m \leq n$.

In the first case, the ACA $\mathcal{A}$ does not have any run on (V, E, λ) due to $\delta_{a,J}((q_c)_{c\in J}) = \emptyset$ whenever $b \in J$. Similarly in the second case, since $\delta_{b,\emptyset} = \emptyset$. In the third case, there is a run of $\mathcal{A}$ on (V, E, λ), but the final global state is of the form $(m, 0)$ and therefore not successful.

Now let $L' = \{b^n a^m \mid 1 \leq n < m\}$ denote the set of reversed words from L. We show that there is no monotone Σ-ACM $\mathcal{A}'$ that accepts the Hasse-diagrams that correspond to words in L' relative to the Hasse-diagrams of linear orders: By contradiction, assume there is such a Σ-ACA $\mathcal{A}'$. For $n > 0$ let t_n denote the Hasse-diagram corresponding to $b^n a^{n+1}$. Since these words belong to L', there exists a successful run r_n of $\mathcal{A}'$ on t_n for any $n > 0$. Let q_n denote the state that is associated by r_n to the last b-labeled vertex in t_n (i.e. $q_n \in Q_b$ is the state of $\mathcal{A}'$ that is reached after performing the b-prefix of $b^n a^{n+1}$). Since $\sqsubseteq_b$ is a wqo, there are $n < m$ with $q_n \sqsubseteq_b q_m$. Now consider the Hasse-diagram t associated to the word $b^m a^{n+1}$. Since $n < m$, this word does not belong to L'. Nevertheless, since $\mathcal{A}'$ is monotone, there is a successful run on $t = (V, E, \lambda)$: This Σ-dag has the form:

$$b_1 \text{——} b_2 \text{——} \ldots \text{——} b_m \text{——} a_1 \text{——} a_2 \text{——} \ldots \text{——} a_n$$

The mapping $r_m \restriction \{b_1, b_2, \ldots, b_m\}$ satisfies the run conditions for b_i relative to t since r_m is a run of $\mathcal{A}'$ on t_m. Furthermore, $r_m(b_m) = q_m \sqsupseteq_b q_n$. Since $r_n(a_1) \in \delta_{a,\{b\}}(q_n)$ and since $\mathcal{A}'$ is monotone, there exists $r(a_1) \in \delta_{a,\{b\}}(q_m)$ with $r_n(a_1) \sqsubseteq_a r(a_1)$. By induction, we obtain states $r(a_i) \sqsupseteq_a r_n(a_i)$ such that $r_m \restriction \{b_1, b_2, \ldots, b_m\} \cup r$ is a run of $\mathcal{A}'$ on t. Since $r(a_{n+1}) \sqsupseteq_a r_n(a_{n+1})$ and $r_m(b_m) \sqsupseteq_b r_n(b_n)$, the final global state $(r_m(b_m), r(a_{n+1}))$ of this run dominates that of r_n which equals $(r_n(b_n), r_n(a_{n+1}))$. But r_n was successful, hence so is this new run, i.e. t is accepted by $\mathcal{A}'$.

Note that the language $\{b^n a^m \mid 1 \leq n < m\}$ cannot be accepted by a finite sequential automaton. Hence, it is not monadically axiomatizable. The last example in this chapter gives an elementary axiomatizable set of Σ-dags that cannot be accepted by a monotone ACM:

Example 2.1.6 Let $\Sigma = \{a, b\}$ and let φ denote the first-order sentence

$$\forall x \exists y ((\lambda(x) = a) \to ((\lambda(y) = b) \wedge (x, y) \in E)).$$

Note that a Σ-dag satisfies φ iff every a-labeled element is the source of an edge that leads to a b-labeled vertex. Furthermore, let L be the set of all Σ-dags that satisfy φ. We show that L cannot be accepted by a monotone Σ-ACM:

By contradiction, we assume $\mathcal{A}$ to be a Σ-ACA such that $L(\mathcal{A}) = L$. For $n \in \mathbb{N}$ consider the Σ-dag $t_n = (V_n, E_n, \lambda_n)$ defined as follows: The set V_n equals $\{a_i, b_i \mid i = 1, 2, \ldots, n\}$ with the edge relation

$$\{(a_i, a_{i+1}) \mid 1 \leq i < n\} \cup \{(b_i, b_{i+1}) \mid 1 \leq i < n\} \cup \{(a_i, b_i) \mid 1 \leq i \leq n\}$$

and the labeling $\lambda_n(a_i) = a$ and $\lambda_n(b_i) = b$ for $1 \leq i \leq n$ (cf. the first Σ-dag in Figure 2.4 for the case $n = 8$).

Recall that φ states that every element labeled by a is the source of an edge leading to an element labeled by b. Hence $t_n \in L$. By the assumption that $\mathcal{A}$ accepts those Σ-dags that satisfy φ, there exists a successful run r_n of $\mathcal{A}$ on t_n for all $n \in \mathbb{N}$. Let w_n denote the word $r_n(a_1) r_n(a_2) r_n(a_3) \ldots r_n(a_n) \in Q_a^\star$. By Higman's Theorem [Hig52], there exist $m < n$ such that w_m is dominated by a subword of w_n that contains the last position, i.e. such that there exist

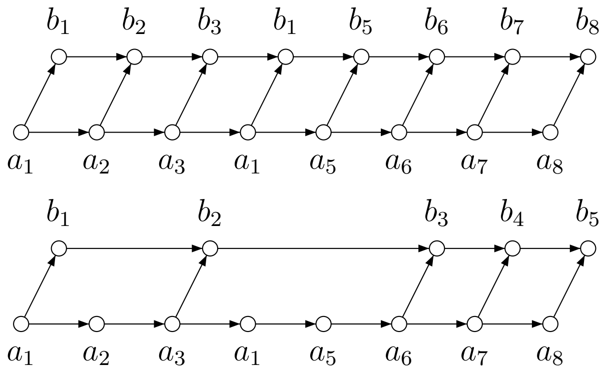


Figure 2.4: cf. Example 2.1.6.

$1 \leq j_1 < j_2 < \cdots < j_m = n$ with $r_m(a_i) \sqsubseteq_a r_n(a_{j_i})$. Now consider the Σ-dag $t = (V, E, \lambda)$ defined by $V = \{a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_m\}$, $\lambda(a_i) = a$ and $\lambda(b_i) = b$ for all suitable i and the edge relation

$$\{(a_i, a_{i+1}) \mid 1 \leq i < n\} \cup \{(b_i, b_{i+1}) \mid 1 \leq i < m\} \cup \{(a_{j_i}, b_i) \mid 1 \leq i \leq m\}$$

(see the second Σ-dag in Figure 2.4 with $m = 5, n = 8, j_1 = 1, j_2 = 3, j_3 = 6, j_4 = 7$, and $j_5 = 8$). Then $t \notin L$. On the other hand we construct a successful run r of $\mathcal{A}$ on t as follows: Let $r(a_i) = r_n(a_i)$ for $1 \leq i \leq n$. Then r satisfies the run condition at a_i since r_n is a run. Recall that $r_m(a_i) \sqsubseteq_a r_n(a_{j_i}) = r(a_{j_i})$. Since $\mathcal{A}$ is monotone, there exists $r(b_1) \in \delta_{b,\{a\}}(r(a_{j_1}))$ with $r_n(b_1) \sqsubseteq_b r(b_1)$. Inductively, we find $r(b_i) \in \delta_{b,\{a,b\}}(r(a_{j_i}), r(b_{i-1}))$ with $r(b_i) \sqsupseteq_b r_m(b_i)$. At the end, $r(b_m) \sqsupseteq_b r_m(b_m)$. Thus, the final global state $(r(a_n), r(b_m))$ dominates $(r_m(a_m), r_m(b_m))$. Since r_m was successful, so is r, i.e. $t \in L(\mathcal{A})$, a contradiction.

On the other hand, the set $\mathbb{D} \setminus L$ can easily be accepted by a Σ-ACA, i.e. with only finitely many states. A Σ-dag does not belong to L iff it has an a-labeled vertex that is not the source of any edge connecting it to a b-labeled vertex, i.e. the state associated to this vertex by a possible run is not seen by any b-labeled vertex. Thus, the idea of a Σ-ACA that accepts the complement of L is to nondeterministically pick at least one a-labeled vertex and mark it by a distinguished state. The b-transitions just have to check that they do not read this distinguished state.

Chapter 3

Decidability results

In the preceding chapter, we defined Σ-ACMs as a model of a computing device that can perform computation tasks concurrently. The behavior of a Σ-ACM is the accepted language, i.e. a set of Σ-dags. Hence a Σ-ACM describes a property of Σ-dags. Since the intersection as well as the union of two languages $L(\mathcal{A}_1)$ and $L(\mathcal{A}_2)$ can be accepted by a Σ-ACM, properties describable by Σ-ACMs can become quite complex. Then it is of interest whether the combined property is contradictory, or, equivalently, whether at least one Σ-dag satisfies it. Thus, one would like to know whether a Σ-ACM accepts at least one Σ-dag. In this chapter, we show that it is possible to gain this knowledge even automatically, i.e. we show that there exists an algorithm that on input of a Σ-ACM decides whether the Σ-ACM accepts at least one Σ-dag. In other words, the aim of this chapter is to show that the question "Does $\mathcal{A}$ accept some Σ-dag?" is decidable. More precisely, it is shown that the set

$$\{\mathcal{A} \mid \mathcal{A} \text{ is monotone and effective and } L(\mathcal{A}) = \emptyset\}$$

is recursive. I am grateful to Peter Habermehl who pointed me to the paper [FS98, FS01] that deals with well-structured transition systems. The proof of the mentioned decidability rests on this result.

3.1 Notational conventions and definitions

Let $\mathbb{N}^+ = \{1, 2, \dots\}$. Nevertheless, in this chapter an expression $\sup(M)$ for $M \subseteq \mathbb{N}^+$ will be understood in the structure $(\mathbb{N}, \leq)$. The useful effect of this convention is that $\sup(M) = 0$ for $M \subseteq \mathbb{N}$ if and only if M is empty.

Let A be a set. Then in this chapter, a *word* is a mapping $w : M \to A$ where M is a finite subset of $\mathbb{N}^+$. If $M = \{n_1, n_2, \dots, n_k\}$ with $n_1 < n_2 < \dots < n_k$, the finite sequence $w(n_1)w(n_2)\dots w(n_k)$ is a word in the usual sense. Two words $v : M \to A$ and $w : N \to A$ are *isomorphic* (and we will identify them) if there is an order isomorphism (with respect to the usual linear order of the natural

$$\begin{array}{ccc} s' & \longrightarrow & t' \\ \curlyvee\!| & & \curlyvee\!| \\ s & \longrightarrow & t \end{array}$$

Figure 3.1: Lifting of a transition in a WSTS

numbers) $\eta : M \to N$ with $v = w \circ \eta$. By $A^\star$ we denote the set of all words over A. Furthermore, for $w \in A^\star$ and $a \in A$ let wa denote the word $v : \operatorname{dom} w \cup \{n\} \to A$ with $n > \operatorname{dom} w$, $v \restriction \operatorname{dom} w = w$ and $v(n) = a$. By ε, we denote the empty word, i.e. the mapping $\varepsilon : \emptyset \to A$.

Recall that we identify isomorphic Σ-dags. Hence, we can impose additional requirements on the carrier set V as long as they can be satisfied in any isomorphism class. It turns out that in the considerations we are going to do in this section, it will be convenient to assume that for any Σ-dag (V, E, λ)

> $V \subseteq \mathbb{N}^+$ and that the partial order $E^\star$ is contained in the usual linear order on $\mathbb{N}^+$.

Note that on the set $H := \lambda^{-1}(a)$ we have two linear orders: $E^\star$ and the order $\leq$ of the natural numbers. Since $\leq$ extends $(V, E^\star)$, these two linear orders on H coincide. Hence, for a run r of some Σ-ACM on $t = (V, E, \lambda)$, the mapping $r \restriction \lambda^{-1}(a) : \lambda^{-1}(a) \to Q_a$ is a word over Q_a whose letters occur in the order given by $(V, E^\star)$.

3.2 Well-structured transition systems

A *transition system* is a set S endowed with a binary relation $\to \subseteq S^2$. For $t \in S$, we denote by $\operatorname{Pred}(t)$ the set of *predecessors* of t in the transition system S, i.e. the set of all $s \in S$ with $s \to t$. A *well-structured transition system* or *WSTS* is a triple $(S, \to, \preceq)$ where $(S, \to)$ is a transition system, $\preceq$ is a wqo on S and for any $s, s', t \in S$ with $s \to t$ and $s \preceq s'$ there exists $t' \in S$ with $s' \to t'$ and $t \preceq t'$. Thus, a WSTS is a well-quasi ordered transition system such that any transition $s \to t$ "lifts" to a larger state $s' \succeq s$ (cf. Figure 3.1). This definition differs slightly from the original one by Finkel & Schnoebelen [FS01] in two aspects: First, they require only $s' \to^\star t'$ and they call WSTS' satisfying our axiom "WSTS with strong compatibility". Secondly, and more seriously, their transition systems are finitely branching. But it is easily checked that the results from [FS01, Section 2 and 3] hold for infinitely branching transition systems, too. Since we use only these results (namely of Theorem 3.6), it is not necessary to restrict well-structured transitions systems in our context to finitely branching ones. In [FS01], several decidability results are shown for WSTSs. In particular, they showed

Theorem 3.2.1 ([FS01, Theorem 3.6]) *Let $(S, \to, \preceq)$ be a WSTS such that $\preceq$ is decidable and a finite basis of* $\text{Pred}(\uparrow s)$ *can be computed effectively for any $s \in S$. Then there is an algorithm that solves the following decision problem:*
input: *two states $s, t \in S$.*
output: *Does there exist a state $s' \in S$ with $s \to^\star s' \succeq t$, i.e. is t dominated by some state reachable from s?*

Since in their proof the algorithm that decides the existence of the state s' is uniformly constructed from the decision algorithm for $\preceq$ and the algorithm that computes a finite predecessor basis, one gets even more:

Theorem 3.2.2 *There exists an algorithm that solves the following decision problem:*
input:
1. *an algorithm that decides $\preceq$,*
2. *an algorithm computing a finite basis for* $\text{Pred}(\uparrow s)$ *for $s \in S$, and*
3. *two states s and t from S*

for some well-structured transition system $(S, \to, \preceq)$.
output: *Does there exists a state $s' \in S$ such that $s \to^\star s' \succeq t$?*

In this section, we will show that there is an algorithm that, given a Σ-ACM, outputs whether this ACM accepts some Σ-dag. To obtain this result we use well-structured transition systems introduced above and in particular Theorem 3.2.2. Of course, the first idea might be to define a transition system as follows: The states are the runs of the Σ-ACM $\mathcal{A}$, i.e. we could define the state set Z to equal $\{(t, r) \mid t \in \mathbb{D} \text{ and } r \text{ is a run of } \mathcal{A} \text{ on } t\}$. The transitions of the transition system should reflect the computation steps of the ACM $\mathcal{A}$, i.e. we could define $(t, r) \rightsquigarrow (t', r')$ iff there exists a maximal vertex x of t' such that $t = t' \setminus \{x\}$ and $r = r' \upharpoonright t$. Then $(Z, \rightsquigarrow)$ is indeed a transition system that mimics the computations of the ACM $\mathcal{A}$. But to make it a well-structured transition system, we need a well-quasi order on Z that is compatible with $\rightsquigarrow$. Since the states of this transition system are labeled *graphs*, one could try the minor relation that is a wqo on *unlabeled* graphs. But (at least to the author) it is not clear whether this can be extended to labeled graphs (it is even unclear what the labeling of a minor should be).

Recall that the transition relation of the WSTS should reflect the atomic computation steps of the Σ-ACM $\mathcal{A}$. But the labeled graph (t, r) contains much information that is not necessary for this purpose. The only information we really need is

1. the state sequence of the a-component of the automaton $\mathcal{A}$, i.e. the Q_a-word $r \upharpoonright \lambda^{-1}(a)$, and

2. which nodes of t can be read by an additional node x, i.e. for each $a, b \in \Sigma$ we need the information which a-labeled node has already been read by some b-labeled node.

For a Σ-ACM $\mathcal{A} = ((Q_a, \sqsubseteq_a)_{a\in\Sigma}, (\delta_{a,J})_{a\in\Sigma,J\subseteq\Sigma}, F)$, this idea is formalized as follows: Let $t = (V, E, \lambda)$ be a Σ-dag and let $r : V \to Q$ be a run of $\mathcal{A}$ on t. For $a \in \Sigma$, let $w_a := r \restriction \lambda^{-1}(a)$. As explained in Section 3.1, $\lambda^{-1}(a)$ is a subset of $\mathbb{N}^+$ where the order relation $E^\star$ coincides with the usual linear order $\leq$ on $\mathbb{N}$. Hence $w_a : \lambda^{-1}(a) \to Q_a$ is a word over Q_a. Now we define mappings $\text{pos}^w_a : \Sigma \to V$ as follows: For $a, b \in \Sigma$, let $\text{pos}^w_a(b)$ denote the last position in the word w_a that is read by some b-labeled vertex. Formally

$$\text{pos}^w_a(b) := \sup\{x \in \lambda^{-1}(a) \mid \exists y \in \lambda^{-1}(b) : (x, y) \in E\}$$

where the supremum is taken in $\mathbb{N}$ such that, if the set is empty, we have $\text{pos}^w_a(b) = 0$. Note that $\text{pos}^w_a(b)$ is in general not the last position in w_a that is *dominated* by some b-labeled vertex in the partial order $(V, E^\star, \lambda)$. The tuple $(w_a, \text{pos}^w_a)_{a\in\Sigma}$ is called the *state associated with the run* r, denoted $\text{state}(r) := (w_a, \text{pos}^w_a)_{a\in\Sigma}$.

Example 2.1.3 (continued) Let $t = (V, E, \lambda)$ be the Σ-dag and let r denote the run of $\mathcal{A}$ depicted in Figure 2.3 (page 16). Then we have the following:

$$\begin{aligned} v_a &= (1,0,0)(2,0,0)(3,0,0) \\ v_b &= (1,1,0)(2,2,0) \\ v_c &= (1,1,1)(1,1,2) \\ \text{pos}^v_a &= \{(a,2),(b,2),(c,0)\} \\ \text{pos}^v_b &= \{(a,0),(b,1),(c,1)\} \\ \text{pos}^v_c &= \{(a,0),(b,0),(c,1)\} \end{aligned}$$

This situation is visualized in Figure 3.2. There, the words v_a, v_b and v_c are drawn vertically. On the left of a node, the associated state of $\mathcal{A}$ can be found. The letter b at the right of the second a-node indicates that this node equals $\text{pos}^v_a(b)$. Finally, $\text{pos}^v_a(c) = 0$ is indicated by the fact that c does not appear at the right of the word v_a.

As explained above, we want the set of states S to contain $\text{state}(r)$. Thus, $S \subseteq \prod_{a\in\Sigma}(Q^\star_a \times \mathbb{N}^\Sigma)$. Now we define the state set S completely:

$$S := \left\{ (w_a, \text{pos}^w_a)_{a\in\Sigma} \in \prod_{a\in\Sigma}(Q^\star_a \times \mathbb{N}^\Sigma) \mid \text{im}(\text{pos}^w_a) \subseteq \text{dom}\, w_a \cup \{0\} \text{ for } a \in \Sigma \right\}.$$

Note that $0 \notin \mathbb{N}^+$ and therefore in general $\text{im}\,\text{pos}^w_a \not\subseteq \text{dom}\, w_a$.

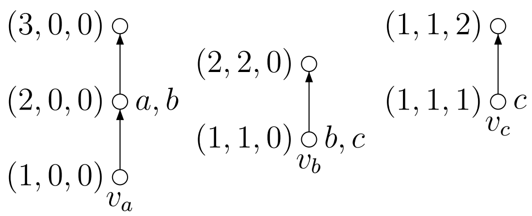


Figure 3.2: The state state(r) of the run from Figure 2.3

The state $(w_a, \text{pos}^w_a)_{a\in\Sigma}$ is a *successor* of the state $(v_a, \text{pos}^v_a)_{a\in\Sigma}$, denoted $(v_a, \text{pos}^v_a)_{a\in\Sigma} \to (w_a, \text{pos}^w_a)_{a\in\Sigma}$, iff there exist $a \in \Sigma$, $\emptyset \neq J \subseteq \Sigma$, $p_b \in Q_b$ for $b \in J$ and $q \in Q_a$ such that

(i) $q \in \delta_{a,J}((p_b)_{b\in J})$,

(ii) $w_c = \begin{cases} v_c q & \text{for } c = a \\ v_c & \text{otherwise,} \end{cases}$

(iii) $\text{pos}^w_c(b) = \text{pos}^v_c(b)$ for all $b, c \in \Sigma$ satisfying either $c \notin J$ or $a \neq b$, and

(iv) $\text{pos}^v_c(a) < \text{pos}^w_c(a) \in \text{dom}\, v_c$ such that $v_c \circ \text{pos}^w_c(a) = p_c$ for $c \in J$.

In this chapter, we will refer to these conditions just as (i),(ii) etc.

The following example indicates that state(r) $\to$ state(r') whenever $(t, r) \rightsquigarrow (t', r')$, i.e. that the transition system $(S, \to)$ really reflects the computations of the ACM $\mathcal{A}$. Even more, we will show that (under some additional assumptions on $\mathcal{A}$) the states of the form state(r) for some run r are precisely those states that are *"reachable"* in the transition system $(S, \to)$ (cf. Lemma 3.3.2). This will enable us to prove the desired decidability result.

Example 2.1.3 (continued) Let t' denote the extension of the Σ-dag t from Figure 2.3 by an a-labeled node as indicated in Figure 3.3 (first picture). Furthermore, this picture shows an extension r' of the run r, too. The second picture depicts the state state(r'). The reader might check that state(r') is a successor state of state(r).

First, we will show that the result of Finkel & Schnoebelen can indeed be applied, i.e. that we can extend the transition system $(S, \to)$ to a well-structured transition system.

So we have to extend the wqo $\sqsubseteq_a$ on Q_a to words over Q_a: To do this, recall that we consider words as mappings from a finite linear order into the well-quasi

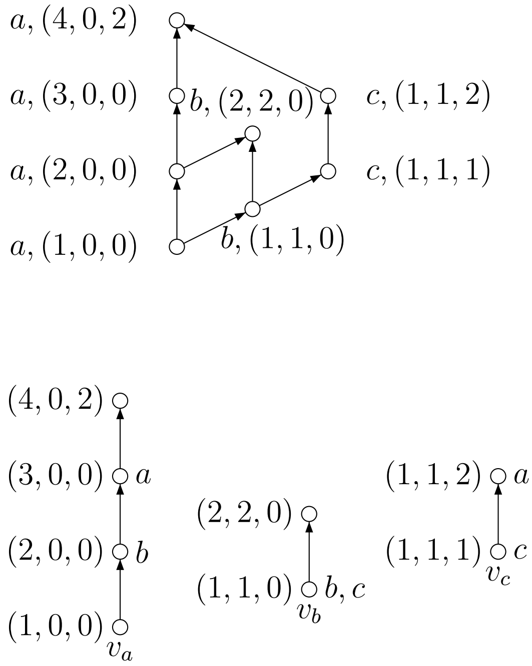


Figure 3.3: A successor state of state(r) from Figure 3.2

ordered set Q_a. Therefore, an *embedding* $\eta : v \hookrightarrow w$ is defined to be an order embedding of $\operatorname{dom} v \cup \{0\}$ into $\operatorname{dom} w \cup \{0\}$ such that

$$\eta(0) = 0,\ \eta(\sup \operatorname{dom} v) = \sup \operatorname{dom} w,\ \text{and } v(i) \sqsubseteq_a w \circ \eta(i) \text{ for } i \in \operatorname{dom} v.$$

Thus, there is an embedding $\eta : v \hookrightarrow w$ iff one obtains v from w by first deleting some letters (but not the last) and then decreasing the remaining ones with respect to $\sqsubseteq_a$. If $\sqsubseteq_a$ is trivial (i.e. the identity relation Δ_{Q_a}), there exists such an embedding iff v is a subword of w and the last letters of v and w coincide. Now a quasi-order $\preceq$ on the states of the transition system $(S, \to)$ is defined by $(v_a, \mathrm{pos}_a^v)_{a\in\Sigma} \preceq (w_a, \mathrm{pos}_a^w)_{a\in\Sigma}$ iff

> there exist embeddings $\eta_a : v_a \hookrightarrow w_a$ such that $\eta_a \circ \mathrm{pos}_a^v = \mathrm{pos}_a^w$ for any $a \in \Sigma$.

As explained above, the existence of the embeddings η_a ensures that v_a is dominated by some subword (including the last letter) of w_a letter by letter. The requirement $\eta_a \circ \mathrm{pos}_a^v = \mathrm{pos}_a^w$ ensures that the pointer $\mathrm{pos}_a^w(b)$ (if not 0) points to some position in this *subword* and that this position corresponds (via η_a) to the position in v_a to which $\mathrm{pos}_a^v(b)$ points. It is obvious that $\preceq$ is reflexive and transitive, i.e. $\preceq$ is a quasiorder. If $\sqsubseteq_a$ is a partial order for any $a \in \Sigma$, the relation $\preceq$ is even a partial order since $(v_a, \mathrm{pos}_a^v)_{a\in\Sigma} \preceq (w_a, \mathrm{pos}_a^w)_{a\in\Sigma}$ implies $|v_a| \le |w_a|$ for any $a \in \Sigma$.

Lemma 3.2.3 *Let $\mathcal{A}$ be a Σ-ACM. Then $(S, \preceq)$ is a well quasi ordering.*

Proof. Let $w \in Q^\star$ and $\mathrm{pos}^w : \Sigma \to \operatorname{dom} w$. We construct a word w' over the set $Q \times 2^\Sigma$ by $\operatorname{dom} w' := \operatorname{dom} w$ and $w(i) := (w(i), (\mathrm{pos}^w)^{-1}(i))$. Now let $v \in Q^\star$ and $\mathrm{pos}^v : \Sigma \to \operatorname{dom} v$ and construct $v' \in (Q \times 2^\Sigma)^\star$ similarly. Then there is an embedding $\eta : v \hookrightarrow w$ with $\eta \circ \mathrm{pos}^v = \mathrm{pos}^w$ iff there exists an embedding $\eta' : v' \hookrightarrow w'$. Thus, $(v_a, \mathrm{pos}_a^v)_{a\in\Sigma} \preceq (w_a, \mathrm{pos}_a^w)_{a\in\Sigma}$ iff v'_a can be embedded into w'_a for any $a \in \Sigma$.

By Higman's Theorem [Hig52], words over a well-quasi ordered set $(Q, \sqsubseteq)$ form a wqo with respect to the embeddability. Since the direct product of finitely many wqos is a wqo, the lemma follows. □

Even though the set of states of the transition system $(S, \to)$ is equiped with a wqo as we saw in the lemma above, the triple $(S, \to, \preceq)$ is in general not a WSTS. For this to hold, we need that the underlying Σ-ACA is monotone: A Σ-ACM is *monotone* if, for any $a \in \Sigma$, $J \subseteq \Sigma$, $p_b, p'_b \in Q_b$ for $b \in J$ and $q \in Q_a$, we have

$$q \in \delta_{a,J}((p_b)_{b\in J}) \text{ and } p_b \sqsubseteq_b p'_b \text{ for } b \in J \Longrightarrow \exists q' \in \delta_{a,J}((p'_b)_{b\in J}) : q \sqsubseteq_a q'.$$

Intuitively, this means that increasing the input of a transition does not disable the transition and increases its output. Furthermore note that any asynchronous cellular automaton is monotone since there the wqo $\sqsubseteq_a$ is just the identity relation Δ_{Q_a} on the finite set Q_a.

Theorem 3.2.4 *Let $\mathcal{A}$ be a monotone Σ-ACM. Then $\mathcal{S}(\mathcal{A}) = (S, \to, \preceq)$ is a well-structured transition system.*

Proof. Let $(v_c, \mathrm{pos}_c^v)_{c\in\Sigma}$, $(w_c, \mathrm{pos}_c^w)_{c\in\Sigma}$ and $(v'_c, \mathrm{pos}_c^{v'})_{c\in\Sigma}$ be states from S such that

$$\begin{array}{ccc} (v'_c, \mathrm{pos}_c^{v'})_{c\in\Sigma} & & \\ \curlyvee\!| & & \\ (v_c, \mathrm{pos}_c^{v})_{c\in\Sigma} & \to & (w_c, \mathrm{pos}_c^{w})_{c\in\Sigma}. \end{array}$$

Let $\eta_c : v_c \hookrightarrow v'{}_c$ denote embeddings that witness $(v_c, \mathrm{pos}_c^v)_{c\in\Sigma} \preceq (v'_c, \mathrm{pos}_c^{v'})_{c\in\Sigma}$. Since $(v_c, \mathrm{pos}_c^v)_{c\in\Sigma} \to (w_c, \mathrm{pos}_c^w)_{c\in\Sigma}$, there exist $a \in \Sigma$, $\emptyset \neq J \subseteq \Sigma$, $p_b \in Q_b$ for $b \in J$, and $q \in Q_a$ satisfying (i)-(iv).

In particular (by (i)) $q \in \delta_{a,J}((p_b)_{b\in J})$. By (iv), we get in addition $p_c = v_c(\mathrm{pos}_c^v(a)) \sqsubseteq_c v'_c \circ \eta_c(\mathrm{pos}_c^{v'}(a)) =: p'_c$. Hence, by the monotonicity of the Σ-ACM $\mathcal{A}$, there exists $q' \in \delta_{a,J}((p'_b)_{b\in J})$ such that $q \sqsubseteq_a q'$.

Let $w'_a := v'_a q'$ and $w'_c := v'_c$ for $c \neq a$. Extend η_a to η'_a by $\eta'_a := \eta_a \cup \{(\sup\mathrm{dom}\, w_a, \sup\mathrm{dom}\, w'_a)\}$ and $\eta'_c := \eta_c$ for $c \neq a$ and define $\mathrm{pos}_c^{w'} := \eta'_c \circ \mathrm{pos}_c^w$ for $c \in \Sigma$. Then $(w'_c, \mathrm{pos}_c^{w'})_{c\in\Sigma} \in S$ and $(w_c, \mathrm{pos}_c^w)_{c\in\Sigma} \preceq (w'_c, \mathrm{pos}_c^{w'})_{c\in\Sigma}$ witnessed by the embeddings η'_c. It remains to show that $(w'_c, \mathrm{pos}_c^{w'})_{c\in\Sigma}$ is a successor of $(v'_c, \mathrm{pos}_c^{v'})_{c\in\Sigma}$, i.e. we have to prove that (i)-(iv) hold for $(v'_c, \mathrm{pos}_c^{v'})_{c\in\Sigma}$, $(w'_c, \mathrm{pos}_c^{w'})_{c\in\Sigma}$, a, J and p'_b for $b \in J$ and q': Property (ii) follows from the definition of w'_c. Now let $b, c \in \Sigma$ with $c \notin J$ or $a \neq b$. Then $\mathrm{pos}_c^{w'}(b) = \eta'_c \circ \mathrm{pos}_c^w(b) = \eta'_c \circ \mathrm{pos}_c^v(b)$ since (iii) holds for the undashed elements. Since $\mathrm{pos}_c^v(b) \in \mathrm{dom}\, v_c$, we have $\eta'_c \circ \mathrm{pos}_c^v(b) = \eta_c \circ \mathrm{pos}_c^v(b) = \mathrm{pos}_c^{v'}(b)$, i.e. we showed (iii). To verify (iv), let $c \in J$. Then $\mathrm{pos}_c^{v'}(a) = \eta_c \circ \mathrm{pos}_c^v(a) = \eta'_c \circ \mathrm{pos}_c^v(a) < \eta'_c \circ \mathrm{pos}_c^w(a)$ since (iv) holds for the undashed elements and η'_c is an order embedding. Since $\eta'_c \circ \mathrm{pos}_c^w(a) = \mathrm{pos}_c^{w'}(a)$, we get $\mathrm{pos}_c^{v'}(a) < \mathrm{pos}_c^{w'}(a) \in \mathrm{im}\,\eta_c$. Since $\mathrm{pos}_c^w(a) \neq 0$ and η'_c is injective, we obtain $\mathrm{pos}_c^{w'}(a) = \eta'_c \circ \mathrm{pos}_c^w(a) \neq 0$, i.e. $\mathrm{pos}_c^{w'}(a) \in \mathrm{im}\,\eta_c \setminus \{0\} \subseteq \mathrm{dom}\, v'_c$. Finally, $v'_c \circ \mathrm{pos}_c^{v'}(a) = p'_c$ holds by the choice of p'_c. □

To apply Theorem 3.2.2 to the WSTS $(S, \to, \preceq)$, our next aim is to show that in $(S, \to, \preceq)$ a finite predecessor basis, i.e. a finite basis of $\mathrm{Pred}(\uparrow(w_c, \mathrm{pos}_c^w)_{c\in\Sigma})$, can be computed for any state $(w_c, \mathrm{pos}_c^w)_{c\in\Sigma}$. Note that $\uparrow(w_c, \mathrm{pos}_c^w)_{c\in\Sigma}$ in this expression is meant with respect to the wqo $\preceq$. Before we can prove this (cf. Lemma 3.2.8), we consider the quasiorder $\sqsubseteq$ on S: For $v, w \in Q_a^\star$, let $v \sqsubseteq'_a w$ iff $|v| = |w|$ and there exists an embedding of v into w. Note that whenever $v \sqsubseteq'_a w$ we can obtain w from v by simply enlarging the letters of v independently from each other. Since comparable words (with respect to $\sqsubseteq_a'$) have the same length, $\sqsubseteq_a'$ is only a quasiorder, but not a wqo. Similarly to $\sqsubseteq_a'$, we define $(v_a, \text{pos}_a^v)_{a\in\Sigma} \sqsubseteq (w_a, \text{pos}_a^w)_{a\in\Sigma}$ iff $|v_a| = |w_a|$ for $a \in \Sigma$ and $(v_a, \text{pos}_a^v)_{a\in\Sigma} \preceq (w_a, \text{pos}_a^w)_{a\in\Sigma}$.

We call a Σ-ACM *effective* if there is an algorithm that given $a \in \Sigma$, $J \subseteq \Sigma$, $p_b \in Q_b$ for $b \in J$ and $q \in Q_a$ computes a finite basis of

$$\left\{ ((p_b')_{b\in J}, q') \in \prod_{b\in J} Q_b \times Q_a \;\middle|\; q' \in \delta_{a,J}((p_b')_{b\in J}), q \sqsubseteq_a q' \text{ and } p_b \sqsubseteq_b p_b' \text{ for } b \in J \right\}$$

with respect to the direct product $\left(\prod_{b\in J} \sqsubseteq_b\right) \times \sqsubseteq_a$. We call such an algorithm a *basis algorithm of* $\mathcal{A}$. Intuitively, an ACM is effective if a finite basis of all transitions above a given tuple of states can be computed. Note that this tuple is not necessarily a transition. On the other hand, we do not require that the set of all transitions, i.e. the set $\{(q, (p_b)_{b\in J}) \mid q \in \delta_{a,J}((p_b)_{b\in J})\}$ is a recursive subset of $Q_a \times \prod_{b\in J} Q_b$, and this might not be the case as the following example shows. Furthermore note that any asynchronous cellular automaton is effective since (as a finite object) it can be given explicitly.

Example 3.2.5 Let $\Sigma = \{a\}$ and $Q_a = \mathbb{N}$. On this set, we consider the complete relation $\mathbb{N} \times \mathbb{N}$ as wqo $\sqsubseteq_a$. Furthermore, let M be some non recursive subset of $\mathbb{N}$ and define, for $n \in \mathbb{N}$:

$$\delta_{a,\{a\}}(n) = \begin{cases} \{n, n+1\} & \text{if } n \in M \\ \{n\} & \text{if } n \notin M. \end{cases}$$

Furthermore, let $\delta_{a,\emptyset} = \{1\}$. Now let $t = (V, E, \lambda)$ be a Σ-dag (i.e. t is the Hasse-diagram of a finite linear order) and let $r : V \to \mathbb{N}$ be some mapping. Then r is a run of the Σ-ACA $\mathcal{A} = (Q_a, (\delta_{a,J})_{J\subseteq\{a\}}, F)$ iff $r(x) \le r(x+1) \le r(x) + 1$ for any $x \in V$ and $\{x \in V \mid r(x) \ne r(x+1)\} \subseteq M$. Since this latter inclusion is not decidable, one cannot decide whether r is a run. On the other hand, $\mathcal{A}$ is effective since $\{(1, 1)\}$ is a finite basis of any nonempty subset of $Q_a \times Q_a$.

The preceding example suggests the question whether $L(\mathcal{A})$ is recursive for any monotone and effective Σ-ACM $\mathcal{A}$. Later (Corollary 3.3.5), we will show that this is indeed the case. Anyway, for an effective ACM, we can show:

Lemma 3.2.6 *There is an algorithm that, on input of an alphabet Σ, a basis algorithm for an effective Σ-ACM $\mathcal{A}$ and a state $(w_c, \text{pos}_c^w)_{c\in\Sigma} \in S$, outputs a finite basis with respect to $\sqsubseteq$ of the set of all states $(v_c', \text{pos}_c^{v'})_{c\in\Sigma} \in S$ satisfying*

$$\begin{array}{rcl} \exists (w_c', \text{pos}_c^{w'})_{c\in\Sigma} : \quad (v_c', \text{pos}_c^{v'})_{c\in\Sigma} & \longrightarrow & (w_c', \text{pos}_c^{w'})_{c\in\Sigma} \\ & & \sqcup\!\!\!| \\ & & (w_c, \text{pos}_c^w)_{c\in\Sigma} \end{array}$$

Proof. In this proof, we assume that $\operatorname{dom} w = \{1, 2, \ldots, \sup \operatorname{dom} w\}$ for any word $w \neq \varepsilon$.

First, we describe the algorithm:
For any $a \in \Sigma$ and any $\emptyset \neq J \subseteq \Sigma$ that satisfy
(a) $w_a \neq \varepsilon$ and $\sup \operatorname{dom} w_a \notin \operatorname{im} \operatorname{pos}_a^w$ and
(b) $\operatorname{pos}_b^w(a) \neq 0$ for $b \in J$
compute a finite basis $B(a, J)$ of the set of all tuples $((p'_b)_{b\in J}, q') \in \prod_{b\in J} Q_b \times Q_a$ satisfying
(c) $q' \in \delta_{a,J}((p'_b)_{b\in J}), w_a(\sup \operatorname{dom} w_a) \sqsubseteq_a q'$ and $w_b \circ \operatorname{pos}_b^w(a) \sqsubseteq_b p'_b$ for $b \in J$.
Such a finite basis can be computed by an application of the basis algorithm with $q = w_a(\sup \operatorname{dom} w_a)$ and $p_b = w_b(\operatorname{pos}_b^w(a))$ for $b \in J$.
For any $((p'_b)_{b\in J}, q') \in B(a, J)$, let $(w'_c, \operatorname{pos}_c^{w'})_{c\in\Sigma} \in S$ denote the uniquely determined state that satisfies
(d) $\operatorname{dom} w'_c = \operatorname{dom} w_c$ and $\operatorname{pos}_c^{w'} = \operatorname{pos}_c^w$ for $c \in \Sigma$ and
(e) $w'_c(i) = \begin{cases} p'_c & \text{if } c \in J, i = \operatorname{pos}_c^{w'}(a) \\ q' & \text{if } c = a, i = \sup \operatorname{dom} w'_a \\ w_c(i) & \text{otherwise.} \end{cases}$
For $c \in \Sigma$, let v'_c denote the word over Q_c uniquely determined by
(f) $\operatorname{dom} v'_c = \begin{cases} \operatorname{dom} w'_a \setminus \{\sup \operatorname{dom} w'_a\} & \text{for } c = a \\ \operatorname{dom} w'_c & \text{otherwise, and} \end{cases}$
(g) $v'_c = w'_c \restriction \operatorname{dom} v'_c$.
Finally, output the finite set of states $(v'_c, \operatorname{pos}_c^{v'})_{c\in\Sigma}$ that satisfy
(h) $\operatorname{pos}_c^{v'}(b) = \operatorname{pos}_c^{w'}(b)$ for $b, c \in \Sigma$ with $c \notin J$ or $a \neq b$ and
(j) $\operatorname{pos}_c^{v'}(a) < \operatorname{pos}_c^{w'}(a)$ for $c \in J$.

First we show that for any $(v'_c, \operatorname{pos}_c^{v'})_{c\in\Sigma}$ that is output by the algorithm above we have $(v'_c, \operatorname{pos}_c^{v'})_{c\in\Sigma} \to (w'_c, \operatorname{pos}_c^{w'})_{c\in\Sigma} \sqsupseteq (w_c, \operatorname{pos}_c^w)_{c\in\Sigma}$:

Since $(v'_c, \operatorname{pos}_c^{v'})_{c\in\Sigma}$ is output, there exist $a \in \Sigma$, $\emptyset \neq J \subseteq \Sigma$, $p'_b \in Q_b$ for $b \in J$ with $((p'_b)_{b\in J}, q') \in B(a, J)$ and $q' \in Q_a$ such that (a)-(j) hold. For $c \in \Sigma$, the identity function $\eta_c : \operatorname{dom} w_c \cup \{0\} \to \operatorname{dom} w'_c \cup \{0\}$ satisfies $\eta_c \circ \operatorname{pos}_c^w = \operatorname{pos}_c^{w'}$ by (d). By (c) and (e), we obtain $w_c(i) \sqsubseteq_c w'_c \circ \eta_c(i)$ for $i \in \operatorname{dom} w_c$. Hence $(w_c, \operatorname{pos}_c^w)_{c\in\Sigma} \preceq (w'_c, \operatorname{pos}_c^{w'})_{c\in\Sigma}$. Since in addition $|w_c| = |w'_c|$, we get $(w_c, \operatorname{pos}_c^w)_{c\in\Sigma} \sqsubseteq (w'_c, \operatorname{pos}_c^{w'})_{c\in\Sigma}$.

It remains to show that (i)-(iv) (page 25) hold for the states $(v'_c, \operatorname{pos}_c^{v'})_{c\in\Sigma}$ and $(w'_c, \operatorname{pos}_c^{w'})_{c\in\Sigma}$ and for a, J, p'_b for $b \in J$ and q':
(i): This is immediate since (c) holds.
(ii): If $c \neq a$, (f) and (g) imply $v'_c = w'_c$. Furthermore, these two statements also ensure $w'_a = v'_a\, w'_a(\sup \operatorname{dom} w'_a) = v'_a\, q'$ by (e).
(iii): This is immediate by (h).
(iv): Let $c \in J$. Then, by (j), $\operatorname{pos}_c^{v'}(a) < \operatorname{pos}_c^{w'}(a)$. Hence $\operatorname{pos}_c^{w'}(a) \neq 0$ and therefore $\operatorname{pos}_c^{w'}(a) \in \operatorname{dom} w'_c$. For $c \neq a$, this implies $\operatorname{pos}_c^{w'} \in \operatorname{dom} v'_c$ since, by (f), $\operatorname{dom} v'_c = \operatorname{dom} w'_c$. To deal with the case $a = c$, recall that $\operatorname{pos}_a^w(a) \neq \sup \operatorname{dom} w_a$ by (a). Hence, from (d), we can infer $\operatorname{pos}_y^{w'}(a) \neq \sup \operatorname{dom} w'_a$ and

therefore $\mathrm{pos}_a^{w'}(a) \in \mathrm{dom}\, w'_a \setminus \{\sup \mathrm{dom}\, w'_a\} = \mathrm{dom}\, v'_a$ by (f). Thus, we showed $\mathrm{pos}_c^{v'}(a) < \mathrm{pos}_c^{w'}(a) \in \mathrm{dom}\, v'_c$ for $c \in J$. Again, let $c \in J$. Then $v'_c(\mathrm{pos}_c^{w'}(a)) = w'_c(\mathrm{pos}_c^{w'}(a)) = p'_c$ by (e).

It remains to show that a state $(v''_c, \mathrm{pos}_c^{v''})_{c\in\Sigma} \in S$ dominates some output of our algorithm whenever there exists a state $(w''_c, \mathrm{pos}_c^{w''})_{c\in\Sigma} \in S$ such that:

$$\begin{array}{ccc} (v''_c, \mathrm{pos}_c^{v''})_{c\in\Sigma} & \longrightarrow & (w''_c, \mathrm{pos}_c^{w''})_{c\in\Sigma} \\ & & \sqsupseteq \\ & & (w_c, \mathrm{pos}_c^{w})_{c\in\Sigma} \end{array}$$

Since $(v''_c, \mathrm{pos}_c^{v''})_{c\in\Sigma} \to (w''_c, \mathrm{pos}_c^{w''})_{c\in\Sigma}$, there exist $a \in \Sigma$, $\emptyset \neq J \subseteq \Sigma$, $p''_b \in Q_b$ for $b \in J$ and $q'' \in Q_a$ satisfying (i)-(iv). We show that a and J satisfy (a) and (b):

(a): Since $(w''_c, \mathrm{pos}_c^{w''})_{c\in\Sigma} \sqsupseteq (w_c, \mathrm{pos}_c^{w})_{c\in\Sigma}$, it holds $|w_a| = |w''_a| = |v''_a q''| > 0$ by (ii) and therefore $w_a \neq \varepsilon$. Furthermore, $\mathrm{im}\, \mathrm{pos}_a^{w''} \subseteq \mathrm{dom}\, v''_a \cup \{0\}$ by (iii) and (iv). But $\mathrm{dom}\, v''_a = \mathrm{dom}\, w''_a \setminus \{\sup \mathrm{dom}\, w''_a\}$ by (ii) and therefore $\sup \mathrm{dom}\, w_a \notin \mathrm{im}\, \mathrm{pos}_a^{w}$.
(b): Let $c \in J$. Then $\mathrm{pos}_c^{w''}(a) \in \mathrm{dom}\, v''_c \not\ni 0$ by (iv). Hence $\mathrm{pos}_c^{w''}(a) \neq 0$ which does not belong to $\mathrm{dom}\, v''_c$.

Furthermore note that $w_b(\mathrm{pos}_b^{w}(a)) \sqsubseteq_b w''_b(\mathrm{pos}_b^{w''}(a)) = v''_b(\mathrm{pos}_b^{w''}(a)) = p''_b$ by (iv) for any $b \in J$. Similarly, $w_a(\sup \mathrm{dom}\, w_a) \sqsubseteq_a w''_a(\sup \mathrm{dom}\, w''_a) = q''$ by (iv) and (by (i)) $q'' \in \delta_{a,J}((p''_b)_{b\in J})$. Since $B(a, J)$ is a basis, there is $((p'_b)_{b\in\Sigma}, q') \in B(a, J)$ such that $p'_b \sqsubseteq_b q''_b$ for $b \in J$, $q' \sqsubseteq_a q''$ and (c) holds. Now construct $(w'_c, \mathrm{pos}_c^{w'})_{c\in\Sigma}$ and v'_c for $c \in \Sigma$ according to (d)-(g) and set $\mathrm{pos}_c^{v'} = \mathrm{pos}_c^{v''}$. To show (h), let $b, c \in \Sigma$ with $c \notin J$ or $a \neq b$. Then, by (iii), $\mathrm{pos}_c^{v''}(b) = \mathrm{pos}_c^{w''}(b)$. Since $(w_d, \mathrm{pos}_d^{w})_{d\in\Sigma} \sqsubseteq (w''_d, \mathrm{pos}_d^{w''})_{d\in\Sigma}$ is witnessed by the identity functions, we get $\mathrm{pos}_c^{v''}(b) = \mathrm{pos}_c^{w''}(b) = \mathrm{pos}_c^{w}(b) = \mathrm{pos}_c^{w'}(b)$ where the last equality holds by (d). Thus, (h) holds. To show (j), let $c \in J$. Then, by (iv), $\mathrm{pos}_c^{v''}(a) < \mathrm{pos}_c^{w''}(a)$ and we can continue as above by $\mathrm{pos}_c^{v''}(a) < \mathrm{pos}_c^{w''}(a) = \mathrm{pos}_c^{w}(a) = \mathrm{pos}_c^{w'}(a)$ thereby proving (j). Hence $h := (v'_c, \mathrm{pos}_c^{v''})_{c\in\Sigma}$ is a state from S that is output by our algorithm. It remains to check $h \sqsubseteq (v''_c, \mathrm{pos}_c^{v''})_{c\in\Sigma}$ which is left to the interested reader. □

Lemma 3.2.7 *Let* $(x_c, \mathrm{pos}_c^x)_{c\in\Sigma}$, $(w''_c, \mathrm{pos}_c^{w''})_{c\in\Sigma}$ *and* $(v''_c, \mathrm{pos}_c^{v''})_{c\in\Sigma}$ *be states from* S *with*

$$\begin{array}{ccc} (v''_c, \mathrm{pos}_c^{v''})_{c\in\Sigma} & \to & (w''_c, \mathrm{pos}_c^{w''})_{c\in\Sigma} \\ & & \rotatebox{90}{$\preceq$} \\ & & (x_c, \mathrm{pos}_c^{x})_{c\in\Sigma} \end{array}$$

Then there exist states $(w_c, \mathrm{pos}_c^w)_{c\in\Sigma}$, $(v'_c, \mathrm{pos}_c^{v'})_{c\in\Sigma}$ *and* $(w'_c, \mathrm{pos}_c^{w'})_{c\in\Sigma}$ *such that*

1. $|w_c| - |x_c| \le 2|\Sigma| + 1$ *for* $c \in \Sigma$ *and*
2. $$\begin{array}{ccc} (v''_c, \mathrm{pos}_c^{v''})_{c\in\Sigma} & \to & (w''_c, \mathrm{pos}_c^{w''})_{c\in\Sigma} \\ \rotatebox{90}{$\preceq$} & & \rotatebox{90}{$\preceq$} \\ (v'_c, \mathrm{pos}_c^{v'})_{c\in\Sigma} & \to & (w'_c, \mathrm{pos}_c^{w'})_{c\in\Sigma} \\ & & \rotatebox{90}{$\sqsubseteq$} \\ & & (w_c, \mathrm{pos}_c^{w})_{c\in\Sigma} \\ & & \rotatebox{90}{$\preceq$} \\ & & (x_c, \mathrm{pos}_c^{x})_{c\in\Sigma} \end{array}$$

Proof. Since $(v''_c, \mathrm{pos}_c^{v''})_{c\in\Sigma} \to (w''_c, \mathrm{pos}_c^{w''})_{c\in\Sigma}$, there are $a \in \Sigma$, $\emptyset \neq J \subseteq \Sigma$, $p_b \in Q_b$ for $b \in J$ and $q \in Q_a$ such that (i)-(iv) (page 25) hold. Let $\eta_c : x_c \hookrightarrow w''_c$ be embeddings that witness $(x_c, \mathrm{pos}_c^x)_{c\in\Sigma} \preceq (w''_c, \mathrm{pos}_c^{w''})_{c\in\Sigma}$. We may assume that η_c is just the identity function, i.e. $\mathrm{dom}\, x_c \subseteq \mathrm{dom}\, w''_c$, $x_c(i) \sqsubseteq_c w''_c(i)$ for $i \in \mathrm{dom}\, x_c$, $\sup\mathrm{dom}\, x_c = \sup\mathrm{dom}\, w''_c$, and $\mathrm{pos}_c^x = \mathrm{pos}_c^{w''}$ for $c \in \Sigma$.

First, we define $(w'_c, \mathrm{pos}_c^{w'})_{c\in\Sigma}$: For $c \in \Sigma$, let

$$\mathrm{dom}\, w'_c := (\mathrm{dom}\, x_c \cup \mathrm{im}\,\mathrm{pos}_c^{v''} \cup \mathrm{im}\,\mathrm{pos}_c^{w''} \cup \{\sup\mathrm{dom}\, v''_c\}) \setminus \{0\}$$

and $w'_c = w''_c \upharpoonright \mathrm{dom}\, w'_c$. Furthermore, let $\mathrm{pos}_c^{w'} := \mathrm{pos}_c^{w''} = \mathrm{pos}_c^x$. Then $\mathrm{im}\,\mathrm{pos}_c^{w'} \subseteq \mathrm{dom}\, x_c \cup \{0\} \subseteq \mathrm{dom}\, w'_c \cup \{0\}$ ensures $(w'_c, \mathrm{pos}_c^{w'})_{c\in\Sigma} \in S$.

We show $(w'_c, \mathrm{pos}_c^{w'})_{c\in\Sigma} \preceq (w''_c, \mathrm{pos}_c^{w''})_{c\in\Sigma}$: Note that $\mathrm{dom}\, x_c \subseteq \mathrm{dom}\, w''_c$. Furthermore, $\mathrm{im}\,\mathrm{pos}_c^{v''} \subseteq \mathrm{dom}\, v''_c \cup \{0\}$ and $(v''_c, \mathrm{pos}_c^{v''})_{c\in\Sigma} \to (w''_c, \mathrm{pos}_c^{w''})_{c\in\Sigma}$ imply $\mathrm{im}\,\mathrm{pos}_c^{v''} \setminus \{0\} \subseteq \mathrm{dom}\, w''_c$. Since $\mathrm{im}\,\mathrm{pos}_c^{w''} \setminus \{0\} \subseteq \mathrm{dom}\, w''_c$ and $\sup\mathrm{dom}\, v''_c \in \mathrm{dom}\, w''_c \cup \{0\}$, we therefore get $\mathrm{dom}\, w'_c \subseteq \mathrm{dom}\, w''_c$. Thus, the identity function $\eta'_c := \mathrm{id}_{\mathrm{dom}\, w'_c \cup \{0\}} : \mathrm{dom}\, w'_c \cup \{0\} \to \mathrm{dom}\, w''_c \cup \{0\}$ is an order embedding that satisfies $w'_c(i) = w''_c \circ \eta'_c(i)$ for $i \in \mathrm{dom}\, w'_c$. Since $(v''_c, \mathrm{pos}_c^{v''})_{c\in\Sigma} \to (w''_c, \mathrm{pos}_c^{w''})_{c\in\Sigma}$, we have $\mathrm{dom}\, v''_c \subseteq \mathrm{dom}\, w''_c$ and therefore $\sup\mathrm{dom}\, v''_c \le \mathrm{dom}\,\mathrm{dom}\, w''_c = \sup\mathrm{dom}\, x_c$. Hence $\sup\mathrm{dom}\, w'_c = \sup\mathrm{dom}\, x_c = \sup\mathrm{dom}\, w''_c$. Thus, $\eta' : w'_c \hookrightarrow w''_c$ is an embedding. Since $\mathrm{pos}_c^{w'} = \mathrm{pos}_c^{w''}$, we in addition get $\mathrm{pos}_c^{w''} = \eta'_c \circ \mathrm{pos}_c^{w'}$ implying $(w'_c, \mathrm{pos}_c^{w'})_{c\in\Sigma} \preceq (w''_c, \mathrm{pos}_c^{w''})_{c\in\Sigma}$.

Next, define $(w_c, \mathrm{pos}_c^w)_{c\in\Sigma}$ by $\mathrm{dom}\, w_c = \mathrm{dom}\, w'_c$, $\mathrm{pos}_c^w := \mathrm{pos}_c^{w'}$ and

$$w_c(i) := \begin{cases} x_c(i) & \text{if } i \in \mathrm{dom}\, x_c \\ w'_c(i) & \text{otherwise.} \end{cases}$$

Again, since $\operatorname{im}\operatorname{pos}_c^w = \operatorname{im}\operatorname{pos}_c^{w'} \subseteq \operatorname{dom} w'_c \cup \{0\} = \operatorname{dom} w_c \cup \{0\}$, the tuple $(w_c, \operatorname{pos}_c^w)_{c\in\Sigma}$ belongs to S. Furthermore $|\operatorname{dom} w_c| = |\operatorname{dom} w'_c| \leq |\operatorname{dom} x_c| + |\operatorname{im}\operatorname{pos}_c^{w''}| + |\operatorname{im}\operatorname{pos}_c^{v''}| + 1$ implies $|w_c| - |x_c| \leq 2|\Sigma| + 1$. Thus, the first statement holds.

Note that $\operatorname{dom} x_c \subseteq \operatorname{dom} w'_c = \operatorname{dom} w_c$. Furthermore, we showed above $\sup\operatorname{dom} x_c = \sup\operatorname{dom} w'_c$; hence $\sup\operatorname{dom} x_c = \sup\operatorname{dom} w_c$. Finally, for $i \in \operatorname{dom} x_c$, we have $x_c(i) = w_c(i)$. Thus, the identity function $\operatorname{dom} x_c \cup \{0\} \to \operatorname{dom} w_c \cup \{0\}$ is an embedding of x_c into w_c. Since, in addition, $\operatorname{pos}_c^w = \operatorname{pos}_c^{w'} = \operatorname{pos}_c^x$, we get $(x_c, \operatorname{pos}_c^x)_{c\in\Sigma} \preceq (w_c, \operatorname{pos}_c^w)_{c\in\Sigma}$.

For $i \in \operatorname{dom} x_c$, we have $w_c(i) = x_c(i) \sqsubseteq_c w''_c(i) = w'_c(i)$. Now $(w_c, \operatorname{pos}_c^w)_{c\in\Sigma} \sqsubseteq (w'_c, \operatorname{pos}_c^{w'})_{c\in\Sigma}$ follows immediately since $w_c(i) = w'_c(i)$ for $i \in \operatorname{dom} w'_c \setminus \operatorname{dom} x_c$, $\operatorname{dom} w_c = \operatorname{dom} w'_c$ and $\operatorname{pos}_c^w = \operatorname{pos}_c^{w'}$.

Finally, we construct $(v'_c, \operatorname{pos}_c^{v'})_{c\in\Sigma}$: Let $\operatorname{dom} v'_c := \operatorname{dom} w'_c \cap \operatorname{dom} v''_c$, and define $v'_c := w'_c \restriction \operatorname{dom} v'_c$ and $\operatorname{pos}_c^{v'} := \operatorname{pos}_c^{v''}$ for $c \in \Sigma$. Then $(v'_c, \operatorname{pos}_c^{v'})_{c\in\Sigma} \in S$ since $\operatorname{im}\operatorname{pos}_c^{v'} = \operatorname{im}\operatorname{pos}_c^{v''} \subseteq (\operatorname{dom} w'_c \cap \operatorname{dom} v''_c) \cup \{0\} = \operatorname{dom} v'_c \cup \{0\}$. For $i \in \operatorname{dom} v'_c$, we have $v'_c(i) = w'_c(i) = w''_c(i)$ by the definition of v'_c and of w'_c, respectively. In addition, $i \in \operatorname{dom} v''_c$ and, from $(v''_c, \operatorname{pos}_c^{v''})_{c\in\Sigma} \to (w''_c, \operatorname{pos}_c^{w''})_{c\in\Sigma}$, we obtain $w'_c(i) = v''_c(i)$ i.e. we showed $v'_c(i) = v''_c(i)$.

Now let $c \neq a$. Above, we showed $\sup\operatorname{dom} w'_c = \sup\operatorname{dom} w''_c$. We infer $v''_c = w''_c$ from $(v''_c, \operatorname{pos}_c^{v''})_{c\in\Sigma} \to (w''_c, \operatorname{pos}_c^{w''})_{c\in\Sigma}$. Therefore $\sup\operatorname{dom} w'_c = \sup\operatorname{dom} v''_c$. Hence $\operatorname{dom} v'_c = \operatorname{dom} w'_c \cap \operatorname{dom} v''_c$ implies $\sup\operatorname{dom} v'_c = \sup\operatorname{dom} v''_c$. Thus, for $c \neq a$, the identity function $\operatorname{dom} v'_c \cup \{0\} \to \operatorname{dom} v''_c \cup \{0\}$ is an embedding of v'_c into v''_c. Next we show this fact for $c = a$: Since $\operatorname{dom} v'_a = \operatorname{dom} w'_a \cap \operatorname{dom} v''_a$, we obtain $\operatorname{dom} v'_a \leq \sup\operatorname{dom} v''_a$. Furthermore, $\sup\operatorname{dom} v''_a \in \operatorname{dom} w'_a \cup \{0\}$ and $\sup\operatorname{dom} \in \operatorname{dom} v''_a \cup \{0\}$ imply $\sup\operatorname{dom} v''_a \in \operatorname{dom} v'_a \cup \{0\}$. Hence $\sup\operatorname{dom} v''_a = \sup\operatorname{dom} v'_a$. Thus, indeed, the identity function $\operatorname{dom} v'_c \cup \{0\} \to \operatorname{dom} v''_c \cup \{0\}$ is an embedding of v'_c into v''_c for any $c \in \Sigma$. Since $\operatorname{pos}_c^{v''} = \operatorname{pos}_c^{v'}$, we have $(v'_c, \operatorname{pos}_c^{v'})_{c\in\Sigma} \preceq (v''_c, \operatorname{pos}_c^{v''})_{c\in\Sigma}$ as required.

It remains to show $(v'_c, \operatorname{pos}_c^{v'})_{c\in\Sigma} \to (w'_c, \operatorname{pos}_c^{w'})_{c\in\Sigma}$, i.e. that (ii)-(iv) hold for the states $(v'_c, \operatorname{pos}_c^{v'})_{c\in\Sigma}$ and $(w'_c, \operatorname{pos}_c^{w'})_{c\in\Sigma}$ and for a, J, p_b for $b \in J$ and q:
(ii) For $c \neq a$ we have $\operatorname{dom} v'_c = \operatorname{dom} w'_c \cap \operatorname{dom} v''_c = \operatorname{dom} w'_c \cap \operatorname{dom} w''_c$ since $\operatorname{dom} v''_c = \operatorname{dom} w''_c$ follows from $(v''_c, \operatorname{pos}_c^{v''})_{c\in\Sigma} \to (w''_c, \operatorname{pos}_c^{w''})_{c\in\Sigma}$. Now $\operatorname{dom} w'_c \subseteq \operatorname{dom} w''_c$ implies $\operatorname{dom} v'_c = \operatorname{dom} w'_c$. Thus $v'_c = w'_c \restriction \operatorname{dom} v'_c = w'_c$. Similarly, we obtain $\operatorname{dom} v'_a = \operatorname{dom} w'_a \cap \operatorname{dom} v''_a = \operatorname{dom} w'_a \cap (\operatorname{dom} w''_a \setminus \{\sup\operatorname{dom} w''_a\})$. Recall that $\sup\operatorname{dom} w'_a = \sup\operatorname{dom} w''_a$ and therefore $\operatorname{dom} v'_a = \operatorname{dom} w'_a \setminus \{\sup\operatorname{dom} w'_a\}$. Since $w'_a(\sup\operatorname{dom} w'_a) = q$, we obtain $w'_a = v'_a q$ from $v'_a = w'_a \restriction \operatorname{dom} v'_a$.
(iii) Let $b, c \in \Sigma$ with $c \notin J$ or $a \neq b$. Then $\operatorname{pos}_c^{v'}(b) = \operatorname{pos}_c^{v''}(b)$ and $\operatorname{pos}_c^{w''}(b) = \operatorname{pos}_c^{w'}(b)$. Using (iii) for the states $(v''_c, \operatorname{pos}_c^{v''})_{c\in\Sigma}$ and $(w''_c, \operatorname{pos}_c^{w''})_{c\in\Sigma}$, we obtain $\operatorname{pos}_c^{v''}(b) = \operatorname{pos}_c^{w''}(b)$ and therefore $\operatorname{pos}_c^{v'}(b) = \operatorname{pos}_c^{w'}(b)$ as required.
(iv) Let $c \in J$. Since (iv) holds for the states $(v''_c, \operatorname{pos}_c^{v''})_{c\in\Sigma}$ and $(w''_c, \operatorname{pos}_c^{w''})_{c\in\Sigma}$, we get $\operatorname{pos}_c^{v'}(a) = \operatorname{pos}_c^{v''}(a) < \operatorname{pos}_c^{w''}(a) = \operatorname{pos}_c^{w'}(a)$ and $\operatorname{pos}_c^{w''}(a) \in \operatorname{dom} v''_c$. Since, in addition, $\operatorname{pos}_c^{w'}(a) \in \operatorname{dom} w'_c \cup \{0\}$, we infer $\operatorname{pos}_c^{w'}(a) \in \operatorname{dom} v''_c \cap (\operatorname{dom} w'_c \cup \{0\}) = \operatorname{dom} v''_c \cap \operatorname{dom} w'_c = \operatorname{dom} v'_c$. Finally, we get $v'_c \circ \operatorname{pos}_c^{w'}(a) = v''_c \circ \operatorname{pos} w''_c(a) = p_c$. $\square$

Lemma 3.2.8 *There exists an algorithm that solves the following problem:*
input: *1. an alphabet* Σ*,*
2. a basis algorithm of an effective and monotone Σ*-ACM* $\mathcal{A}$*,*
3. a finite basis B_c *of* $(Q_c, \sqsubseteq_c)$ *and an algorithm to decide* $\sqsubseteq_c$ *for* $c \in \Sigma$*, and*
4. a state $(x_c, \mathrm{pos}_c^x)_{c\in\Sigma} \in S$
output: *a finite basis of the set* $\mathrm{Pred}(\uparrow(x_c, \mathrm{pos}_c^x)_{c\in\Sigma})$*.*

Proof. For simplicity, let M denote the set $\mathrm{Pred}(\uparrow(x_c, \mathrm{pos}_c^x)_{c\in\Sigma})$. Let H be the finite set of all states $(w_c, \mathrm{pos}_c^w)_{c\in\Sigma}$ in S that satisfy

$\mathrm{dom}\, x_c \subseteq \mathrm{dom}\, w_c$,
$w_c(i) = x_c(i)$ for $i \in \mathrm{dom}\, x_c$ and $w_c(i) \in B_c$ otherwise,
$\mathrm{pos}_c^x = \mathrm{pos}_c^w$ and
$|w_c| - |x_c| \leq 2|\Sigma| + 1$ for $c \in \Sigma$.

Note that H can be computed effectively. Furthermore, the identity functions witness $(x_c, \mathrm{pos}_c^x)_{c\in\Sigma} \preceq (w_c, \mathrm{pos}_c^w)_{c\in\Sigma}$ for $(w_c, \mathrm{pos}_c^w)_{c\in\Sigma} \in H$.

For $(w_c, \mathrm{pos}_c^w)_{c\in\Sigma} \in H$, by Lemma 3.2.6, we can compute a finite basis $B((w_c, \mathrm{pos}_c^w)_{c\in\Sigma})$ (with respect to $\sqsubseteq$) of the set of all states $(v'_c, \mathrm{pos}_c^{v'})_{c\in\Sigma}$ satisfying

$$\exists (w'_c, \mathrm{pos}_c^{w'})_{c\in\Sigma} \in S: \quad \begin{array}{ccc} (v'_c, \mathrm{pos}_c^{v'})_{c\in\Sigma} & \to & (w'_c, \mathrm{pos}_c^{w'})_{c\in\Sigma} \\ & & \sqcup\!\!| \\ & & (w_c, \mathrm{pos}_c^w)_{c\in\Sigma}. \end{array}$$

Let $(v'_c, \mathrm{pos}_c^{v'})_{c\in\Sigma} \in B((w_c, \mathrm{pos}_c^w)_{c\in\Sigma})$. Then there exists a state $(w'_c, \mathrm{pos}_c^{w'})_{c\in\Sigma}$ that is a successor of $(v'_c, \mathrm{pos}_c^{v'})_{c\in\Sigma}$ and dominates $(w_c, \mathrm{pos}_c^w)_{c\in\Sigma}$ with respect to $\sqsubseteq$. Since $(x_c, \mathrm{pos}_c^x)_{c\in\Sigma} \preceq (w_c, \mathrm{pos}_c^w)_{c\in\Sigma}$, we therefore get $(x_c, \mathrm{pos}_c^x)_{c\in\Sigma} \preceq (w'_c, \mathrm{pos}_c^{w'})_{c\in\Sigma}$. But this implies $(v'_c, \mathrm{pos}_c^{v'})_{c\in\Sigma} \in \mathrm{Pred}(\uparrow(x_c, \mathrm{pos}_c^x)_{c\in\Sigma})$, i.e. we showed $B((w_c, \mathrm{pos}_c^w)_{c\in\Sigma}) \subseteq \mathrm{Pred}(\uparrow(x_c, \mathrm{pos}_c^x)_{c\in\Sigma}) = M$. Now define

$$B := \bigcup_{(w_c, \mathrm{pos}_c^w)_{c\in\Sigma} \in H} B((w_c, \mathrm{pos}_c^w)_{c\in\Sigma}).$$

It remains to show that B is a basis of M: So let $(v'_c, \mathrm{pos}_c^{v'})_{c\in\Sigma} \in B$. Then there exist $(w_c, \mathrm{pos}_c^w)_{c\in\Sigma} \in H$ and $(w'_c, \mathrm{pos}_c^{w'})_{c\in\Sigma} \in S$ such that

$$\begin{array}{ccc} (v'_c, \mathrm{pos}_c^{v'})_{c\in\Sigma} & \to & (w'_c, \mathrm{pos}_c^{w'})_{c\in\Sigma} \\ & & \sqcup\!\!| \\ & & (w_c, \mathrm{pos}_c^w)_{c\in\Sigma} \\ & & \curlyvee\!| \\ & & (x_c, \mathrm{pos}_c^x)_{c\in\Sigma}. \end{array}$$

Hence $(w'_c, \mathrm{pos}_c^{w'})_{c\in\Sigma} \succeq (x_c, \mathrm{pos}_c^{x})_{c\in\Sigma}$ and therefore $(v'_c, \mathrm{pos}_c^{v'})_{c\in\Sigma} \in M$, i.e. we showed $B \subseteq M$ which implies $\uparrow B \subseteq \uparrow M$.

Now let $(v''_c, \mathrm{pos}_c^{v''})_{c\in\Sigma} \in M$. Then there exists $(w''_c, \mathrm{pos}_c^{w''})_{c\in\Sigma} \in S$ such that $(v''_c, \mathrm{pos}_c^{v''})_{c\in\Sigma} \to (w''_c, \mathrm{pos}_c^{w''})_{c\in\Sigma} \succeq (x_c, \mathrm{pos}_c^{x})_{c\in\Sigma}$. Hence, by Lemma 3.2.7, there are $(w_c, \mathrm{pos}_c^{w})_{c\in\Sigma} \in H$, $(v'_c, \mathrm{pos}_c^{v'})_{c\in\Sigma} \in B$ and $(w'_c, \mathrm{pos}_c^{w'})_{c\in\Sigma} \in S$ with

$$\begin{array}{ccc}
(v''_c, \mathrm{pos}_c^{v''})_{c\in\Sigma} & \to & (w''_c, \mathrm{pos}_c^{w''})_{c\in\Sigma} \\
\rotatebox[origin=c]{-90}{$\succeq$} & & \rotatebox[origin=c]{-90}{$\succeq$} \\
(v'_c, \mathrm{pos}_c^{v'})_{c\in\Sigma} & \to & (w'_c, \mathrm{pos}_c^{w'})_{c\in\Sigma} \\
 & & \rotatebox[origin=c]{-90}{$\sqsupseteq$} \\
 & & (w_c, \mathrm{pos}_c^{w})_{c\in\Sigma} \\
 & & \rotatebox[origin=c]{-90}{$\succeq$} \\
 & & (x_c, \mathrm{pos}_c^{x})_{c\in\Sigma}.
\end{array}$$

Hence $v'' \in \uparrow B$ and therefore $M \subseteq \uparrow B$. Since this trivially implies $\uparrow M \subseteq \uparrow B$, the set B is indeed a finite basis of M. □

3.3 The emptiness is decidable for ACMs

To apply the decidability result of Finkel and Schnoebelen (Theorem 3.2.2) to Σ-ACMs, we have to relate runs of a Σ-ACM and paths in the transition system $(S, \to)$. Roughly speaking, states of the form $\mathrm{state}(r)$ for some run r correspond to reachable states in $(S, \to)$. Unfortunately, the truth is not that simple. Therefore, we need some more notions: Let Σ be an alphabet. A *weak Σ-dag* is a triple (V, E, λ) where (V, E) is a finite directed acyclic graph and $\lambda : V \to \Sigma$ is a labeling function such that

1. for all $x, y \in \min(V, E^\star)$ with $\lambda(x) = \lambda(y)$, we have $x = y$, and
2. for any $(x, y), (x', y') \in E$ with $\lambda(x) = \lambda(x')$, $\lambda(y) = \lambda(y')$, we have $x = x'$ if and only if $y = y'$.

Note that any Σ-dag is a weak Σ-dag. Similarly to Σ-dags, we can define $\mathrm{R}(y)$ for a node y in a weak Σ-dag (V, E, λ) to be the set of all labels $\lambda(x)$ with $(x, y) \in E$. Since in a weak Σ-dag for any node y and any $a \in \mathrm{R}(y)$ there is a unique node x with $\lambda(x) = a$ and $(x, y) \in E$, we can also use the notion $\partial_a(y)$ to denote this vertex. Hence, for a Σ-ACM $\mathcal{A}$, we can speak of a mapping $r : V \to Q$ that satisfies the run condition at a node $x \in V$ relative to t.

Lemma 3.3.1 *There exists an algorithm that on input of an alphabet Σ and a function $f : \Sigma \to \Sigma$ outputs an asynchronous cellular automaton $\mathcal{A}(f)$ such that*

1. $\bigcup_{f\in\Sigma^\Sigma} L(\mathcal{A}(f)) = \mathbb{D}$, *and*
2. *for any weak Σ-dag $t = (V, E, \lambda)$, any $f : \Sigma \to \Sigma$ and any mapping r that satisfies the run condition of $\mathcal{A}(f)$ for any $x \in V$ relative to t, the set $\lambda^{-1}(a)$ is a chain w.r.t. $E^\star$ for any $a \in \Sigma$.*

Proof. First, we give the construction of the ACAs $\mathcal{A}(f)$: Let $f : \Sigma \to \Sigma$. The set of local states shared by all processes equals the set of nonempty partial functions from Σ to itself, i.e. $Q = Q_a = \text{part}(\Sigma, \Sigma)$ for $a \in \Sigma$. The transition functions $\delta_{a,J}$ are defined by

$$\delta_{a,\emptyset} = \{g \in \text{part}(\Sigma, \Sigma) \mid a \in \text{dom}(g) = f^{-1}(a) \neq \emptyset\}$$

and for $J \neq \emptyset$ by

$$g \in \delta_{a,J}((g_b)_{b\in J}) \iff a \in \text{dom}(g) \text{ and } (\forall c \in \text{dom}(g)\exists b \in J : g_b(c) = a)$$

for $g_b \in \text{part}(\Sigma, \Sigma)$ for $b \in J$. Finally, all tuples of local states are accepting.

To show the first statement, let $t = (V, E, \lambda) \in \mathbb{D}$ be a Σ-dag. Since t is a Σ-dag, nodes that carry the same label are linearly ordered with respect to $E^\star$. Hence, we can choose maximal chains $C_a \subseteq V$ with $\lambda^{-1}(a) \subseteq C_a$ for any $a \in \Sigma$. Note that the minimal node of the chain C_a is minimal in t. We set $f(a) := \lambda(\min C_a)$ and obtain a function $f : \Sigma \to \Sigma$. To prove the first statement, it remains to show that $\mathcal{A}(f)$ accepts t: We define a mapping $r : V \to Q = \text{part}(\Sigma, \Sigma)$ with $\text{dom}(r(x)) = \{a \in \Sigma \mid x \in C_a\}$. Now let $x \in V$ and $a \in \text{dom}(r(x))$. If there exists $y \in C_a$ with $(x, y) \in E$, then there exists a least such node y since C_a is a chain. Let $r(x)(a)$ be the label of this minimal node. If there is no such node y, define $r(x)(a) := a$. Since $x \in C_{\lambda(x)}$ for any $x \in V$, the function $r(x)$ is indeed nonempty and therefore belongs to Q. Now let $y \in V$ be some node with $a = \lambda(y)$. We want to show that r satisfies the run condition of $\mathcal{A}(f)$ at y relative to t: First let y be minimal in t. Since $\lambda^{-1}(a) \subseteq C_a$, we get $a \in \text{dom}(r(y))$. Now let $b \in f^{-1}(a)$, i.e. $f(b) = a$. Then by the choice of f, we get $a = \lambda(\min C_b)$. Since C_b is a maximal chain, the node $\min C_b$ is minimal in t. Since t is a Σ-dag, its minimal nodes carry mutually different labels. Hence $y = \min C_b \in C_b$. This implies $b \in \text{dom}(r(y))$ and therefore $f^{-1}(a) \subseteq \text{dom}(r(y))$. Conversely let $b \in \text{dom}(r(y))$. Then $y \in C_b$ and, since y is minimal in t, $y = \min C_b$. Hence $a = \lambda(y) = \lambda(\min C_b) = f(b)$ ensures $\text{dom}(r(y)) \subseteq f^{-1}(a)$. Thus, the mapping r satisfies the run condition of $\mathcal{A}(f)$ at the minimal nodes of t relative to t. Now let $y \in V$ be nonminimal. Then $J := \text{R}(y) \neq \emptyset$. Since $y \in \lambda^{-1}(a) \subseteq C_a$, we get $a \in \text{dom}(r(y))$. Now let $c \in \text{dom}(r(y))$, i.e. $y \in C_c$. Since C_c is a maximal chain, there exists a lower neighbor (with respect to the partial order $E^\star$) x of y which belongs to the chain C_c. Hence $(x, y) \in E$ and $c \in \mathrm{dom}(r(x))$. Furthermore, x is not maximal in $(V, E^\star)$. Let $y' \in C_c$ be minimal with $(x, y') \in E$. Then $\lambda(y') = r(x)(c)$. Since $(x, y) \in E^+$ and $y \in C_c$, we obtain $xE^+y'E^\star y$ which ensures $y' = y$. Hence $\lambda(y) = r(x)(c)$.

Now we prove the second statement of the lemma. Let $f : \Sigma \to \Sigma$ be some mapping. Furthermore let $t = (V, E, \lambda)$ be a weak Σ-dag and let $r : V \to Q$ be a mapping that satisfies the run condition of $\mathcal{A}(f)$ for any node $x \in V$ relative to t. We will prove that $C_a := \{x \in V \mid a \in \mathrm{dom}(r(x))\}$ is a chain. Since by the definition of the transition functions $\delta_{a,J}$ we have $\lambda(x) \in \mathrm{dom}(r(x))$ for any $x \in V$, this will imply $\lambda^{-1}(a) \subseteq C_a$ and therefore that $\lambda^{-1}(a)$ is linearly ordered.

Now let $x, y \in C_c$. Since r satisfies the run condition of $\mathcal{A}(f)$, there exist $x_0, x_1, \ldots, x_n \in V$ such that $x_0 \in \min(t)$, $x_n = x$, $(x_i, x_{i+1}) \in E$ for $0 \leq i < n$, and $c \in \mathrm{dom}(r(x_i))$ and $r(x_i)(c) = \lambda(x_{i+1})$ for $0 \leq i < n$.

Similarly, we find nodes $y_0, y_1, \ldots, y_m \in V$ such that $y_0 \in \min(t)$, $y_m = y$, $(y_i, y_{i+1}) \in E$ for $0 \leq i < m$, and $c \in \mathrm{dom}(r(y_i))$ and $r(y_i)(c) = \lambda(y_{i+1})$ for $0 \leq i < m$. Without loss of generality, we may assume $n \leq m$.

Since $c \in \mathrm{dom}(r(x_0))$ and $\mathrm{R}(x_0) = \emptyset$, we obtain $c \in f^{-1}(\lambda(x_0))$ since the run condition is satisfied at the node x_0. Hence $f(c) = \lambda(x_0)$ and similarly $f(c) = \lambda(y_0)$. Since the minimal nodes of the weak Σ-dag t carry different labels, this implies $x_0 = y_0$. By induction, let $0 \leq i < n$ with $x_i = y_i$. Then $(x_i, x_{i+1}) \in E$, $(y_i, y_{i+1}) \in E$ and $\lambda(x_{i+1}) = r(x_i)(c) = r(y_i)(c) = \lambda(y_{i+1})$. Since t is a weak Σ-dag, this implies $x_{i+1} = y_{i+1}$. Thus, we get $x = x_n = y_n E^\star y$ as required. ☐

Let $\mathcal{A}$ be some Σ-ACM and $\mathcal{S}(\mathcal{A}) = (S, \to, \preceq)$ be the associated WSTS. A state $(w_a, \mathrm{pos}_a^w)_{a\in\Sigma} \in S$ is a *depth-1-state* if

1. $|w_a| \leq 1$ for $a \in \Sigma$,
2. $\mathrm{pos}_a^w(b) = 0$ for $a, b \in \Sigma$, and
3. $w_a(\min \mathrm{dom}(w_a)) \in \delta_{a,\emptyset}$ for $a \in \Sigma$ with $w_a \neq \emptyset$.

Let $(w_a, \mathrm{pos}_a^w)_{a\in\Sigma}$ be some depth-1-state. Let $V = \{a \in \Sigma \mid w_a \neq \emptyset\}$ and $E = \emptyset$. Finally, let $\lambda = \mathrm{id}_V$. Since $w_a(\min \mathrm{dom}(w_a)) \in \delta_{a,\emptyset}$ for $a \in V$, the mapping $a \mapsto w_a(\min \mathrm{dom}(w_a))$ is a run of $\mathcal{A}$ on the Σ-dag $t = (V, E, \lambda)$. Furthermore, the Σ-dag t is (considered as a partial order) an antichain since $E = \emptyset$. If conversely t is an antichain and r is a run of $\mathcal{A}$ on t, then state(r) is a depth-1-state.

Now let $\mathcal{A}_i = ((Q_a^i, \sqsubseteq_a^i)_{a\in\Sigma}, (\delta_{a,J}^i)_{a\in\Sigma, J\subseteq\Sigma}, F^i)$ for $i = 1, 2$ be two Σ-ACMs. Then the direct product $\mathcal{A}_1 \times \mathcal{A}_2 = ((Q_a, \sqsubseteq_a)_{a\in\Sigma}, (\delta_{a,J})_{a\in\Sigma,J\subseteq\Sigma}, F)$ has the following obvious definition:

$$\begin{aligned} Q_a &:= Q_a^1 \times Q_a^2, \\ \sqsubseteq_a &:= \sqsubseteq_a^1 \times \sqsubseteq_a^2 \\ \delta_{(a,J)}((p_b^1, p_b^2)_{b\in J}) &:= \delta^1_{(a,J)}((p_b^1)_{b\in J}) \times \delta^2_{(a,J)}((p_b^2)_{b\in J}), \text{ and} \\ F &:= \{(q_a^1, q_a^2)_{a\in J} \mid (q_a^i)_{a\in J} \in F^i \text{ for } i = 1, 2\}. \end{aligned}$$

It is easily seen that the direct product of monotone and effective ACMs is monotone and effective, again. Furthermore, this direct product accepts the intersection of the two languages, i.e. $L(\mathcal{A}_1 \times \mathcal{A}_2) = L(\mathcal{A}_1) \cap L(\mathcal{A}_2)$. Hence, to decide whether $L(\mathcal{A})$ is empty, by the first statement of the preceding lemma, it suffices to decide whether $L(\mathcal{A} \times \mathcal{A}(f))$ is empty for each function $f : \Sigma \to \Sigma$. This is essentially the reason why we now start to consider these direct products.

Lemma 3.3.2 *Let $\mathcal{A}'$ be a Σ-ACM and $f : \Sigma \to \Sigma$. Let $\mathcal{A} = \mathcal{A}' \times \mathcal{A}(f)$ and let $\mathcal{S}(\mathcal{A}) = (S, \to, \preceq)$ be the WSTS associated with $\mathcal{A}$. Let $(w_a, \mathrm{pos}_a^w)_{a\in\Sigma}$ be a state of $(S, \to)$. Then the following are equivalent:*

(1) There exist a Σ-dag t and a run r of the Σ-ACM $\mathcal{A}$ on t such that $\mathrm{state}(r) = (w_a, \mathrm{pos}_a^w)_{a\in\Sigma}$.

(2) The state $(w_a, \mathrm{pos}_a^w)_{a\in\Sigma}$ is reachable from some depth-1-state in the transition system $(S, \to)$.

Proof. Throughout this proof, let $\mathcal{A} = ((Q_a, \sqsubseteq_a)_{a\in\Sigma}, (\delta_{a,J})_{a\in\Sigma, J\subseteq\Sigma}, F)$.

(1) $\to$ (2): Let $t = (V, E, \lambda)$ be a Σ-dag and $r : V \to Q$ a run of $\mathcal{A}$ on t. Recall that we assume $V \subseteq \mathbb{N}^+$ with $x < y$ whenever $(x, y) \in E$. We can in addition require that $x \in \min(V, E^\star)$ and $y \in V \setminus \min(V, E^\star)$ imply $x < y$. Since the linear order $\leq$ of natural numbers extends the partial order $E^\star$ on V, we can enumerate V such that $V = \{x_1, x_2, \ldots, x_n\}$ with $x_i < x_{i+1}$. Furthermore, there is $k \in \mathbb{N}^+$ such that $\min(V, E^\star) = \{x_1, x_2, \ldots, x_k\}$ by our additional requirement. For $i = k, k+1, k+2, \ldots, n$, let $V_i := \{x_1, x_2, \ldots, x_i\}$, $t_i := (V_i, E, \lambda)$ and $r_i : V_i \to Q$ be the restriction of r to V_i. Then, for all suitable i, r_i is a run of the Σ-ACM $\mathcal{A}$ on the Σ-dag t_i. Furthermore, $V_k = \{x_1, x_2, \ldots, x_k\}$ is the set of minimal nodes of t with respect to $E^\star$ and t_k is the restriction of t to its minimal nodes. Hence $\mathrm{state}(r_k)$ is a depth-1-state. It remains to prove $\mathrm{state}(r_i) \to \mathrm{state}(r_{i+1})$ for $k \leq i < n$ to obtain the desired result by induction. Let $(v_a, \mathrm{pos}_a^v)_{a\in\Sigma} = \mathrm{state}(r_i)$ and $(w_a, \mathrm{pos}_a^w)_{a\in\Sigma} = \mathrm{state}(r_{i+1})$. Furthermore, let $a = \lambda(x_{i+1})$, $J = \mathrm{R}(x_{i+1})$, $p_b = r\partial_b(x_{i+1})$ for $b \in J$, and $q = r(x_{i+1})$.

We show that (i)-(iv) hold for these elements: Since $i+1 > k$, the node x_{i+1} is not minimal in t. Hence it is the target of some edge from E, i.e. $J \neq \emptyset$. Since r_{i+1} is a run on t_{i+1}, we get $q \in \delta_{a,J}((p_b)_{b\in J})$ and therefore (i). Since $V_{i+1} \setminus V_i = \{x_{i+1}\}$ and $r_i = r_{i+1} \restriction V_i$, we get $r_i \restriction \lambda^{-1}(c) = r_{i+1} \restriction \lambda^{-1}(c)$ for $c \neq a$. Hence $w_c = v_c$

for $c \neq a$. Furthermore, $w_a = r_{i+1} \restriction \lambda^{-1}(a) = (r_i \restriction \lambda^{-1}(a))\, r_{i+1}(x_{i+1}) = v_a q$. Thus, we showed (ii). Note that the only edges in t_{i+1} that do not belong to t_i are of the form (x, x_{i+1}) with $\lambda(x) \in \mathrm{R}(x_{i+1}) = J$, i.e. their source is labeled by an element of J while the target is labeled by a. Hence, for $b, c \in \Sigma$ with $c \in J$ or $a \neq b$, we have $\mathrm{pos}_c^v(b) = \mathrm{pos}_c^w(b)$, i.e. (iii) holds.

To show (iv), let $c \in J$. Then $\mathrm{pos}_c^w(a) = \partial_c(x_{i+1})$ since there is an edge (x, x_{i+1}) in t_{i+1} with $\lambda(x) = c$. Let $y, z \in V_i$ such that $\lambda(y) = a$ and $\lambda(z) = b$ with $(z, y) \in E$. Then $y < x_{i+1}$ and therefore $z < \partial_c(x_{i+1})$ by the second requirement on Σ-dags. Hence $\mathrm{pos}_c^v(a) < \mathrm{pos}_c^w(a)$. Since $\partial_c(x_{i+1}) \in V_i$, we also get $\mathrm{pos}_c^w(a) \in \lambda^{-1}(c) \cap V_i = \mathrm{dom}\, v_c$. By the very definition of p_c, we have $w_c \circ \mathrm{pos}_c^w(a) = r \circ \mathrm{pos}_c^w(a) = r\partial_c(x_{i+1}) = p_c$, i.e. (iv) holds. Thus we showed $\mathrm{state}(r_i) \to \mathrm{state}(r_{i+1})$ and therefore the implication (1) $\to$ (2).

(2) $\to$ (1): When we defined the concept of a depth-1-state, we showed that they are of the form $\mathrm{state}(r)$ for some run r of $\mathcal{A}$. Hence the implication (2) $\to$ (1) holds for depth-1-states and it remains to show that, given a run r, any successor of $\mathrm{state}(r)$ in $(S, \to)$ is of the form $\mathrm{state}(r')$ for some run r' of the Σ-ACM $\mathcal{A}$. So let $t = (V, E, \lambda)$ be a Σ-dag and let $r : V \to Q$ be a run of $\mathcal{A}$ on t. Furthermore, let $\mathrm{state}(r) = (v_a, \mathrm{pos}_a^v)_{a\in\Sigma} \to (w_a, \mathrm{pos}_a^w)_{a\in\Sigma}$. Then there exist $a \in \Sigma$, $\emptyset \neq J \subseteq \Sigma$, $p_b \in Q_b$ for $b \in J$ and $q \in Q_a$ such that (i)-(iv) hold. Define $V' := V \,\dot\cup\, \{z\}$ and let $\lambda' := \lambda \cup \{(z, a)\}$. The set of edges E' will consist of all edges from E and some edges of the form (x, z) with $x \in V$. According to the definition of a run, we should have additional edges with $\lambda(x) \in J$ only and, conversely, for any $c \in J$ there has to be a new edge (x, z) with $\lambda(x) = c$. Furthermore, the state at the source of this new edge should equal p_c. By (iv), $\mathrm{pos}_c^w(a) \in \mathrm{dom}\, v_c = \lambda^{-1}(c)$. Hence $\mathrm{pos}_c^w(a)$ belongs to V and is labeled by c. Now we define

$$E' := E \,\dot\cup\, \{(\mathrm{pos}_c^w(a), z) \mid c \in J\}.$$

Then (V', E') is a dag since the only new edges have a common target z. We show that $t' = (V', E', \lambda')$ is a weak Σ-dag: Since $J \neq \emptyset$, there is an edge whose target is z, i.e. z is not minimal in $(V', (E')^\star)$. In other words $\min(t) = \min(t')$. Since t is a Σ-dag, this implies that the minimal nodes of t' carry mutually different labels as required by the first axiom for weak Σ-dags. Now let $(x, y), (x', y') \in E'$ with $\lambda(x) = \lambda(x')$ and $\lambda(y) = \lambda(y')$. We have to show $x = x' \iff y = y'$. Since t is a Σ-dag, this holds if $(x, y), (x', y') \in E$. So assume $(x, y) \in E' \setminus E$, i.e. $y = z$ and $x = \mathrm{pos}_c^w(a)$ for some $c \in J$. If $x = x'$, we get $x' = x = \mathrm{pos}_c^w(a) > \mathrm{pos}_c^v(a)$ by (iv) since $c \in J$. Thus, $x' > \sup\{\overline{x} \in \lambda^{-1}(c) \mid \exists \overline{y} \in \lambda^{-1}(a) : (\overline{x}, \overline{y}) \in E\}$. Hence $(x', y') \notin E$ and therefore $y' = z = y$. Conversely assume $y = y'$. Then $x = \mathrm{pos}_{\lambda(x)}^w(a) = \mathrm{pos}_{\lambda(x')}^w(a) = x'$. Thus, t' is indeed a weak Σ-dag.

Now let $r' := r \,\dot\cup\, \{(z, q)\}$. For $x \in V$, this mapping satisfies the run condition of $\mathcal{A}$ relative to t and therefore relative to t'. Since the edges in t' with target z are of the form $(\mathrm{pos}_c^w(a), z)$ with $c \in J$, we have $\mathrm{R}(z) = J$ and $\partial_c(z) = \mathrm{pos}_c^w(a)$ for $c \in J$. Hence, by (iv), $r\partial_c(z) = p_c$ for $c \in J$. Since $q \in \delta_{a,J}((p_c)_{c\in J})$, the mapping r' satisfies the run condition at z relative to t', too. Recall that $\mathcal{A}$ is the direct product of $\mathcal{A}'$ and $\mathcal{A}(f)$. Hence $\pi_2 \circ r'$ satisfies the run condition of $\mathcal{A}(f)$ at any node $x \in V'$ relative to the weak Σ-dag t'. Hence, by Lemma 3.3.1 (2), the set $(\lambda')^{-1}(b)$ is a chain w.r.t. $(E')^\star$ for any $b \in \Sigma$. To show that t' is a Σ-dag, it remains to prove the second condition, i.e. that for any $(x,y),(x',y') \in E'$ with $\lambda(x) = \lambda(x')$ and $\lambda(y) = \lambda(y')$ we have

$$(x,x') \in (E')^\star \iff (y,y') \in (E')^\star.$$

Since t is a Σ-dag, this equivalence holds if $(x,y),(x',y') \in E$.

So assume $(x',y') \in E' \setminus E$. Then $y' = z$. Since $\lambda(y) = \lambda(y')$, the nodes y and $y' = z$ are ordered w.r.t. $(E')^\star$. Since $z = y'$ is maximal in t' w.r.t. $(E')^\star$, this implies $(y,y') \in (E')^\star$. We show $(x,x') \in (E')^\star$: If $(x,y) \notin E$, we are done since then $y = y'$ and therefore $x = x'$. So assume $(x,y) \in E$. Since $(x',y') \in E' \setminus E$, there exists $c \in J$ with $x' = \mathrm{pos}_c^w(a)$. Hence $x' > \mathrm{pos}_c^v(a)$ by (iv). But $\mathrm{pos}_c^v(a) = \sup\{\overline{x} \in \lambda^{-1}(c) \mid \exists \overline{y} \in \lambda^{-1}(a) : (\overline{x},\overline{y}) \in E\}$ and the node x is contained in this set. Hence, indeed $x' > x$ w.r.t. the linaer order on the natural numbers. Since x' and x carry the same label, they are comparable w.r.t. $E^\star$. Hence $(x,x') \in E^\star$. Thus, we showed the required equivalence in case $(x',y') \in E' \setminus E$.

Now assume $(x,y) \in E' \setminus E$ and therefore $y = z$. First, let $(x,x') \in (E')^\star$ and therefore $x \leq x'$. Since $x = \mathrm{pos}_c^w(a)$ for some $c \in J$, we obtain $x \geq x'$ as above. Hence $x = x'$ and, since t' is a weak Σ-dag, $y = y'$. Thus, we showed $(y,y') \in (E')^\star$. Now assume $(y,y') \in (E')^\star$. Since $(x,y) \in E' \setminus E$, we obtain similarly to above, $(y',y) \in (E')^\star$, i.e. $y = y'$. Since t' is a weak Σ-dag, this implies $x = x'$.

Thus, t' is indeed a Σ-dag. Hence r' is a run of the Σ-ACM $\mathcal{A}$ on the Σ-dag t'. It is an easy exercise to show $\mathrm{state}(r') = (w_a, \mathrm{pos}_a^w)_{a\in\Sigma}$ proving the implication (2) $\to$ (1). □

By Theorem 3.2.4 and Lemma 3.2.8, we can apply Theorem 3.2.2 to $(S, \to, \preceq)$, i.e. there is an algorithm that, given a monotone and effective Σ-ACM $\mathcal{A}$ and a state $(w_c, \mathrm{pos}_c^w)_{c\in\Sigma} \in S$, decides whether $(w_c, \mathrm{pos}_c^w)_{c\in\Sigma}$ is dominated by some reachable state in the WSTS $(S, \to, \preceq)$. It remains to transfer this decidability to the question whether the language accepted by $\mathcal{A}$ is empty:

Let $\mathcal{A}' = ((Q'_a, \sqsubseteq'_a)_{a\in\Sigma}, (\delta'_{a,J})_{a\in\Sigma, J\subseteq\Sigma}, F')$ be a Σ-ACM and $f : \Sigma \times \Sigma$. Define $\mathcal{A} = \mathcal{A}' \times \mathcal{A}(f)$ and let $\mathcal{A} = ((Q_a, \sqsubseteq_a)_{a\in\Sigma}, (\delta_{a,J})_{a\in\Sigma, J\subseteq\Sigma}, F)$. Furthermore, let B_c be a finite basis of the set of local states Q_c of the product automaton $\mathcal{A} = \mathcal{A}' \times \mathcal{A}(f)$ for $c \in \Sigma$. Now let $J \subseteq \Sigma$ and let q_c be some local state from the product automaton for $c \in J$. We define $\mathrm{States}((q_c)_{c\in J})$ to consist of all states $(w_c, \mathrm{pos}_c^w)_{c\in\Sigma}$ from $\mathcal{S}(\mathcal{A})$ such that for all $c \in \Sigma$:

$$|w_c| \leq |\Sigma|, \ (w_c = \varepsilon \iff c \notin J), \text{ and } w_c \in B_c^\star q_c \text{ for } c \in J.$$

Note that due to the restrictions $|w_c| \leq |\Sigma|$ and $w_c \in B_c^\star q_c \cup \{\varepsilon\}$, the set $\mathrm{States}((q_c)_{c\in J})$ is finite. Since, in addition, the set F of accepting states of $\mathcal{A}' \times \mathcal{A}(f)$ is finite, we even have that $\bigcup_{\overline{q}\in F} \mathrm{States}(\overline{q})$ is finite. The following lemma states that $L(\mathcal{A}' \times \mathcal{A}(f))$ is not empty iff some state of this finite set $\bigcup_{\overline{q}\in F} \mathrm{States}(\overline{q})$ is dominated by a state in $\mathcal{S}(\mathcal{A}' \times \mathcal{A}(f))$ that is reachable from a depth-1-state.

Lemma 3.3.3 *Let $\mathcal{A}'$ be a Σ-ACM, $f : \Sigma \to \Sigma$ and $\mathcal{A} = \mathcal{A}' \times \mathcal{A}(f)$. Furthermore, let $\mathcal{S}(\mathcal{A}) = (S, \to, \preceq)$. Then the following are equivalent:*

1. *$\mathcal{A}$ accepts some Σ-dag, i.e. $L(\mathcal{A}) \cap L(\mathcal{A}(f)) \neq \emptyset$.*
2. *There exist an accepting state $(q_a)_{a\in J}$ of $\mathcal{A}$, a depth-1-state $(v'_a, \mathrm{pos}_a^{v'})_{a\in\Sigma}$ from S, a state $(w_a, \mathrm{pos}_a^w)_{a\in\Sigma} \in \mathrm{States}((q_a)_{a\in J})$ and a state $(w'_a, \mathrm{pos}_a^{w'})_{a\in\Sigma}$ in S such that $(v'_a, \mathrm{pos}_a^{v'})_{a\in\Sigma} \to^\star (w'_a, \mathrm{pos}_a^{w'})_{a\in\Sigma} \succeq (w_a, \mathrm{pos}_a^w)_{a\in\Sigma}$.*

Proof. Let $t = (V, E, \lambda) \in L(\mathcal{A})$. Then there exists a successful run r of $\mathcal{A}$ on t. By Lemma 3.3.2, $\mathrm{state}(r) = (w'_a, \mathrm{pos}_a^{w'})_{a\in\Sigma}$ is reachable from some depth-1-state $(v'_a, \mathrm{pos}_a^{v'})_{a\in\Sigma}$. Since r is successful, there is $(q_c)_{c\in\lambda(V)} \in F$ such that $w'_c(\sup\mathrm{dom}\, w'_c) = r(\sup \lambda^{-1}(c)) \sqsupseteq_c q_c$ for $c \in \lambda(V) =: J$.

For $a \in \Sigma$, define a word $w_a \in B_a^\star q_a \cup \{\varepsilon\}$ as follows: Let $\mathrm{dom}\, w_a := (\mathrm{im}\,\mathrm{pos}_a^{w'} \cup \{\sup\mathrm{dom}\, w'_a\}) \setminus \{0\}$. If $\mathrm{dom}\, w_a \neq \emptyset$, let $w_a(\max\mathrm{dom}\, w_a) := q_a$. For $1 \leq i < \max\mathrm{dom}\, w'_a$ choose $w_a(i) \in B_a$ with $w_a(i) \sqsubseteq_a w'_a(i)$. Furthermore, let $\mathrm{pos}_a^w = \mathrm{pos}_a^{w'}$. Then $(w_a, \mathrm{pos}_a^w)_{a\in\Sigma} \preceq (w'_a, \mathrm{pos}_a^{w'})_{a\in\Sigma}$ witnessed by the identity mapping from $\mathrm{dom}\, w_a \cup \{0\}$ to $\mathrm{dom}\, w'_a \cup \{0\}$. By the very construction it can easily be checked that $(w_c, \mathrm{pos}_c^w)_{c\in\Sigma} \in \mathrm{States}((q_c)_{c\in J}) \subseteq \bigcup_{\overline{q}\in F} \mathrm{States}(\overline{q})$.

Conversely, let $(q_a)_{a\in J} \in F$ be an accepting state of $\mathcal{A}$. Furthermore, let $(w_a, \mathrm{pos}_a^w)_{a\in\Sigma} \in \mathrm{States}((q_a)_{a\in J})$ and suppose $(v'_a, \mathrm{pos}_a^{v'})_{a\in\Sigma} \to^\star (w'_a, \mathrm{pos}_a^{w'})_{a\in\Sigma} \succeq (w_a, \mathrm{pos}_a^w)_{a\in\Sigma}$ for some depth-1-state $(v'_a, \mathrm{pos}_a^{v'})_{a\in\Sigma}$. We assume furthermore that $(w'_a, \mathrm{pos}_a^{w'})_{a\in\Sigma} \succeq (w_a, \mathrm{pos}_a^w)_{a\in\Sigma}$ is witnessed by the embeddings $\eta_a : w_a \hookrightarrow w'_a$. By Lemma 3.3.2, there exists a Σ-dag $t = (V, E, \lambda)$ and a run r of $\mathcal{A}$ on t such that $(w'_a, \mathrm{pos}_a^{w'})_{a\in\Sigma} = \mathrm{state}(r)$. Since w'_a is the empty word iff w_a is empty, we obtain $\lambda(V) = \{a \in \Sigma \mid w'_a \neq \varepsilon\} = \{a \in \Sigma \mid w_a \neq \varepsilon\}$. For $a \in \Sigma$ with $w_a \neq \varepsilon$, i.e. for $a \in \lambda(V)$, we have $w'_a(\sup\mathrm{dom}\, w'_a) \sqsupseteq_a w_a(\sup\mathrm{dom}\, w_a) = q_a$. Hence the run r is successful. □

Summarizing the results of this section, finally we show that the emptiness of effective and monotone Σ-ACMs is uniformly decidable:

Theorem 3.3.4 *There exists an algorithm that solves the following decision problem:*
input: *1. an alphabet Σ,*
2. a basis algorithm of an effective and monotone Σ-ACM $\mathcal{A}'$,
3. the set of final states F' of $\mathcal{A}'$,
4. a finite basis of $(Q_c, \sqsubseteq_c)$, and an algorithm to decide $\sqsubseteq_c$ for $c \in \Sigma$.
output: *Is $L(\mathcal{A}')$ empty?*

Proof. We may assume that there is $a \in \Sigma$ such that $\delta_{a,\emptyset} = \{\bot\}$ and $\delta_{c,\emptyset} = \emptyset$ for $c \neq a$. Then there is only one depth-1-state $(v_c, \text{pos}_c^v)_{c\in\Sigma}$.

By Lemma 3.3.1 (1), it holds $L(\mathcal{A}') = \bigcup_{f\in\Sigma^\Sigma} L(\mathcal{A}' \times \mathcal{A}(f))$. Hence it suffices to decide the emptiness of $L(\mathcal{A}' \times \mathcal{A}(f))$ for $f : \Sigma \to \Sigma$. So let $\mathcal{A} = \mathcal{A}' \times \mathcal{A}(f)$. Note that this Σ-ACM if monotone and effective, that we have access to a basis algorithm for this ACM, that we know a finite basis for the sets of local states and that we can decide the wqos of local states for any $c \in \Sigma$. Now let $\mathcal{S}(\mathcal{A}) = (S, \to, \preceq)$ be the associated transition system. By Theorem 3.2.4, it is a WSTS. It is clear that $\preceq$ is decidable using the algorithms that decide the wqos of local states in $\mathcal{A}$. By Lemma 3.2.8, from a state $(w_a, \text{pos}_a^w)_{a\in\Sigma} \in S$, a finite basis of the set $\text{Pred}(\uparrow(w_a, \text{pos}_a^w)_{a\in\Sigma})$ can be computed effectively. Hence, by Theorem 3.2.2 the set of states that are dominated by a state reachable from $(v_a, \text{pos}_a^v)_{a\in\Sigma}$ is recursive. Since $\bigcup_{\overline{q}\in F} \text{States}(\overline{q})$ is finite, the result follows from Lemma 3.3.3. □

A consequence of Theorem 3.3.4 is that for any monotone and effective Σ-ACM $\mathcal{A}$ the membership in $L(\mathcal{A})$ is decidable:

Corollary 3.3.5 *Let $\mathcal{A}$ be a monotone and effective Σ-ACM. Then the set $L(\mathcal{A})$ is recursive.*

Proof. Let $t \in \mathbb{D}$ be some Σ-dag. Then one can easily construct a Σ-ACA $\mathcal{A}_t$ with $L(\mathcal{A}_t) = \{t\}$. Hence $L(\mathcal{A} \times \mathcal{A}_t)$ is empty iff $t \notin L(\mathcal{A})$. Since the emptiness of $L(\mathcal{A} \times \mathcal{A}_t)$ is decidable, so is the question "$t \in L(\mathcal{A})$?". □

Unfortunately, Theorem 3.3.4 keeps the promise made by the title of this section only partially since we have to impose additional requirements on the Σ-ACMs:

- Of course, one cannot expect that the emptiness for arbitrary Σ-ACMs is decidable. There is even a formal reason: In general, a Σ-ACM is an infinite object that has to be given in some finite form. Hence some effectiveness requirement is necessary.

- On the other hand, the monotonicity originates only in our proof using well structured transition systems. These transition systems clearly require some monotonicity but it is not clear whether this is really needed for the result on asynchronous cellular machines.

Recall that by Example 2.1.3 the set of Hasse-diagrams of all pomsets without autoconcurrency over an alphabet Σ can be accepted by some Σ-ACM. One can check that the ACM we gave is not monotone. Unfortunately, we were not able to construct a monotone Σ-ACM accepting all Hasse-diagrams nor did we succeed in showing that such a Σ-ACM does not exist. If we were able to accept all Hasse-diagrams by a monotone and effective Σ-ACM, the question "Is $L(\mathcal{A}) \cap \mathrm{Ha}$ empty?" would be decidable for monotone and effective ACMs $\mathcal{A}$.

An asynchronous cellular *automaton* over Σ is a Σ-ACM where the sets of local states Q_c are finite for $c \in \Sigma$. Hence the identity relations on Q_c for $c \in \Sigma$ are well quasi orders. Thus, the set of Σ-ACAs $\mathcal{A}$ with $L(\mathcal{A}) \neq \emptyset$ is recursive. It is easily seen that a *deterministic* ACA can effectively be complemented. Similarly, one can effectively construct a deterministic ACA that accepts the intersection of two languages accepted by deterministic ACAs. Hence, as a consequence of the theorem above, the equivalence of deterministic Σ-ACAs is decidable. The following chapter shows that this is not the case for nondeterministic Σ-ACAs.

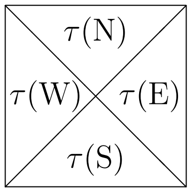
Chapter 4

The undecidability results

The result of the preceding chapter shows that one can automatically check whether a property of Σ-dags described by a Σ-ACM is contradictory. Another natural question is whether two properties are equivalent, i.e. whether two Σ-ACMs accept the same language. Since there is a Σ-ACM that accepts all Σ-dags, a special case of this equivalence problem is to ask whether a given Σ-ACM accepts all Σ-dags. This latter question, called universality, essentially asks whether the described property is always satisfied.

The corresponding question for finite sequential automata has a positive answer which is a consequence of the decidability of the emptiness: If one wants to know whether a sequential automaton accepts all words, one constructs the complementary automaton and checks whether its language is empty. Thus, the crucial point for sequential automata is that they can effectively be complemented. But Example 2.1.6 shows that the set of acceptable Σ-dag-languages is not closed under complementation. Therefore, Theorem 3.3.4 does not imply that the universality of an Σ-ACM is decidable. On the contrary, we show that the universality is undecidable even for Σ-ACAs. This implies that the equivalence of two Σ-ACAs, the complementability and the determinisability of a Σ-ACA are undecidable, too. This result was announced in [Kus98] for Hasse-diagrams together with the sketch of a proof. This original proof idea used the undecidability of the Halting Problem. Differently, our proof here is based on the undecidability of the Tiling Problem. This change, as well as the formulation and proof of Lemmas 4.1.4 and 4.1.5 were obtained in collaboration with Paul Gastin. *Throughout this section, let $\Sigma = \{a, b\}$ if not stated otherwise.*

Let $\mathfrak{C}$ be a finite set of colors with white $\in \mathfrak{C}$. A mapping $\tau : \{\mathrm{W}, \mathrm{N}, \mathrm{E}, \mathrm{S}\} \to \mathfrak{C}$ is called a *tile*. Since the elements W, N etc. stand for the cardinal points, a tile can be visualized as follows:



Now let $\mathcal{T}$ be a set of tiles and $k, \ell \in \mathbb{N}^+$. A mapping $T : [k] \times [\ell] \to \mathcal{T}$ is a *tiling of the grid* $[k] \times [\ell]$ provided for any $(i, j) \in [k] \times [\ell]$ we have

1. $f(i,j)(\mathrm{W}) = \begin{cases} \text{white} & \text{if } i = 1 \\ f(i-1,j)(\mathrm{E}) & \text{otherwise} \end{cases}$

2. $f(i,j)(\mathrm{S}) = \begin{cases} \text{white} & \text{if } j = 1 \\ f(i,j-1)(\mathrm{N}) & \text{otherwise} \end{cases}$

Note that then $f(i,j)(\mathrm{E}) = f(i+1,j)(\mathrm{W})$ for $i < k$, and similarly $f(i,j)(\mathrm{N}) = f(i,j+1)(\mathrm{S})$ for $j < \ell$. An *infinite tiling* is a mapping $f : \mathbb{N}^+ \times \mathbb{N}^+ \to \mathcal{T}$ such that for any $k \in \mathbb{N}^+$ the restriction of f to $[k] \times [k]$ is a tiling. It is known that for a set $\mathcal{T}$ of tiles the existence of an infinite tiling is undecidable [Ber66].

A set of grids is *unbounded* if, for any $k, \ell \in \mathbb{N}^+$, it contains a grid $[k'] \times [\ell']$ with $k \leq k'$ and $\ell \leq \ell'$.

Lemma 4.1.1 *Let $\mathcal{T}$ be a set of tiles for the finite set of colors $\mathfrak{C}$. Then $\mathcal{T}$ allows an infinite tiling iff the set of grids that allow a tiling is unbounded.*

Proof. Let $f : \mathbb{N}^+ \times \mathbb{N}^+ \to \mathcal{T}$ be an infinite tiling. Then, for $k, \ell \in \mathbb{N}^+$, let $k' = \max(k, \ell)$. By definition, the restriction of f to $[k'] \times [k']$ is a tiling. Thus, the set of tilable grids is unbounded.

For the converse let T denote the set of all tilings of squares $[k] \times [k]$ for some $k \in \mathbb{N}^+$ ordered by inclusion. Then this is a tree. Any node of the tree has finitely many upper neighbors. Since the set of all tilable grids is unbounded, all squares can be tiled. Hence T is infinite. By König's Lemma, it has an infinite branch $(f_i)_{i \in \mathbb{N}^+}$. Then $f = \bigcup_{i \in \mathbb{N}^+} f_i$ is an infinite tiling. □

To encode the tiling problem into our setting of Σ-dags, we will consider the (k, ℓ)-grids $[k] \times [\ell]$ with $k, \ell \in \mathbb{N}^+$ and the edge relation

$$E' = \begin{array}{l} \{((i,j),(i,j+1)) \mid 1 \leq i \leq k, 1 \leq j < \ell\} \cup \\ \{((i,j),(i+1,j)) \mid 1 \leq i < k, 1 \leq j \leq \ell\}. \end{array}$$

Let $\leq$ be the reflexive and transitive closure of E'. Then the partial orders $([k] \times [\ell], \leq)$ contain antichains of size $\min(k, \ell)$. Hence they do not fit into our setting of Σ-dags where the size of antichains is restricted to n. Therefore, we define

$$((i,j),(i',j')) \in E \text{ iff } ((i,j),(i',j')) \in E' \text{ or } j+2=j',\ i=\ell \text{ and } i'=1$$

(see Figure 4.1). The Σ-dag $([k]\times[\ell],E,\lambda)$ is *the folding of the grid* $[k]\times[\ell]$ or a *folded grid*. Let $\preceq$ denote the transitive and reflexive closure of E. Then the partially ordered set $([k]\times[\ell],\preceq)$ contains antichains of size 2, only, and E is the covering relation of $\preceq$. Furthermore, the chains $\{(i,2j+1) \mid i\in[k], 2j+1\in[\ell]\}$ and $\{(i,2j) \mid i\in[k], 2j\in[\ell]\}$ form a partition of the partial order $([k]\times[\ell],\preceq)$. We label the elements $(i,2j+1)$ of the first chain by a. Similarly, the elements $(i,2j)$ of the second chain are labeled by b. Thus, two elements get the same label iff their second components have the same parity. Note that in the folded grid all vertices except $(1,1)$ have a lower neighbor labeled by a, and that all vertices (i,j) with $j>1$ have a lower neighbor labeled by b. Hence for $1\le i\le k$ and $1\le j\le \ell$ it holds

$$\mathrm{R}(i,j)=\begin{cases}\emptyset & \text{for } i=j=1\\ \{a\} & \text{for } 1<i\le k, j=1 \text{ or } (i,j)=(1,2)\\ \{a,b\} & \text{otherwise.}\end{cases}$$

Furthermore, we have

$$\partial_a(i,j)=\begin{cases}\text{undefined} & \text{for } i=j=1\\ (i-1,j) & \text{for } 1<i\le k, j \text{ odd}\\ (k,j-2) & \text{for } i=1, 1<j\le\ell \text{ odd}\\ (i,j-1) & \text{for } j \text{ even, and}\end{cases}$$

$$\partial_b(i,j)=\begin{cases}\text{undefined} & \text{for } j=1 \text{ or } (i,j)=(1,2)\\ (i-1,j) & \text{for } 1<i\le k, j \text{ even}\\ (k,j-2) & \text{for } i=1, 1<j\le\ell \text{ even}\\ (i,j-1) & \text{for } j>2 \text{ odd.}\end{cases}$$

Let $\mathbb{G}$ comprise the set of all folded grids, i.e. we define $\mathbb{G}\subseteq\mathbb{D}$ by

$$\mathbb{G}=\{([k]\times[\ell],E,\lambda) \mid k,\ell\in\mathbb{N}^+\}.$$

Next, from a set of tiles $\mathcal{T}$, we construct a Σ-ACA $\mathcal{A}_{\mathcal{T}}$ that recognizes among all folded grids those that allow a tiling. This automaton guesses a tile for any vertex and checks that it is a tiling. So let $\mathfrak{C}$ be a finite set of colors and $\mathcal{T}$ a set of tiles. Then the ACA $\mathcal{A}_{\mathcal{T}}$ is given by $Q_a=Q_b=\mathcal{T}\times\{0,1\}$ and

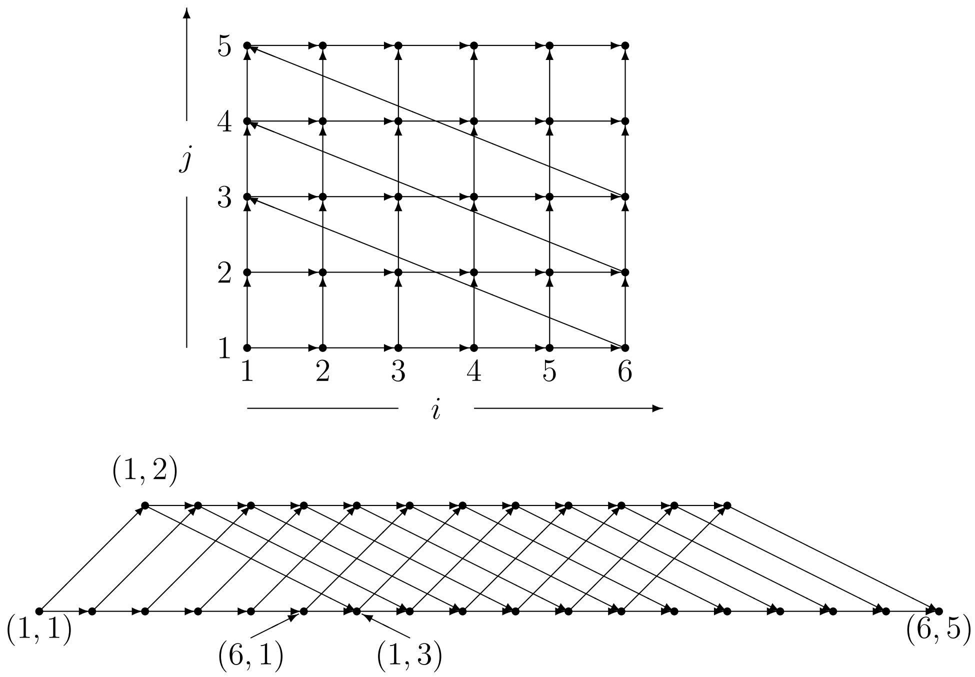


Figure 4.1: The folded grid $([6] \times [5], E)$

$$\begin{aligned}
\delta_{a,\emptyset} &= \{g \in \mathcal{T} \mid g(\mathrm{W}) = g(\mathrm{S}) = \text{white}\} \times \{1\}, \\
\delta_{a,\{a\}}((q_a, s_a)) &= \{g \in \mathcal{T} \mid g(\mathrm{W}) = q_a(\mathrm{E}), g(\mathrm{S}) = \text{white}\} \times \{0\}, \\
\delta_{a,\Sigma}((q_c, s_c)_{c\in\Sigma}) &= \begin{cases} \{g \in \mathcal{T} \mid g(\mathrm{W}) = \text{white}, g(\mathrm{S}) = q_b(\mathrm{N})\} \times \{1\} & \text{if } s_b = 1 \\ \{g \in \mathcal{T} \mid g(\mathrm{W}) = q_a(\mathrm{E}), g(\mathrm{S}) = q_b(\mathrm{N})\} \times \{0\} & \text{if } s_b = 0, \end{cases} \\
\delta_{b,\Sigma}((q_c, s_c)_{c\in\Sigma}) &= \begin{cases} \{g \in \mathcal{T} \mid g(\mathrm{W}) = \text{white}, g(\mathrm{S}) = q_a(\mathrm{N})\} \times \{1\} & \text{if } s_a = 1 \\ \{g \in \mathcal{T} \mid g(\mathrm{W}) = q_b(\mathrm{E}), g(\mathrm{S}) = q_a(\mathrm{N})\} \times \{0\} & \text{if } s_a = 0. \end{cases}
\end{aligned}$$

All tuples of local states are accepting. Now let $t = (V, E, \lambda)$ be a folded grid with $V = [k] \times [\ell]$ and let f be a tiling of this grid. We define a mapping $r : [k] \times [\ell] \to \mathcal{T}$ by

$$r((i, j)) = \begin{cases} (f(i, j), 1) & \text{if } i = 1 \\ (f(i, j), 0) & \text{if } i > 1 \end{cases}$$

and show that it is a successful run of $\mathcal{A}_\mathcal{T}$: Since f is a tiling of $[k] \times [\ell]$, we get $f(1, 1)(\mathrm{W}) = f(1, 1)(\mathrm{S}) = \text{white}$. Hence $r(1, 1) = (f(1, 1), 1) \in \delta_{a,\emptyset} = \delta_{\lambda(1,1),\mathrm{R}(1,1)}$. Now let $1 < i \leq \ell$. Since f is a tiling, we have $f(i, 1)(\mathrm{W}) = f(i - 1, 1)(\mathrm{E})$ and $f(i, 1)(\mathrm{S}) = \text{white}$. Hence $r(i, 1) = (f(i, 1), 0) \in \delta_{a,\{a\}}((f(i - 1, 1), s_a))$

for any $s_a \in \{0,1\}$. Note that $\lambda(i,1) = a$ and $\mathrm{R}(i,1) = \{a\}$. Furthermore, $\partial_a((i,1)) = (i-1,1)$. Hence we get $r(i,1) \in \delta_{\lambda(i,1),\mathrm{R}(i,1)}(r\partial_a((i,1)))$, i.e. the run condition of $\mathcal{A}_\mathcal{T}$ is satisfied at all nodes of the form $(i,1)$ with $i \in [k]$.

Next consider a vertex $(1,j)$ with $1 < j \leq k$ odd. Then $r(1,j-1)$ equals $(f(1,j-1),1)$. Since f is a tiling, we obtain $f(1,j)(\mathrm{W}) = \text{white}$ and $f(1,j)(\mathrm{S}) = f(1,j-1)(\mathrm{N})$. Hence

$$r(1,j) = (f(1,j),1) \in \delta_{a,\Sigma}(r(k,j-2), r(1,j-1)).$$

Since $j > 2$ is odd, $\lambda(1,j) = a$ and $\mathrm{R}(1,j) = \Sigma$. From $3 \leq j$ we get $(k,j-2) = \partial_a((1,j))$ and $(1,j-1) = \partial_b((1,j))$. Thus, we showed

$$r(1,j) \in \delta_{\lambda(1,j),\mathrm{R}(1,j)}(r\partial_a((1,j)), r\partial_b((1,j))).$$

For j even we can argue similarly. Hence we showed that the run condition of $\mathcal{A}_\mathcal{T}$ is satisfied at all nodes of the form $(i,1)$ or $(1,j)$ with $i \in [k]$ and $j \in [\ell]$.

It remains to consider a vertex (i,j) with $1 < i \leq k$ and $1 < j \leq \ell$. Assume j to be even. Since $i > 1$, $r(i,j-1) = (f(i,j-1),0)$. Since f is a tiling, we have $f(i,j)(\mathrm{W}) = f(i-1,j)(\mathrm{E})$ and $f(i,j)(\mathrm{S}) = f(i,j-1)(\mathrm{N})$. Hence $r(i,j) = (f(i,j),0) \in \delta_{b,\Sigma}(r(i,j-1), r(i-1,j))$. Since j is even, $\lambda(i,j) = b$. Note that $\mathrm{R}(i,j) = \Sigma$, $\partial_a((i,j)) = (i,j-1)$ and $\partial_b((i,j)) = (i-1,j)$. Hence we have $r(i,j) \in \delta_{\lambda(i,j),\mathrm{R}(i,j)}(r\partial_a((i,j)), r\partial_b((i,j)))$. Again, for j odd we can argue similarly. Thus the mapping r is a run of the ACA $\mathcal{A}_\mathcal{T}$ on t. Since any tuple is accepting, $t \in L(\mathcal{A}_\mathcal{T})$. Thus we showed that $\mathcal{A}_\mathcal{T}$ accepts all foldings of grids that allow a tiling.

Conversely, let r be a successful run of $\mathcal{A}_\mathcal{T}$ on the folded grid $t = (V,E,\lambda)$ with $V = [k] \times [\ell]$. We show that $f := \pi_1 \circ r$ is a tiling: First observe that $\pi_2 \circ r(i,j) = 1$ iff (i,j) is minimal in $(V,\preceq)$ or (i,j) has a lower neighbor $x \in V$ with $\lambda(x) \neq \lambda(i,j)$ and $\pi_2 \circ r(x) = 1$. Since $(i,j-1)$ is the only possible lower neighbor with a different label, $\pi_2 \circ r(i,j) = 1$ iff $i = j = 1$ or $\pi_2 \circ r(i,j-1) = 1$. Hence by induction $\pi_2 \circ r(i,j) = 1$ iff $i = 1$.

Since r is a run and $\lambda(1,1) = a$, we obtain $r(1,1) \in \delta_{\lambda(1,1),\emptyset}$. Hence $f(1,1)(\mathrm{W})$ and $f(1,1)(\mathrm{S})$ both equal white, i.e. f satisfies the conditions for a tiling at the point $(1,1)$.

Next let $1 < i \leq k$. Then $\mathrm{R}(i,1) = \{a\}$, $\lambda(i,1) = a$ and $\partial_a((i,1)) = (i-1,1)$. Since r is a run, this implies $r(i,j) \in \delta_{a,\{a\}}(r(i-1,1))$. The definition of $\delta_{a,\{a\}}$ implies $f(i,1)(\mathrm{W}) = f(i-1,1)(\mathrm{E})$ and $f(i,1)(\mathrm{S}) = \text{white}$ since $\pi_1 \circ r(i-1,1) = f(i-1,1)$. Hence $f \restriction ([k] \times [1])$ is a tiling.

Now let $1 < j \leq \ell$ be odd. Then $\mathrm{R}(1,j) = \Sigma$, $\lambda(1,j) = a$, $\partial_a((1,j)) = (k,j-2)$ and $\partial_b((1,j)) = (1,j-1)$. Since r is a run, this implies

$$r(1,j) \in \delta_{a,\Sigma}(r(k,j-2), r(1,j-1)).$$

Note that $\pi_2 \circ r(1,j-1) = 1$. Hence by the definition of $\delta_{a,\Sigma}$, $f(1,j)(\mathrm{W}) = \text{white}$ and $f(1,j)(\mathrm{S}) = f(1,j-1)(\mathrm{N})$. Since we can argue similarly for j even, the restriction $f \restriction ([1] \times [\ell])$ of f is a tiling.

It remains to consider the case $1 < i \leq k$ and $1 < j \leq \ell$. Then $\mathrm{R}(i,j) = \Sigma$. Now let j be even. Then $\lambda(i,j) = b$, $\partial_a((i,j)) = (i, j-1)$ and $\partial_b((i,j)) = (i-1, j)$. Since r is a run, $r(i,j) \in \delta_{b,\Sigma}(r(i,j-1), r(i-1,j))$. Since $i > 0$, we have $\pi_2 \circ r(i,j-1) = 0$. Thus the definition of $\delta_{b,\Sigma}$ yields $f(i,j)(\mathrm{W}) = f(i-1,j)(\mathrm{E})$ and $f(i,j)(\mathrm{S}) = f(i,j-1)(\mathrm{N})$. Again, for j odd we can argue similarly. Thus, f is indeed a tiling of the grid $[k] \times [\ell]$, i.e. we proved

Lemma 4.1.2 *Let t be the folding of the grid $[k] \times [\ell]$. Then $t \in L(\mathcal{A}_\mathcal{T})$ iff $[k] \times [\ell]$ admits a tiling. In particular, $L(\mathcal{A}_\mathcal{T}) \cap \mathbb{G}$ is the set of all foldings of tilable grids.* □

Note that $\mathcal{A}_\mathcal{T}$ accepts the foldings of an unbounded set of grids iff it accepts all folded grids. Lemma 4.1.1 and 4.1.2 imply that $\mathcal{A}_\mathcal{T}$ accepts an unbounded set of grids iff $\mathcal{T}$ admits an infinite tiling. Since the existence of an infinite tiling is undecidable, it is undecidable whether a given Σ-ACA $\mathcal{A}$ accepts the foldings of an unbounded set of grids and therefore whether $\mathbb{G} \subseteq L(\mathcal{A})$. Since $\mathbb{G}$ is not recognizable (cf. Lemma 4.1.9 below), this result cannot be used immediately to show the undecidability of the equivalence of ACAs. Nevertheless, it is a milestone in our proof that continues by showing that $\mathbb{D} \setminus \mathbb{G}$ is recognizable. This will imply that for a tiling systems $\mathcal{T}$ the set of all Σ-dags that are

a) no folded grid, or
b) a folded grid that can be tiled

is recognizable. But this set equals $\mathbb{D}$ iff the tiling system $\mathcal{T}$ allows an infinite tiling, and the latter is undecidable. Thus, indeed, it remains to show that $\mathbb{D} \setminus \mathbb{G}$ is recognizable.

Recall that $\mathrm{Ha} \subseteq \mathbb{D}$ is the set of Hasse-diagrams in $\mathbb{D}$. It is easily seen that $(V, E, \lambda) \in \mathbb{D}$ belongs to Ha iff it satisfies

$$(x,z),(y,z) \in E \Longrightarrow (x,y) \notin E^+$$

for all $x, y, z \in V$. Then $\mathbb{G} \subseteq \mathrm{Ha} \subseteq \mathbb{D}$ implies $\mathbb{D} \setminus \mathbb{G} = \mathbb{D} \setminus \mathrm{Ha} \cup \mathrm{Ha} \setminus \mathbb{G}$.

By Example 2.1.3, the set of Hasse-diagrams can be accepted by a Σ-ACM. Next, we prove that the *complement* of this set can be accepted using only finitely many states, i.e. by a Σ-ACA:

Lemma 4.1.3 *There exists a Σ-ACA $\mathcal{A}_{\mathrm{Ha}^{co}}$ with $\mathcal{A}_{\mathrm{Ha}^{co}} = \mathbb{D} \setminus \mathrm{Ha}$.*

Proof. We present an automaton $\mathcal{A}_a$ that recognizes all Σ-dags (V, E, λ) satisfying

there are an a-labeled vertex x and vertices y and z with $(x,z),(y,z) \in E$ and $(x,y) \in E^+$. $(\star)$

Let $\mathcal{A}_b$ be the analogous automaton that accepts all Σ-dags satisfying the above condition where x is supposed to carry the label b. Then $\mathbb{D} \setminus \mathrm{Ha} = L(\mathcal{A}_a) \cup L(\mathcal{A}_b)$ is recognizable.

To construct $\mathcal{A}_a$, let $Q_a = Q_b = \{0, 1, 2, 3\}$. Then, the transition functions are defined as follows:

$$\delta_{a,J}((q_j)_{j\in J}) = \begin{cases} \{0,1\} & \text{if } \{q_j \mid j \in J\} \subseteq \{0\} \\ \{3\} & \text{if } 3 \in \{q_j \mid j \in J\} \\ \{2,3\} & \text{if } \{q_j \mid j \in J\} = \{1,2\} \\ \{2\} & \text{otherwise, and} \end{cases}$$

$$\delta_{b,J}((q_j)_{j\in J}) = \begin{cases} \{0\} & \text{if } \{q_j \mid j \in J\} \subseteq \{0\} \\ \{3\} & \text{if } 3 \in \{q_j \mid j \in J\} \\ \{2,3\} & \text{if } \{q_j \mid j \in J\} = \{1,2\} \\ \{2\} & \text{otherwise.} \end{cases}$$

A tuple of states is accepting, i.e. belongs to F, if it contains the local state 3.

Let $t = (V, E, \lambda) \in \mathbb{D}$ satisfy $(\star)$. Then there are x, y, z with $\lambda(x) = a$, $(x, z), (y, z) \in E$ and $(x, y) \in E^+$. Define a mapping (cf. Figure 4.2) $r : V \to Q$ by

$$r(v) := \begin{cases} 0 & \text{if } x \not\leq v \\ 1 & \text{if } v = x \\ 3 & \text{if } z \leq v \\ 2 & \text{otherwise.} \end{cases}$$

In Figure 4.2, this mapping is depicted. There, solid vectors correspond to edges from E, the dotted vector connecting x and y denotes that $(x, y) \in E^+$. Furthermore, the dashed lines indicate the borders between e.g. $r^{-1}(0)$ and $r^{-1}(2)$, the values taken by r in an area is written there. Note that the small triangle around x depicts $r^{-1}(1)$ and contains x only.

We have to show $r(v) \in \delta_{\lambda(v),\mathrm{R}(v)}((r\partial_b(v))_{b\in\mathrm{R}(v)})$ $(\star\star)$ for any $v \in V$: Note that $r^{-1}(3)$ is a principal filter. Each nonminimal element of this filter reads a state 3, i.e. these elements satisfy $(\star\star)$. Since $(x, z), (y, z) \in E$ and x and y are different, they carry different labels. Hence $\lambda(x) = a$ implies $\lambda(y) = b$. Thus we have $\mathrm{R}(z) = \Sigma$, $r\partial_a(z) = 1$ and $r\partial_b(z) = 2$. Hence $(\star\star)$ holds for the minimal element z of this principal filter, too. The set $r^{-1}(2) = \{v \in V \mid x < v, z \not\leq v\}$ is convex. Note that $2 \in \delta_{c,J}((q_d)_{d\in J})$ iff $3 \notin \{q_d \mid d \in J\} \not\subseteq \{0\}$. Now let $v \in r^{-1}(2)$. Since $z \not\leq v$, $3 \notin \{r\partial_c(v) \mid c \in \mathrm{R}(v)\}$. If v is nonminimal in $r^{-1}(2)$, it satisfies $(\star\star)$ since it reads the state 2. The minimal elements read the state at the vertex x which equals 1. Hence they satisfy $(\star\star)$, too. Note that $\{\partial_c(x) \mid c \in \mathrm{R}(x)\} \subseteq r^{-1}(0)$. Hence x satisfies $(\star\star)$. Since, finally, $r^{-1}(0)$ is an order ideal, $(\star\star)$ holds for its elements, too. Thus, r is a successful run of $\mathcal{A}_a$.

Conversely, let r be a successful run of $\mathcal{A}_a$ on a Σ-dag $t = (V, E, \lambda)$. For simplicity, let $\leq := E^\star$ denote the partial order induced by the edge relation E.

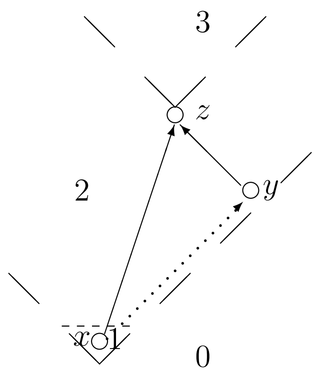


Figure 4.2: cf. Proof of Lemma 4.1.3

Then $r^{-1}(3)$ is a filter, i.e. an upward closed subset of V with respect to $\leq$. Since the run is successful, this filter is not empty. Note that an element of $r^{-1}(3)$ is minimal in this filter iff it reads a state 1 and a state 2. Since the filter in question is not empty, it contains a minimal element z and there are elements $x, y \in V$ with $(x, z), (y, z) \in E$, $r(x) = 1$ and $r(y) = 2$. Whenever a vertex v carries the state 2, it reads the state 1 or the state 2. Hence, by induction, we find $x' \in V$ with $(x', y) \in E^+$ and $r(x') = 1$. Since $r(x) = r(x') = 1$, they both carry the label a implying that they are comparable. Furthermore, $\{r\partial_c(x) \mid c \in \mathrm{R}(x)\} \subseteq \{0\}$ and $\{r\partial_c(x') \mid c \in \mathrm{R}(x')\} \subseteq \{0\}$. Since, as is easy to see, $r^{-1}(0)$ is an order ideal (i.e. downword closed), this implies $x = x'$. Hence $(x, y) \in E^+$, i.e. all Σ-dags accepted by $\mathcal{A}_a$ satisfy the condition $(\star)$. $\square$

Before showing that $\mathrm{Ha}\setminus\mathbb{G}$ is recognizable within Ha, we need an internal characterization of those Hasse-diagrams that are folded grids. This characterization is based on the notion of an *alternating covering chain*: Let $t = (V, \longrightarrow\!\!\!<, \lambda) \in \mathrm{Ha}$ and $C \subseteq V$. The set C is called *alternating covering chain* if it is a chain (with respect to $\leq := \longrightarrow\!\!\!<^\star$) such that

1. for all $y \in C$ with $y \neq \min(C)$, there exists $x \in C$ with $x \longrightarrow\!\!\!< y$ and $\lambda(x) \neq \lambda(y)$, and

2. for all $y \in C$ with $y \neq \max(C)$, there exists $z \in C$ with $y \longrightarrow\!\!\!< z$ and $\lambda(y) \neq \lambda(z)$.

Since we consider only Hasse-diagrams of width 2, it is easy to see that for any $x \in V$ there exists a unique maximal alternating covering chain C with $x \in C$. This chain is denoted by $C(x)$.

Lemma 4.1.4 *Let $t = (V, \multimap, \lambda) \in \mathrm{Ha}$ be a Hasse-diagram. Then $t \in \mathbb{G}$ if and only if*

(1) for any $x \in V$, the element $\min C(x)$ does not dominate any b-labeled vertex, and

(2) for any $x, y \in V$ with $x \multimap y$ such that y does not dominate any b-labeled element, we have (A) $\forall x' \in C(x) \exists y' \in C(y) : x' \multimap y'$
(B) $\forall y' \in C(y) \exists x' \in C(x) : x' \multimap y'$.

Proof. First, let $k, \ell \in \mathbb{N}^+$ and define $K_i = \{i\} \times [\ell]$ for $1 \leq i \leq k$. Then, in the folded grid $([k] \times [\ell], E)$, K_i is a chain. Since $((i,j),(i,j+1)) \in E$ and $\lambda(i,j) = a$ iff j is odd, it is even an alternating covering chain. We show that it is maximal: Let $x \multimap \min(K_i) = (i,1)$. Then $x = (i-1,1)$ and therefore $\lambda(x) = a = \lambda(i,1)$. Hence K_i cannot be extended downwards. Similarly, let $y \in [k] \times [\ell]$ with $(i,\ell) = \max K_i \multimap y$. Then $y = (i+1,\ell)$ and therefore carries the same label as (i,ℓ). Hence K_i is indeed a maximal alternating covering chain. Hence, for $(i,j) \in [k] \times [\ell]$, $C(i,j) = K_i$. Now it is routine to check properties (1) and (2) (cf. Figure 4.1).

Conversely, suppose $t = (V, \multimap, \lambda)$ satisfies the conditions (1) and (2) and let $\leq$ denote the transitive and reflexive closure of $\multimap$. By (1), $\lambda(\min C(x)) = a$ for any $x \in V$. Now let $\{a_1, a_2, \ldots, a_k\} = \{\min C(x) \mid x \in V\}$. Since each a_i is labeled by a, this set forms a chain. So let $a_1 < a_2 \cdots < a_k$. Since (again by (1)) none of the elements a_i dominates a b-labeled vertex, we have even $a_1 \multimap a_2 \ldots \multimap a_k$. For simplicity, let $C_i := C(a_i)$. The tuple $(C_i)_{i \in [k]}$ is a partition of V. We denote the jth element of the ith alternating covering chain C_i by x_i^j, i.e. $C_i = \{x_i^1, x_i^2, x_i^3 \ldots, x_i^{\ell_i}\}$ with $a_i = x_i^1 \multimap x_i^2 \multimap x_i^3 \ldots \multimap x_i^{\ell_i}$. Note that $\lambda(x_i^j) = a$ iff j is odd and that ℓ_i is the size of the chain C_i.

Claim 1 for any $1 \leq i < k$ and any $1 \leq j \leq \ell_i$ we have $j \leq \ell_{i+1}$ and $x_i^j \multimap x_{i+1}^j$.

This is shown inductively on j. Clearly, $1 \leq \ell_{i+1}$ since $a_{i+1} = x_{i+1}^1$. We already remarked that $x_i^1 \multimap x_{i+1}^1$. Now suppose $1 < j \leq \ell_i$ and $x_i^{j-1} \multimap x_{i+1}^{j-1}$. We can apply (2A) since $x_i^1 \multimap x_{i+1}^1$ and $x_i^j \in C_i$. Hence there exists $y' \in C_{i+1}$ with $x_i^j \multimap y'$. Since C_{i+1} is a chain containing x_{i+1}^{j-1} and y', these two elements are comparable. If $y' \leq x_{i+1}^{j-1}$, we had $x_i^{j-1} \multimap x_i^j \multimap y' \leq x_{i+1}^{j-1}$, contradicting $x_i^{j-1} \multimap x_{i+1}^{j-1}$. Hence $x_{i+1}^{j-1} < y'$. Since they both belong to the alternating covering chain C_{i+1}, there exists $y'' \in C_{i+1}$ with $x_{i+1}^{j-1} \multimap y'' \leq y'$. From $x_i^1 \multimap x_{i+1}^1$ and $y'' \in C_{i+1}$, we obtain by (2B) the existence of $x'' \in C_i$ with $x'' \multimap y''$. Now the elements x_i^{j-1} and x'' are comparable. If $x'' \leq x_i^{j-1}$, we had $x'' \leq x_i^{j-1} \multimap x_{i+1}^{j-1} \multimap y''$, contradicting $x'' \multimap y''$. Hence $x_i^{j-1} < x''$. Since $x'', x_i^j \in C_i$, they are comparable. Now $x_i^{j-1} \multimap x_i^j$ implies $x_i^j \leq x''$. Hence we have $x_i^j \leq x'' \multimap y'' \leq y'$ and $x_i^j \multimap y'$. This implies $y' = y''$. Recall that

$x_{i+1}^{j-1} \longrightarrow\!\!< y'' \in C_{i+1}$. Hence we showed $y' = x_{i+1}^j$, i.e. $j \le \ell_{i+1}$ and $x_i^j \longrightarrow\!\!< x_{i+1}^j$ as claimed.

Claim 2 For any $1 \le i < k$ and any $1 \le j \le \ell_{i+1}$, we have $j \le \ell_i$ and $x_i^j \longrightarrow\!\!< x_{i+1}^j$.

Again, this is shown by induction on j. Clearly, $1 \le \ell_i$ since $a_i = x_i^1$. Now, $x_i^1 \longrightarrow\!\!< x_{i+1}^1$ follows from Claim 1. Now suppose $1 < j \le \ell_i$ such that $x_i^{j'} \longrightarrow\!\!< x_{i+1}^{j'}$ for any $j' < j$. Then we can apply (2B) since $x_i^1 \longrightarrow\!\!< x_{i+1}^1$ and $x_{i+1}^j \in C_i$. Hence there exists $x' \in C_i$ with $x' \longrightarrow\!\!< x_{i+1}^j$. Since $x_i^{j'} \longrightarrow\!\!< x_{i+1}^{j'}$ for $j' < j$, we have $x' \ne x_i^{j'}$ for $j' < j$. Hence the chain C_i contains at least j elements, i.e. $j \le \ell_i$. Now $x_i^j \longrightarrow\!\!< x_{i+1}^j$ follows from Claim 1.

Note that Claim 1 in particular implies $\ell_1 \le \ell_2 \cdots \le \ell_k$. Similarly, by Claim 2, $\ell_1 \ge \ell_2 \cdots \ge \ell_k$, i.e. $\ell_1 = \ell_2 \cdots = \ell_k =: \ell$. Hence $g : [k] \times [\ell] \to V : (i,j) \mapsto x_i^j$ is a bijection.

Claim 3 For $1 \le i \le i' \le k$ and $1 \le j \le \ell$, $x_{i'}^j$ is the least element of $C_{i'}$ dominating x_i^j, i.e. $x_{i'}^j = \min\{x \in C_{i'} \mid x_i^j \le x\}$.

This is trivial for $i = i'$. For $i + 1 = i'$ it is clear by Claim 1. By induction, suppose we showed that $x_{i'-1}^j$ is the least element of $C_{i'-1}$ that dominates x_i^j. Let $x_{i'}^{j'}$ be the least element of $C_{i'}$ dominating x_i^j. Since $|\Sigma| = 2$, this element $x_{i'}^{j'}$ has at most two lower neighbors, namely $x_{i'}^{j'-1}$ (if $j' > 1$) since it precedes $x_{i'}^{j'}$ in the alternating covering chain $C_{i'}$, and $x_{i'-1}^{j'}$ by Claim 1. Since x_i^j is not dominated by $x_{i'}^{j'-1} \in C_{i'}$, we therefore have $x_{i'-1}^{j'} \ge x_i^j$. Hence, by the induction hypothesis, $j' \ge j$. Since $x_i^j \le x_{i'-1}^j \longrightarrow\!\!< x_{i'}^j$, we therefore showed that $x_{i'}^j$ is the least element in $C_{i'}$ dominating x_i^j.

Now we show that the bijection $g : ([k] \times [\ell], E) \to (V, \longrightarrow\!\!<)$ is order preserving: Let $(i,j), (i',j') \in [k] \times [\ell]$ with $((i,j),(i',j')) \in E$. Then we have

a) $i = i'$ and $j + 1 = j'$, or
b) $i + 1 = i'$ and $j = j'$, or
c) $i = k$, $i' = 1$ and $j + 2 = j'$.

In the first case, we get immediately $x_i^j \longrightarrow\!\!< x_{i'}^{j'}$ since they are consecutive elements of the alternating covering chain C_i. In the second case, Claim 1 implies $x_i^j \longrightarrow\!\!< x_{i'}^{j'}$. In the third case, we get $g(i,j) = x_k^j$ and $g(i',j') = x_1^{j+2}$. Since j and $j+2$ have the same parity, $\lambda(x_k^j) = \lambda(x_1^{j+2})$ and therefore x_k^j and x_1^{j+2} are comparable. If $x_1^{j+2} \le x_k^j$, by Claim 3 we have $x_k^j \ge x_k^{j+2}$ which is properly larger than x_k^j, a contradiction. Hence $x_k^j < x_1^{j+2}$, i.e. $g(i,j) < g(i',j')$. Thus we showed that g is order preserving.

Next we show that $f = g^{-1} : (V, \longrightarrow\!\!<) \to ([k] \times [\ell], E)$ is order preserving. So let $x_i^j, x_{i'}^{j'} \in V$ with $x_i^j \longrightarrow\!\!< x_{i'}^{j'}$. If x_i^j and $x_{i'}^{j'}$ carry the same label, j and j' have the same parity. Hence $\lambda(i,j) = \lambda(i',j')$. This ensures that (i,j) and (i',j') are comparable. If $(i',j') \preceq (i,j)$, we get $x_{i'}^{j'} = g(i',j') \le g(i,j) = x_i^j$ since g is order preserving. But this contradicts the assumption $x_i^j \longrightarrow\!\!< x_{i'}^{j'}$. Hence $(i,j) < (i',j')$, i.e. $f(x_i^j) < f(x_{i'}^{j'})$. Now assume that x_i^j and $x_{i'}^{j'}$ carry different labels. Since $x_i^j \longrightarrow\!\!< x_{i'}^{j'}$ this implies that they belong to the same maximal alternating covering chain, i.e. $i = i'$ and $j + 1 = j'$. But then $f(x_i^j) = (i, j)E(i, j + 1) = f(x_{i'}^{j'})$. $\square$

Now we are able to show that $\mathbb{D} \setminus \mathbb{G}$ is recognizable by a Σ-ACA.

Lemma 4.1.5 *There exists a Σ-ACA $\mathcal{A}$ such that $L(\mathcal{A}) = \mathbb{D} \setminus \mathbb{G}$.*

Proof. By Lemma 4.1.3, there exists a Σ-ACA $\mathcal{A}_{\mathrm{Ha}^{co}}$ with $L(\mathcal{A}_{\mathrm{Ha}^{co}}) = \mathbb{D} \setminus \mathrm{Ha}$. From $\mathbb{G} \subseteq \mathrm{Ha} \subseteq \mathbb{D}$, we get $\mathbb{D} \setminus \mathbb{G} = \mathbb{D} \setminus \mathrm{Ha} \cup \mathrm{Ha} \setminus \mathbb{G}$. Hence it suffices to construct a Σ-ACA $\mathcal{A}$ with $L(\mathcal{A}) \cap \mathrm{Ha} = \mathrm{Ha} \setminus \mathbb{G}$.

As a prerequisite, we give an ACA $\mathcal{A}^1$ that marks all vertices which dominate a b-labeled vertex: Let $Q_a^1 = Q_b^1 = \{0, 1\}$ where 0 stands for "does not dominate any b-labeled vertex". The transition functions are defined by $\delta_{b,J}^1((q_c)_{c \in J}) = \{1\}$ and

$$\delta_{a,J}^1((q_c)_{c \in J}) = \begin{cases} \{1\} & \text{if } 1 \in \{q_c \mid c \in J\} \\ \{0\} & \text{otherwise} \end{cases}$$

for any $J \subseteq \Sigma$ and $q_c \in Q_c^1$. Then, obviously, for any run r of $\mathcal{A}^1$ on a Hasse-diagram $t = (V, \longrightarrow\!\!<, \lambda)$, we have $r(x) = 0$ iff $b \notin \{\lambda(y) \mid y \leq x\}$ as claimed.

Next we prove that the set of Hasse-diagrams violating Lemma 4.1.4(1) can be accepted by a Σ-ACA relative to Ha: Note that Lemma 4.1.4(1) is violated iff there exists an a-labeled vertex x that dominates, but does not cover any b-labeled vertex. To find such a vertex, we enrich the automaton $\mathcal{A}^1$ by a second component that propagates the information whether a transition of the form $\delta_{a,\{a\}}(1)$ has been applied. If the run of this enriched automaton uses such a transition, it accepts, otherwise, it rejects. Note that the application of the transition $\delta_{a,\{a\}}(1)$ at a vertex x denotes that x is a-labeled, does not cover any b-labeled vertex, and dominates such a vertex according to the definition of $\mathcal{A}^1$. Hence the enriched automaton $\mathcal{A}_{\neg(1)}$ accepts precisely those Σ-pomsets that violate Lemma 4.1.4(1).

It remains to prove that the negation of statement (2) of Lemma 4.1.4 can be recognized. First, we show how to guess the element x and to mark the chain $C(x)$: Let $Q' = \{0, 1, 2\}$ where 2 stands for "belongs to $C(x)$", 1 for "does not belong to $C(x)$, but dominates an element of $C(x)$", and 0 for "does not dominate any element from $C(x)$. The transition functions of the automaton $\mathcal{A}'$ are given by

$$\delta'_{a,J}((q_c)_{c \in J}) = \begin{cases} \{2\} & \text{if } b \in J, q_b = 2 \\ \{1\} & \text{if } a \in J, q_a > 0 \text{ or } b \in J, q_b = 1 \\ \{0, 2\} & \text{otherwise} \end{cases}$$

$$\delta'_{b,J}((q_c)_{c\in J}) = \begin{cases} \{2\} & \text{if } a \in J, q_a = 2 \\ \{1\} & \text{if } b \in J, q_b > 0 \text{ or } a \in J, q_a = 1 \\ \{0\} & \text{otherwise.} \end{cases}$$

Let r be a run of this automaton on the Hasse-diagram $t = (V, -\!\!<, \lambda)$. If $x \in V$ with $r(x) > 0$ then, either x covers some y with $r(y) > 0$, or $\lambda(x) = a$. Hence the set of all vertices x with $r(x) > 0$ (if not empty) is a principal filter (with respect to the partial order $\leq$ induced by $-\!\!<$) whose minimal element is labeled by a. In this principal filter, $r(x) = 2$ holds iff x covers some y with different label and $r(y) = 2$, or x is the minimal element of the principal filter. Hence, the set of all $x \in V$ with $r(x) = 2$ forms an alternating covering chain whose least element is labeled by a. Using the automaton $\mathcal{A}^1$, it is easily possible to ensure that this minimal element does not dominate any b-labeled vertex. Thus, we can construct a Σ-ACA $\mathcal{A}^2 = ((Q_a^2, Q_b^2), (\delta_{a,J}^2, \delta_{b,J}^2)_{J\subseteq\Sigma}, F^2)$ and subsets $S^x, S^y \subseteq Q_a^2 \cup Q_b^2$ such that for any successful run r on a Hasse-diagram $t = (V, -\!\!<, \lambda)$, we have

(a) $r^{-1}(S^x)$ and $r^{-1}(S^x)$ form nonempty alternating covering chains with minimal elements x and y,

(b) x and y do not dominate any b-labeled vertex, and

(c) $x -\!\!< y$.

Note that $t = (V, -\!\!<, \lambda)$ violates Lemma 4.1.4(2B) iff there exists a successful run r of the ACA $\mathcal{A}^2$ on t and an element $y' \in V$ with $r(y') \in S^y$ that does not cover any $x' \in V$ with $r(x') \in S^x$. Since this can easily be checked, we are therefore able to construct a Σ-ACA $\mathcal{A}_{\neg(2B)}$ such that $L(\mathcal{A}_{\neg(2B)}) \cap \mathrm{Ha}$ is the set of all Hasse-diagrams t that violate Lemma 4.1.4(2B).

To check the negation of Lemma 4.1.4(2A), we again use the automaton $\mathcal{A}^2$ that marks nondeterministically two alternating covering chains $C(x)$ and $C(y)$. This automaton will be enriched by the ability to mark some vertices from $C(x)$ and check that they are not covered by any element from $C(y)$. More formally, let $Q^3 = Q^2 \times \{0, 1\}$. For $z \in \{a, b\}$, the transition function is given by

$$\delta_{z,J}^3((q_c, s_c)_{c\in J}) = \begin{cases} (\delta_{z,J}^2((q_c)_{c\in J}) \setminus S^y) \times \{0,1\} & \text{if } \exists c \in J : (q_c \in S^x \wedge s_c = 1) \\ \delta_{z,J}^2((q_c)_{c\in J}) \times \{0,1\} & \text{otherwise.} \end{cases}$$

As before, a successful run r of $\mathcal{A}^3$ on a Hasse-diagram $t = (V, -\!\!<, \lambda)$ determines two alternating covering chains $C(x) = r^{-1}(S^x \times \{0, 1\})$ and $C(y) = r^{-1}(S^y \times \{0, 1\})$. Now suppose there is some $x' \in V$ with $r(x) \in S^x \times \{1\}$. Then, according to the definition of the transition function $\delta_{z,J}$, there is no $y' \in C(y)$ with $x' -\!\!< y'$ (since otherwise $r(y') \notin S^y \times \{0, 1\}$, a contradiction). Hence, in this case, (2A) does not hold. Since the existence of a vertex x with $r(x) \in S^x \times \{1\}$ is easily checked, we can construct a Σ-ACA $\mathcal{A}_{\neg(2A)}$ such that $L(\mathcal{A}_{\neg(2A)}) \cap \mathrm{Ha}$ is the set of all Hasse-diagrams t that violate condition Lemma 4.1.4(2A).

Combining the automata $\mathcal{A}_{\neg(1)}$, $\mathcal{A}_{\neg(2B)}$ and $\mathcal{A}_{\neg(2A)}$, we get a Σ-ACA $\mathcal{A}'$ such that $L(\mathcal{A}') \cap \mathrm{Ha}$ is the set of all Hasse-diagrams t that violate condition (1) or (2). Using Lemma 4.1.3 and Lemma 4.1.4, one gets a Σ-ACA $\mathcal{A}$ with $L(\mathcal{A}) = \mathbb{D} \setminus \mathbb{G}$. □

Lemma 4.1.6 *Let $\mathcal{T}$ be a set of tiles. Then there exists a Σ-ACA $\mathcal{A}(\mathcal{T})$ such that $L(\mathcal{A}(\mathcal{T}))$ is the set of all Σ-dags that are no folded grid or a folding of a grid that allows a tiling by $\mathcal{T}$.*

Proof. By Lemma 4.1.5, there exists a Σ-ACA $\mathcal{A}'$ with $L(\mathcal{A}') = \mathbb{D} \setminus \mathbb{G}$. By Lemma 4.1.2, $L(\mathcal{A}_{\mathcal{T}}) \cap \mathbb{G}$ is the set of all foldings of tilable grids. Let $\mathcal{A}(\mathcal{T})$ denote the disjoint union of $\mathcal{A}'$ and $\mathcal{A}_{\mathcal{T}}$. Then $\mathcal{A}(\mathcal{T})$ has the desired property. □

Theorem 4.1.7 *Let Σ be an alphabet with at least two letters. Then there is no algorithm that on input of a Σ-ACA $\mathcal{A}$ decides whether it accepts all Σ-dags, i.e. whether $L(\mathcal{A}) = \mathbb{D}$.*

Proof. It is clearly sufficient to consider the case $\Sigma = \{a, b\}$. Let $\mathfrak{C}$ be a finite set of colors and $\mathcal{T}$ be a set of tiles. By Lemma 4.1.6, $\mathcal{A}(\mathcal{T})$ accepts all Σ-dags iff all grids allow a tiling. But this is equivalent to the existence of an infinite tiling which is undecidable. □

Since there is a Σ-ACA that accepts all Σ-dags, we get as an immediate

Corollary 4.1.8 *Let Σ be an alphabet with at least two letters. Then the equivalence of Σ-ACAs, i.e. the question whether $L(\mathcal{A}_1) = L(\mathcal{A}_2)$, is undecidable.*

With $M = \mathbb{G}$, the following lemma implies in particular that $\mathbb{G}$ cannot be accepted by a Σ-ACA. Thus, the set of recognizable languages in $\mathbb{D}$ is not closed under complementation since $\mathbb{D} \setminus \mathbb{G}$ is acceptable by Lemma 4.1.5. Our next goal is to show that it is even undecidable whether a given ACA can be complemented.

Lemma 4.1.9 *Let $M \subseteq \mathbb{G}$ be such that for any $i \in \mathbb{N}^+$ there exist $k, \ell \in \mathbb{N}^+$ with $i \leq k$ and $1 < \ell$ such that $([i] \times [\ell], E, \lambda) \in M$. Let $\mathcal{A}$ be a Σ-ACA with $M \subseteq L(\mathcal{A})$. Then $L(\mathcal{A}) \not\subseteq \mathbb{G}$.*

Proof. Let $k \geq |Q_b| + 3$ and $1 < \ell$ such that $([k] \times [\ell], E, \lambda) \in M$. Then $k - 3$ is at least the number of states of the second process of $\mathcal{A}$. Since $\mathcal{A}$ accepts all elements of M, there is a successful run r of $\mathcal{A}$ on $([k] \times [\ell], E, \lambda)$. Since k is sufficiently large, there exist m, n with $1 < m < n < k$ such that $r(m, \ell) = r(n, \ell)$.

Now delete all vertices (m', ℓ) in $[k] \times [\ell]$ with $m < m' \leq n$, i.e. define P to be the set $[k] \times [\ell] \setminus \{(m', \ell) \mid m < m' \leq n\}$. Furthermore, let $E' := (E \cap P^2) \cup \{((m, \ell), (n+1, \ell))\}$. Then one can easily check that $(P, E', \lambda \restriction P)$ is a Σ-dag that does not belong to $\mathbb{G}$. We show that the restriction of the run r to P is a successful run of $\mathcal{A}$ on (P, E', λ): Note that the node $(n+1, \ell)$ is the only one from P whose set of lower neighbors in $([k] \times [\ell], E, \lambda)$ (where it equals $\{(n, \ell), (n+1, \ell-1)\}$) and in $(P, E', \lambda \restriction P)$ (where it equals $\{(m, \ell), (n+1, \ell-1)\}$) differ. But since $r(m, \ell) = r(n, \ell)$, this does not influence the run condition. Hence $(P, E', \lambda \restriction P)$ is accepted by $\mathcal{A}$, i.e. $L(\mathcal{A}) \not\subseteq \mathbb{G}$. $\square$

Theorem 4.1.10 *Let Σ be an alphabet with at least two letters. Then there is no algorithm that on input of a Σ-ACA $\mathcal{A}$ decides any of the following questions:*

1. *Is $\mathbb{D} \setminus L(\mathcal{A})$ recognizable?*
2. *Is $\mathcal{A}$ equivalent with some deterministic Σ-ACA?*

Proof. Again, it is sufficient to consider the case $\Sigma = \{a, b\}$. Let $\mathcal{T}$ be a finite set of tiles and let $\mathcal{A}(\mathcal{T})$ be the Σ-ACA from Lemma 4.1.6, i.e. $\mathcal{A}(\mathcal{T})$ accepts a Σ-dag $t = (V, E, \lambda)$ iff

a) t is no folded grid, or
b) t is a folded grid that allows a tiling by $\mathcal{T}$.

Then $L := \mathbb{D} \setminus L(\mathcal{A}(\mathcal{T}))$ is the set of all folded grids that do not allow a tiling by $\mathcal{T}$. We show that L is recognizable iff $\mathcal{T}$ allows an infinite tiling:

If $\mathcal{T}$ allows an infinite tiling, L is empty and therefore trivially recognizable. Conversely, let $\mathcal{A}$ be a Σ-ACA that recognizes L. By contradiction, suppose that $\mathcal{T}$ does not allow an infinite tiling. Then, by Lemma 4.1.1, the set of tilable grids is not unbounded, i.e. there exist $k, \ell \in \mathbb{N}^+$ such that for any $k' \geq k$ and $\ell' \geq \ell$ the grid $[k'] \times [\ell']$ cannot be tiled. Thus, any folding of a grid $[k'] \times [\ell]$ with $k' \geq k$ belongs to L. Let $M := \{[k'] \times [\ell] \mid k' \geq k\}$. Then this set satisfies the condition of Lemma 4.1.9 and $M \subseteq L = L(\mathcal{A})$. Hence $L(\mathcal{A}) \not\subseteq \mathbb{G}$, contradicting $L(\mathcal{A}) = L \subseteq \mathbb{G}$.

This finishes the proof of the first statement since the existence of an infinite tiling and therefore the recognizability of $\mathbb{D} \setminus L(\mathcal{A}(\mathcal{T}))$ is undecidable.

Along the same line we can prove the second statement: If $\mathcal{A}(\mathcal{T})$ is equivalent with a deterministic Σ-ACA, $\mathbb{D} \setminus L(\mathcal{A}(\mathcal{T}))$ is recognizable since any deterministic Σ-ACA can be complemented. Hence $\mathbb{D} \setminus L(\mathcal{A}(\mathcal{T})) = \emptyset$ and therefore $\mathcal{T}$ allows an infinite tiling. Conversely, if $\mathcal{T}$ allows an infinite tiling, $\mathbb{D} \setminus L(\mathcal{A}(\mathcal{T})) = \emptyset$ implying $L(\mathcal{A}(\mathcal{T})) = \mathbb{D}$. But this set can be recognized deterministically, i.e. the

ACA $\mathcal{A}(\mathcal{T})$ is equivalent with a deterministic one. $\square$

By Corollary 4.1.8, the equivalence of two Σ-ACAs is undecidable. Rice's Theorem states that for any Turing machine $\mathcal{M}$, the set of equivalent Turing machines is not recursive. This does not hold for Σ-ACAs in general: Let $L \subseteq \mathbb{D}$ be finite. Then the set of Σ-ACAs $\mathcal{A}$ with $L(\mathcal{A}) = L$ is recursive: Let $n := \max\{|V| \mid (V, E, \lambda) \in L\}$. Then, given a Σ-ACA $\mathcal{A}$, one can first check whether $L(\mathcal{A}) \cap \{(V, E, \lambda) \in \mathbb{D} \mid |V| \leq n\} = L$ since the set $\{(V, E, \lambda) \in \mathbb{D} \mid |V| \leq n\}$ is finite and $L(\mathcal{A})$ is recursive. In addition, one can easily construct from $\mathcal{A}$ a Σ-ACA $\mathcal{A}'$ such that $L(\mathcal{A}') = L(\mathcal{A}) \setminus \{(V, E, \lambda) \in \mathbb{D} \mid |V| \leq n\}$ (the Σ-ACA $\mathcal{A}'$ has to count the vertices up to n and accepts only if $\mathcal{A}$ accepts and there are at least $n+1$ nodes). Now, by Theorem 3.3.4, it can be checked whether $L(\mathcal{A}') = \emptyset$, i.e. whether $L(\mathcal{A}) = L$.

It is not clear whether there are other sets $L \subsetneq \mathbb{D}$ such that the question whether $L(\mathcal{A}) = L$ can be decided.

Chapter 5

The expressive power of ACAs

This chapter deals with the question which properties can be expressed by a Σ-ACA. By Corollary 3.3.5, the expressible properties are at least recursive. On the other hand, Example 2.1.6 shows that not all recursive sets of Σ-dags are recognizable. The situation is similar to that of finite sequential automata and sets of words: Any language that is accepted by a finite sequential automaton is recursive, but the converse is false. In this setting, several answers are known to the question which properties can be checked by a finite sequential automaton: Kleene showed that these are precisely the rational properties. By the Myhill-Nerode Theorem, a property can be checked by a finite sequential automaton if its syntactic monoid is finite. Furthermore, Büchi and Elgot [Büc60, Elg61] showed that a property of words can be checked by a finite automaton if and only if it can be expressed in the monadic second order logic. This relation between a model of a computational device (finite sequential automata) and monadic second order logic is a paradigmatic result. It has been extended in several directions, e.g. to infinite words [Büc60], to trees [Rab69], to finite [Tho90b] and to real [EM93, Ebi94] traces, and to computations of concurrent automata [DK96, DK98]. This relation does clearly not hold for Σ-ACMs in general: Example 2.1.5 provides a word language that can be accepted by a Σ-ACM (that is even monoton and effective), but not by a finite sequential automaton. Hence, this set of Σ-dags cannot be axiomatized in monadic second order logic. Therefore, we examine whether there is such a close relation between Σ-ACAs and MSO.

It is shown that any recognizable set can be axiomatized by a sentence of the monadic second order logic. Since the converse is not true (cf. Example 2.1.6), we then restrict furthermore to so called (Σ, k)-dags and show that a set of (Σ, k)-dags is recognizable (relative to the set of all (Σ, k)-dags and even relative to the set of all Σ-dags) iff it can be monadically axiomatized. But it is necessary to allow nondeterminism in the automata since the expressive power of deterministic Σ-ACAs is shown to be strictly weaker.

5.1 From ACAs to MSO

In this section, we will prove that for any ACAs $\mathcal{A}$, there exists a monadic sentence which axiomatizes the language accepted by $\mathcal{A}$. The proof of this result follows [DG96] (see also [DGK00]). There, the restricted case of Σ-dags that are Hasse-diagrams was dealt with. The only difference between this former result and the result we are going to prove now is the following: The monadic second order logic considered in [DG96] makes statements on partial orders and not on dags. Since the partial order $E^\star$ can be expressed by a monadic formula over dags, this is no difference as far as the expressive power is concerned. But one needs more quantifier alternations which is the reason why in our setting the following theorem states only the existence of a monadic sentence which might not be existential.

Theorem 5.1.1 *Let $\mathcal{A}$ be a possibly nondeterministic Σ-ACA. There exists a monadic sentence φ over Σ such that*

$$L(\mathcal{A}) = \{t \in \mathbb{D} \mid t \models \varphi\}.$$

Proof. Let $\mathcal{A} = ((Q_a)_{a\in\Sigma}, (\delta_{a,J})_{a\in\Sigma,J\subseteq\Sigma}, F)$ be a Σ-ACA. We will construct a monadic sentence which will be satisfied exactly by the Σ-dags that are accepted by $\mathcal{A}$. Let k be the number of states in $\bigcup_{a\in\Sigma} Q_a$. We may assume that $\bigcup_{a\in\Sigma} Q_a = [k] = \{1, \dots, k\}$. The following sentence ψ claims the existence of a successful run of the automaton.

$$\psi = \exists X_1 \dots \exists X_k \ \big(\mathrm{partition}(X_1, \dots, X_k) \wedge \big(\forall x \ \mathrm{trans}(x)\big) \wedge \mathrm{accepted}\big)$$

We will now explain this sentence and give the sub-formulas *partition*, *trans* and *accepted*. A run over a Σ-dag $t = (V, E, \lambda)$ is coded by the set-variables $X_1, \dots, X_k$. More precisely, X_i stands for the set of vertices mapped to the state i by the run. The formula partition$(X_1, \dots, X_k)$ ensures that the set-variables $X_1, \dots, X_k$ describe a mapping from V to $\bigcup_{a\in\Sigma} Q_a$:

$$\mathrm{partition}(X_1, \dots, X_k) = \left(\forall x \bigvee_{i\in[k]} x \in X_i\right) \wedge \left(\bigwedge_{1\le i<j\le k} X_i \cap X_j = \emptyset\right).$$

Then, we have to claim that this labeling of vertices by states agrees with the transition functions of the automaton.

$$\mathrm{trans}(x) = \bigvee_{q\in\delta_{a,J}((q_b)_{b\in J})} \Big(\lambda(x) = a \wedge x \in X_q \wedge \forall y \ ((y, x) \in E \to \lambda(y) \in J) \\ \wedge \bigwedge_{b\in J} \exists y \ ((y, x) \in E \wedge \lambda(y) = b \wedge y \in X_{q_i})\Big)$$

where the disjunction ranges over all letters $a \in \Sigma$, states $q \in Q_a$, subsets $J \subseteq \Sigma$ and tuples $(q_b)_{b\in J} \in \prod_{b\in J} Q_b$ such that $q \in \delta_{a,J}((q_b)_{b\in J})$.

It remains to state that the run reaches a final state of the automaton. Let *accepted* denote the disjunction of the following sentence for $(f_b)_{b\in J} \in F$:

$$\Big(\forall x\, \big(\lambda(x) \in J\big) \wedge \bigwedge_{b\in J} \exists x\big((\neg\exists y\, (\lambda(x) = \lambda(y) \wedge x < y)) \wedge \lambda(x) = b \wedge x \in X_{f_b}\big)\Big).$$

Since the formula ψ describes an accepting run of the automaton for Σ-dags, we get the statement of the theorem. □

Note that the proof of the theorem above makes use of the finiteness of the sets of local states Q_a in a Σ-ACA. The first language from Example 2.1.5 shows that this finiteness is necessary for the theorem to hold: The language given there can be accepted by a monotone ACM but it is not regular and therefore not monadically axiomatizable. Furthermore, Example 2.1.6 shows that the converse of the theorem does not hold: There, we presented a language that is elementary axiomatizable, but not acceptable by a monotone ACM and can therefore in particular not be accepted by a Σ-ACA.

5.2 (Σ, k)-dags

Theorem 4.1.10 in particular implies that the set of recognizable Σ-dag-languages is not closed under complementation. Hence, there are monadically axiomatizable languages that cannot be accepted by any Σ-ACA. This section is devoted to the class of (Σ, k)-dags that we introduce next where the expressive power of Σ-ACAs and MSO coincide. The results presented here were originally shown for Hasse-diagrams in [Kus98]. Here, the presentation follows [DGK00] and is in addition extended to (Σ, k)-dags.

Let $t = (V, E, \lambda)$ be a Σ-dag. Furthermore, let k be a positive integer and $C_\ell \subseteq V$ for $1 \le \ell \le k$. We call the tuple $(C_1, C_2, \ldots, C_k)$ a *k-chain covering* of t if

1. C_ℓ is a chain with respect to the partial order $E^\star$ for $\ell = 1, 2, \ldots, k$,
2. $V = \bigcup_{\ell\in[k]} C_\ell$ and
3. for any $(x, y) \in E$, there exists $\ell \in [k]$ with $x, y \in C_\ell$ and there is no element of C_ℓ properly between x and y (i.e. $xE^\star zE^+y$ and $z \in C_\ell$ imply $x = z$).

The Σ-dag t is a *(Σ, k)-dag* if it has a k-chain covering. Let $\mathbb{D}_k$ denote the set of all (Σ, k)-dags.

Example 2.1.6 (continued) Consider the first Σ-dag in Figure 2.4. It can be covered by the chains $C_i = \{a_1, a_2, \ldots, a_i, b_i, b_{i+1}, b_{i+2}, \ldots, b_8\}$ for $1 \leq i \leq 8$. Hence it is a $(\Sigma, 8)$-dag. The reader may check that it is not possible to cover it by fewer chains, i.e. that it is not a (Σ, k)-dag for $k < 8$. Recall that the set L cannot be accepted by a Σ-ACA. Later (Theorem 5.2.9) we will see that the reason for this is that L is not contained in $\mathbb{D}_k$ for any $k \in \mathbb{N}$.

Example 5.2.1 Let (Σ, D) be some trace alphabet and $(V, \leq, \lambda) \in \mathbb{M}(\Sigma, D)$. Then $(V, \leq, \lambda)$ is a pomset without autoconcurrency. Hence the Hasse-diagram $t = \text{Ha}(V, \leq, \lambda)$ of this trace is a Σ-dag. Even more, it is a (Σ, k)-dag with $k = |D|$: For $(a, b) \in D$, let $C_{a,b} = \lambda^{-1}(a) \cup \lambda^{-1}(b) \subseteq V$. Since a and b are dependent, this set is a chain. Now let $x, y \in V$ with $x \longrightarrow\!\!< y$. Then $\lambda(x)$ and $\lambda(y)$ are dependent, i.e. x and y belong to some chain $C_{a,b}$ with $(a, b) \in D$.

As explained above, we want to show that the expressive power of Σ-dags and of MSO relative to the class of (Σ, k)-dags coincide. Opposite to this statement, the following proposition shows that this does not hold for deterministic Σ-ACAs.

Proposition 5.2.2 *Let $k \in \mathbb{N}$ with $k > 1$ and let the alphabet Σ contain at least two letters. Then there exists a set of (Σ, k)-dags that is monadically axiomatizable relative to $\mathbb{D}_k$, but not acceptable by any deterministic Σ-ACA.*

Proof. It suffices to prove the statement for $k = 2$, and $\Sigma = \{a, b\}$. So, let L consist of all (Σ, k)-dags (V, E, λ) over Σ that have a largest (with respect to $E^\star$) vertex. This language is trivially axiomatizable in MSO relative to $\mathbb{D}_k$.

We show that there is no deterministic Σ-ACA $\mathcal{A}$ accepting among the (Σ, k)-dags all those that have a largest vertex: By contradiction, assume $\mathcal{A}$ is such a Σ-ACA. Let $\ell = |Q_a| + 2$ and consider the (Σ, k)-dag $t = (V, E, \lambda)$ with vertex set $V = \{a_i \mid i = 1, 2, \ldots, \ell\} \cup \{b_1\}$, $a_1 E a_2 \ldots E a_\ell E b_1$ and with the canonical labeling λ with $\lambda(a_i) = a$ and $\lambda(b_1) = b$. Then $t \in L$. Hence there is a successful run r of $\mathcal{A}$ on t. Since $\ell > |Q_a| + 1$, there are $i < j < \ell$ such that $r(a_i) = r(a_j)$. Now consider the (Σ, k)-dags t_1 and t_2 with $V_1 = V_2 = \{a_\ell \mid \ell = 1, 2, \ldots, j\} \cup \{b_1\}$ and the canonical labeling. The edge relations are defined by $a_1 E_1 a_2 E_1 a_3 \ldots E_1 a_j E_1 b_1$ (i.e. $E_1^\star$ is a linear ordering with largest element b_1) and $a_1 E_2 a_2 E_2 a_3 \ldots E_2 a_j$ and $a_i E_2 b_1$ (i.e. in $E_2^\star$, the a-labeled elements are linearly ordered, but the maximal element b_1 covers a_i and is not the largest element of $(V_2, E_2^\star)$). Since $t_1 \in L$, there is a successful run r_1 of $\mathcal{A}$ on t_1. Since $\mathcal{A}$ is deterministic, we have $r_1(a_\ell) = r(a_\ell)$ for $\ell \leq j$. This implies $r_1(a_i) = r_1(a_j)$ since the equality holds for the run r. Hence r_1 is a run on t_2, too. The global final state of r_1 considered on t_1 equals that of r_1 considered on t_2. Hence t_2 is accepted by $\mathcal{A}$, contradicting our assumption since t_2 does not have any largest vertex. $\square$

Thus, differently from traces, for (Σ, k)-dags the deterministic ACAs are strictly weaker in expressive power than the monadic second order logic. The aim of the following considerations is to show that nondeterministic ACAs have the same expressive power as monadic second order logic in the class of (Σ, k)-dags for any $k \in \mathbb{N}$. First, we define k-chain mappings. Later, we will see that a Σ-dag admits a k-chain mapping iff it is a (Σ, k)-dag.

Definition 5.2.3 Let $t = (V, E, \lambda)$ be a Σ-dag, $k \in \mathbb{N}$ and $\Lambda : V \to (2^{[k]} \setminus \{\emptyset\})$. The function Λ is a *k-chain mapping* if

(1) for all minimal vertices $x, y \in V$, if $x \neq y$ then $\Lambda(x) \cap \Lambda(y) = \emptyset$,

(2) for all non minimal vertices $y \in V$ and $\ell \in \Lambda(y)$, there exists $x \in V$ with $(x, y) \in E$ and $\ell \in \Lambda(x)$,

(3) for all vertices $x \in V$ that are not maximal and for all $\ell \in \Lambda(x)$, the set $\{y \in V \mid (x, y) \in E, \ell \in \Lambda(y)\}$ is empty or has a least element, and

(4) for all $(x, y) \in E$, there is $\ell \in \Lambda(x) \cap \Lambda(y)$ such that for any $z \in V$ with xE^+zE^+y it holds $\ell \notin \Lambda(z)$.

The following lemma relates k-chain mappings and k-chain coverings thereby justifying the name k-chain mapping.

Lemma 5.2.4 *Let $t = (V, E, \lambda)$ be a Σ-dag. Then $t \in \mathbb{D}_k$ iff there exists a k-chain mapping. In particular, if Λ is a k-chain mapping of t and $\ell \in [k]$, then the set $\Lambda^{-1}(\ell) = \{x \in V \mid \ell \in \Lambda(x)\}$ is a chain with respect to $E^\star$ and $(\Lambda^{-1}(\ell))_{\ell \in [k]}$ is a k-chain covering. Conversely, if $(C_\ell)_{\ell \in [k]}$ is a maximal k-chain covering, then $\Lambda(x) = \{\ell \in [k] \mid x \in C_\ell\}$ defines a k-chain mapping.*

Proof. Let $t \in \mathbb{D}_k$. Then there exists a k-chain covering $(C_\ell)_{\ell \in [k]}$ of t. We may assume that the chain covering $(C_\ell)_{\ell \in [k]}$ is maximal with respect to the componentwise inclusion (i.e. incorporating any vertex newly into one of the chains C_ℓ destroys its property to be a k-chain covering). Now define $\Lambda(x) := \{\ell \in [k] \mid x \in C_\ell\}$. Then $\Lambda : V \to (2^{[k]} \setminus \{\emptyset\})$ since $V = \bigcup_{\ell \in [k]} C_\ell$. Since C_ℓ is a chain for each $\ell \in [k]$, any two different minimal elements of t belong to disjoint sets of chains. Hence the first property of Definition 5.2.3 is satisfied. Now let $y \in V$ be non minimal and $\ell \in \Lambda(y)$. Since the k-chain covering $(C_\ell)_{\ell \in [k]}$ is maximal, there exists $x \in V$ with $(x, y) \in E$ and $\ell \in \Lambda(x)$. Hence, the second requirement is satisfied. The targets in C_ℓ of edges that originate in a nonmaximal vertex x are linearly ordered. Hence this set admits a least element as required by the third condition. If $(x, y) \in E$, there exists $\ell \in [k]$ such that $x, y \in C_\ell$ and no element of C_ℓ is properly between x and y. Hence $\ell \in \Lambda(x) \cap \Lambda(y)$ and for any z properly between x and y we have $\ell \notin \Lambda(z)$. Thus, we proved the last statement of Definition 5.2.3.

Conversely, let Λ be a k-chain mapping of the Σ-dag t. For $\ell \in [k]$, define $C_\ell := \{x \in V \mid \ell \in \Lambda(x)\}$. Since $\Lambda(x) \neq \emptyset$ for all $x \in V$, we get $V = \bigcup_{\ell\in[k]} C_\ell$. By the last property for Λ, for any $(x, y) \in E$ there exists $\ell \in [k]$ with $x, y \in C_\ell$ such that no element of C_ℓ lies properly between x and y. It remains to show that C_ℓ is a chain for any ℓ: Let $x, y \in C_\ell$. By the second property of Λ, there exists a sequence $x_0, x_1, \dots x_m = x$ of elements of C_ℓ with x_0 minimal in $(V, E^\star)$ and $(x_i, x_{i+1}) \in E$. We can even assume that x_{i+1} is the least element of C_ℓ above x_i such that $(x_i, x_{i+1}) \in E$. Similarly, there exist elements $y_0, y_1, \dots y_n = y$ of C_ℓ with y_0 minimal in $(V, E^\star)$ and $(y_j, y_{j+1}) \in E$ such that y_{j+1} is the least element of C_ℓ above y_j with $(y_j, y_{j+1}) \in E$. Now let $m \leq n$. By the first property of Λ, $x_0 = y_0$. Let $0 \leq i < m$ be such that $x_i = y_i$. This element is the source of edges going to x_{i+1} and to y_{i+1}. Since we chose x_{i+1} and y_{i+1} minimal in C_ℓ above $x_i = y_i$ with $(x_i, x_{i+1}) \in E$ and $(y_i, y_{i+1}) \in E$, we obtain $x_{i+1} = y_{i+1}$. This shows that $(x, y) \in E^\star$, i.e. C_ℓ is a chain. □

Next we construct an ACA $\mathcal{A}_k$ that accepts a Σ-dag iff it is a (Σ, k)-dag. This ACA will be used later to relabel (Σ, k)-dags into traces.

Recall that $\text{part}([k], \Sigma)$ is the set of *partial* functions g from $[k]$ to Σ with $\text{dom}(g) \neq \emptyset$. We write $\text{part}(k, \Sigma)$ for this set $\text{part}([k], \Sigma)$. For a partial function $f \in \text{part}(k, \Sigma)$, we first define an ACA $\mathcal{A}_k(f)$ whose local states are partial functions in $\text{part}(k, \Sigma)$. Intuitively, a node x of some simple (Σ, k)-dag t will be labeled by the partial function g in some run of $\mathcal{A}_k(f)$ if $\text{dom}(g)$ is the set of chains C_ℓ going through x and for all $\ell \in \text{dom}(g)$, $g(\ell)$ is the next action for the chain ℓ. The partial mapping f is in some sense the initial state of the automaton $\mathcal{A}_k(f)$: $f(\ell) = a$ iff the chain ℓ starts with an action a. As we will see, runs of this automaton correspond to k-chain mappings.

More precisely, the ACA $\mathcal{A}_k(f)$ is defined as follows: The set of local states (common for all processes) is $Q = \text{part}(k, \Sigma)$. For $a \in \Sigma$, let $\delta_{a,\emptyset}$ consist of all nonempty partial functions $g \in Q$ with $\text{dom}(g) = f^{-1}(a)$. For $\emptyset \neq J \subseteq \Sigma$ and $g_b \in Q$ for $b \in J$, we let $\delta_{a,J}((g_b)_{b\in J})$ be the set of all nonempty partial functions $g \in Q$ such that

1. for $b \in J$ there exists $\ell \in \text{dom}(g)$ with $g_b(\ell) = a$ and

2. for $\ell \in \text{dom}(g)$ there exists $b \in J$ with $g_b(\ell) = a$.

Finally, all tuples of states are accepting. Let $\mathcal{A}_k$ denote the disjoint union of the automata $\mathcal{A}_k(f)$ for all partial functions $f \in \text{part}(k, \Sigma)$. Note that not all runs of $\mathcal{A}_k$ are successful, only those that lie completely inside $\mathcal{A}_k(f)$ for some $f \in \text{part}(k, \Sigma)$ are. This can be easily checked by considering the final global state.

The following lemma shows that the k-chain mappings Λ on a Σ-dag t coincide precisely with the mappings $\text{dom} \circ r : V \to 2^{[k]}$ where r is a successful run of the automaton $\mathcal{A}_k$ constructed above.

Lemma 5.2.5 *For $k \in \mathbb{N}$ and $t = (V, E, \lambda) \in \mathbb{D}$, we have:*

1. *for any successful run r of $\mathcal{A}_k$ on t, the mapping* $\mathrm{dom} \circ r : V \to 2^{[k]} \setminus \{\emptyset\}$ *is a k-chain mapping.*

2. *For any k-chain mapping Λ on t, there exists a successful run r of $\mathcal{A}_k$ on t such that* $\Lambda = \mathrm{dom} \circ r$.

Proof. 1. Let $r : V \to \mathrm{part}(k, \Sigma)$ be a successful run of $\mathcal{A}_k$ on t and let $\Lambda = \mathrm{dom} \circ r$. There exists a partial function $f \in \mathrm{part}(k, \Sigma)$ such that r is a run of $\mathcal{A}_k(f)$. Now let $x, y \in V$ be minimal and distinct. Then $r(x) \in \delta_{\lambda(x),\emptyset}$, and therefore $\mathrm{dom} \circ r(x) = f^{-1}(\lambda(x))$. Similarly, $\mathrm{dom} \circ r(y) = f^{-1}(\lambda(y))$. Since x and y are incomparable with respect to $E^\star$, $\lambda(x) \neq \lambda(y)$. Hence $\Lambda(x)$ and $\Lambda(y)$ are disjoint. Thus we showed the first condition of Definition 5.2.3.

Now, let $x \in V$ be non minimal. For $b \in \mathrm{R}(x)$, there exists a unique vertex $x_b \in V$ with $(x_b, x) \in E$ and $\lambda(x_b) = b$. Let also $g_b = r(x_b)$ and $g = r(x)$. Since r satisfies the run condition of $\mathcal{A}(f)$ at x, we have $g \in \delta_{\lambda(x),\mathrm{R}(x)}((g_b)_{b\in\mathrm{R}(x)})$. Now we deduce that for all $\ell \in \Lambda(x) = \mathrm{dom}(g)$, there exists $b \in \mathrm{R}(x)$ with $\ell \in \mathrm{dom}(g_b) = \Lambda(x_b)$ showing Definition 5.2.3 (2). Next, we show Definition 5.2.3 (4) for the edge (x_b, x): Since $g \in \delta_{\lambda(x),\mathrm{R}(x)}((g_b)_{b\in\mathrm{R}(x)})$, there is $\ell \in \mathrm{dom}(g) \cap \mathrm{dom}(g_b) = \Lambda(x) \cap \Lambda(x_b)$ such that $r(x_b)(\ell) = \lambda(x)$. Now assume $x_b E z E^\star x$ with $\ell \in \Lambda(x)$. Then $x_b = \partial_b(x)$. Since r is a run of $\mathcal{A}_k(f)$, we obtain $\lambda(z) = r(x_b)(\ell) = \lambda(x)$. This shows that z and x are targets of edges that originate in x_b, and that they carry the same label $\lambda(x)$. Hence they are equal, i.e. there is no element z properly between x_b and x such that $\ell \in \Lambda(z)$. Thus, Definition 5.2.3 (4) holds. To show Definition 5.2.3(3), let $x \in V$ and $\ell \in \Lambda(x)$ such that the set $\{y \in V \mid (x, y) \in E \text{ and } \ell \in \Lambda(x)\}$ is not empty. Note that this set is a subset of the chain C_ℓ. Hence it has a least element.

2. Assume now that Λ is a k-chain mapping. We will construct a successful run r of $\mathcal{A}_k$ such that $\mathrm{dom} \circ r = \Lambda$. Let $x \in V$. Indeed, the domain of the partial function $r(x) \in \mathrm{part}(k, \Sigma)$ will be $\Lambda(x)$. Now, for all $\ell \in \mathrm{dom}(r(x)) = \Lambda(x)$, if there is $y \in V$ with $(x, y) \in E$ and $\ell \in \Lambda(y)$, then, by Definition 5.2.3 (3), there is a least such y. If such a vertex y exists then we set $r(x)(\ell) = \lambda(y)$ and otherwise we set $r(x)(\ell) = a$ for some $a \in \Sigma$ (in this last case, we can give any value since it will never be used).

Let $f \in \mathrm{part}(k, \Sigma)$ be the partial function defined by $\ell \in \mathrm{dom}(f)$ iff there exists a minimal vertex $x \in V$ with $\ell \in \Lambda(x)$ and in this case we set $f(\ell) = \lambda(x)$. Note that f is well-defined thanks to Definition 5.2.3 (1).

We show that indeed r is a run of $\mathcal{A}_k(f)$: Clearly, if $x \in V$ is minimal then we have $\mathrm{dom}(r(x)) = \Lambda(x) = f^{-1}(\lambda(x))$ as required by the initial transitions of $\mathcal{A}_k(f)$.

Now, let $x \in V$ be non minimal. For all $b \in \mathrm{R}(x)$, let $(x_b, x) \in E$ be such that $\lambda(x_b) = b$. We will show that $r(x) \in \delta_{\lambda(x),\mathrm{R}(x)}(r(x_b)_{b\in\mathrm{R}(x)})$. First, for all $b \in \mathrm{R}(x)$, by Definition 5.2.3 (4), there exists $\ell \in \Lambda(x) \cap \Lambda(x_b) = \mathrm{dom}(r(x)) \cap \mathrm{dom}(r(x_b))$ such that no element z with $\ell \in \Lambda(z)$ lies properly between x_b and x. By the construction of $r(x_b)$, it follows that $r(x_b)(\ell) = \lambda(x)$. Second, for $\ell \in \mathrm{dom}(r(x)) = \Lambda(x)$, there exists $b \in \mathrm{R}(x)$ with $\ell \in \Lambda(x_b) = \mathrm{dom}(r(x_b))$ by Definition 5.2.3 (2). By definition of $r(x_b)$, it follows that $r(x_b)(\ell) = \lambda(x)$. Thus we have shown that r is a run of $\mathcal{A}_k(f)$ which concludes the proof. □

As an immediate consequence of the lemma above and Lemma 5.2.4, we obtain

Corollary 5.2.6 *For $k \in \mathbb{N}$, we have $L(\mathcal{A}_k) = \mathbb{D}_k$.* □

Now we define a trace alphabet (Γ, D) as follows: Let

$$\Gamma := \Sigma \times (2^{[k]} \setminus \{\emptyset\}).$$

The dependence relation D is defined by

$$D = \{((a, M), (b, N)) \mid M \cap N \neq \emptyset \text{ or } a = b\}.$$

This binary relation on Γ is obviously reflexive and symmetric. Thus (Γ, D) is indeed a dependence alphabet. Let $\mathbb{M}(\Gamma, D)$ denote the trace monoid over (Γ, D). Now let $t = (V, \leq, \lambda_\Gamma)$ be a trace over (Γ, D). From this trace, we define a Σ-dag as follows: For $x, y \in V$ let $(x, y) \in E$ iff there exists $\ell \in \pi_2 \circ \lambda_\Gamma(x) \cap \pi_2 \circ \lambda_\Gamma(y)$ such that

$$x = \max\{w < y \mid \ell \in \pi_2 \circ \lambda_\Gamma(w)\}.$$

Now let $\Pi(V, \leq, \lambda) = (V, E, \pi_1 \circ \lambda_\Gamma)$.

For an arbitrary trace $t \in \mathbb{M}(\Gamma, D)$, $\Pi(t)$ is a directed acyclic graph whose vertices are labeled by elements from Σ. Let $\mathbb{M}'$ denote the set of all traces $t \in \mathbb{M}(\Gamma, D)$ such that $\Pi(t) \in \mathbb{D}_k$, i.e. that are mapped to a (Σ, k)-dag by the mapping Π. Note that the relation E defined above is monadically definable in $(V, \leq, \lambda_\Gamma)$. Since in addition the set of (Σ, k)-dags is monadically axiomatizable relative to all Σ-labeled dags, the set $\mathbb{M}'$ is axiomatizable relative to $\mathbb{M}(\Gamma, D)$.

Next, we define the "inverse" of Π: Let $t = (V, E, \lambda)$ be a (Σ, k)-dag. Then there exists a maximal k-chain covering $(C_\ell)_{\ell \in [k]}$. For $y \in V$, define

$$\lambda_\Gamma(y) := (\lambda(y), \{\ell \in [k] \mid y \in C_\ell\}).$$

The following lemma in particular implies that any (Σ, k)-dag is the image under Π of some trace from $\mathbb{M}'$, i.e. $\Pi(\mathbb{M}') = \mathbb{D}_k$.

Lemma 5.2.7 *Let $t = (V, E, \lambda)$ be a (Σ, k)-dag and let $(C_i)_{i \in [k]}$ be a maximal k-chain covering of t. Let $\lambda_\Gamma(x) = (\lambda(x), \{\ell \in [k] \mid x \in C_\ell\})$ for $x \in V$. Then $\Pi(V, E^\star, \lambda_\Gamma) = t$ and $(V, E^\star, \lambda_\Gamma) \in \mathbb{M}'$.*

Proof. Let $\leq$ denote the partial order $E^\star$. First we show that $(V, \leq, \lambda_\Gamma)$ is a trace from $\mathbb{M}(\Gamma, D)$: Let $x, y \in V$ with $x \longrightarrow\!\!\!< y$ (with respect to the partial order $\leq$). Then $(x, y) \in E$. Since $(C_i)_{i\in[k]}$ is a k-chain covering, there exists $\ell \in [k]$ with $x, y \in C_\ell$. Hence $\ell \in \pi_2 \circ \lambda_\Gamma(x) \cap \pi_2 \circ \lambda_\Gamma(y)$ implying $(\lambda_\Gamma(x), \lambda_\Gamma(y)) \in D$. Now let $x, y \in V$ be incomparable. Since C_i is a chain with respect to $\leq$ for $1 \leq i \leq k$, we get $\emptyset = \pi_2 \circ \lambda_\Gamma(x) \cap \pi_2 \circ \lambda_\Gamma(y)$. Since (V, E, λ) is a Σ-dag, x and y carry different labels from Σ. Hence we showed $(\lambda_\Gamma(x), \lambda_\Gamma(y)) \notin D$ which concludes the proof that $(V, \leq, \lambda_\Gamma)$ is a trace.

Now let $\Pi(V, E^\star, \lambda_\Gamma) = (V, E', \lambda')$. Then, $\lambda' = \pi_1 \circ \lambda_\Gamma = \lambda$. It remains to show $E = E'$. So let $(x, y) \in E$. Since $(C_\ell)_{\ell\in[k]}$ is a k-chain covering, there exists $\ell \in [k]$ such that $x, y \in C_\ell$ and no $z \in C_\ell$ lies properly between x and y. Hence $x = \max\{w \in C_\ell \mid w < y\}$ implying $x = \max\{w < y \mid \ell \in \pi_2 \circ \lambda_\Gamma(w)\}$. Hence $(x, y) \in E'$.

If, conversely, $(x, y) \in E'$, then there exists $\ell \in \pi_2 \circ \lambda_\Gamma(y)$ such that $x = \max\{w < y \mid \ell \in \pi_2 \circ \lambda_\Gamma(w)\}$. Since $\pi_2 \circ \lambda_\Gamma(x) = \{\ell \in [k] \mid x \in C_\ell\}$, we obtain $x \in C_\ell$. In addition, $x < y$ implies xE^+y. By contradiction, assume $(x, y) \notin E$. Then there exists $z \in V$ with xE^+zE^+y. Since there is no element of C_ℓ properly between x and y, the set $C_\ell \cup \{z\}$ is a chain with respect to $E^\star$. Since $(x, y) \notin E$, the tuple $(C_1, \ldots, C_{\ell-1}, C_\ell \cup \{z\}, C_{\ell+1}, \ldots, C_k)$ is a k-chain covering contradicting our assumption that $(C_\ell)_{\ell\in[k]}$ is maximal. Thus, we showed $(x, y) \in E$. $\square$

Lemma 5.2.8 *Let φ be a sentence of the monadic second order logic over the alphabet Σ. Then there exists a sentence ψ of the monadic second order logic over the alphabet Γ using the binary relation $\leq$ such that*

$$\{\Pi(s) \mid s \in \mathbb{M}(\Gamma, D) \text{ and } s \models \psi\} = \{t \in \mathbb{D}_k \mid t \models \varphi\}.$$

Proof. The sentence φ contains atomic formulas of the form $\lambda(x) = a$ for $a \in \Sigma$ and of the form $(x, y) \in E$. Replace any occurrence of an atomic formula $\lambda(x) = a$ by $\bigvee_{A\in\Gamma,\pi_1(A)=a} \lambda_\Gamma(x) = A$. There is a monadic formula η using the relation $\leq$ and the mapping λ_Γ that states for any two vertices x, y in a trace $t \in \mathbb{M}(\Gamma, D)$ that there exists $\ell \in [k]$ such that $x = \max\{w < y \mid \ell \in \pi_2 \circ \lambda_\Gamma(w)\}$. Replace any subformula of φ of the form $(x, y) \in E$ by $\eta(x, y)$. The result of these replacements is denoted by $\overline{\varphi}$. Note that $\overline{\varphi}$ is a sentence of the monadic second order logic over the alphabet Γ using the relation $\leq$. Now let $s \in \mathbb{M}'$. Then it is easily seen that $s \models \overline{\varphi}$ iff $\Pi(s) \models \varphi$. Furthermore, there is a monadic second order sentence μ axiomatizing $\mathbb{M}'$ relative to $\mathbb{M}(\Gamma, D)$. Thus, we have the required equality for $\psi = \eta \wedge \overline{\varphi}$. $\square$

Before showing that any monadically axiomatizable set of (Σ, k)-dags can be accepted by an ACA, we have to introduce a variant of asynchronous cellular automata. This variant is meant to work on traces from $\mathbb{M}(\Gamma, D)$. Differently from ACAs considered so far, not every letter of Γ has its own sequential process, but some of the processes are collected into one new sequential component. This collection is given by a partition of Γ into dependence cliques: So let $\Gamma_i \subseteq \Gamma$ for $i \in [n]$ be mutually disjoint sets satisfying $\Gamma_i \times \Gamma_i \subseteq D$ (i.e. the letters from Γ_i are mutually dependent). A trace-ACA over $(\Gamma_i)_{i\in[n]}$ is a tuple $\mathcal{A} = ((Q_i)_{i\in[n]}, (\delta_{a,J})_{a\in\Gamma, J\subseteq[n]}, F)$ where

- Q_i is a finite set of local states for process i,
- $\delta_{a,J} : \prod_{j\in J} \to 2^{Q_i}$ is a local transition function with $a \in \Gamma_i$, and
- $F \subseteq \bigcup_{\emptyset\neq J\subseteq[n]} \prod_{j\in J} Q_j$ is a set of final states.

As remarked earlier, these automata will run on traces from $\mathbb{M}(\Gamma, D)$, more precisely, on the Hasse-diagram of a trace. The only difference in the definition of a run for trace-ACAs is that the transition $\delta_{a,J}$ writes into the process i with $a \in \Gamma_i$. Thus the formal definition is an obvious variation of that from page 14. Therefore, we omit it here.

Theorem 5.2.9 *Let φ be a monadic sentence over the alphabet Σ and let $k \in \mathbb{N}$. Then there exists a Σ-ACA $\mathcal{A}$ such that $L(\mathcal{A}) = \{t \in \mathbb{D}_k \mid t \models \varphi\}$.*

Proof. By Lemma 5.2.8, there is a monadically axiomatizable set $L \subseteq \mathbb{M}(\Gamma, D)$ such that $\Pi(L) = \{t \in \mathbb{D}_k \mid t \models \varphi\}$. Hence by [Tho90b, EM96], the set L is recognizable in $\mathbb{M}(\Gamma, D)$.

For $a \in \Sigma$, let $\Gamma_a = \{(a, M) \in \Gamma\}$ denote the set of letters from Γ whose first component equals a. Then Γ_a is a dependence clique in (Γ, D) and the sets Γ_a are mutually disjoint and cover Γ. By an immediate variant of Zielonka's result [Zie87] (cf. also [CMZ93, Die90]), there exists a trace-ACA

$$\mathcal{A}_\varphi = ((Q^\varphi_a)_{a\in\Sigma}, (\delta^\varphi_{(a,M),J})_{(a,M)\in\Gamma, J\subseteq\Sigma}, F^\varphi)$$

over $(\Gamma_a)_{a\in\Sigma}$ that accepts L relative to $\mathbb{M}(\Gamma, D)$.

Furthermore, let $\mathcal{A}_k = ((Q_a)_{a\in\Sigma}, (\delta_{a,J}), F)$ be the ACA constructed above that accepts the set of all (Σ, k)-dags. We define a Σ-ACA $\mathcal{A}' = ((Q'_a)_{a\in\Sigma}, (\delta'_{a,J}), F')$ over the alphabet Σ as follows: $Q'_a = Q_a \times Q^\varphi_a$ and a tuple $(g_b, q_b)_{b\in J}$ belongs to F' iff $(g_b)_{b\in J} \in F$ and $(q_b)_{b\in J} \in F^\varphi$. To define the transition functions, let $\delta'_{a,J}((g_b, q_b)_{b\in J})$ be the set of all pairs (g, q) satisfying $g \in \delta_{a,J}((g_b)_{b\in J})$ and $q \in \delta^\varphi_{(a,M),J}((q_b)_{b\in J})$ with $M = \text{dom}(g)$. Note that a run of the Σ-ACA $\mathcal{A}'$ "contains" a run of $\mathcal{A}_k$. This run "relabels" the (Σ, k)-dag t in consideration into some trace $s \in \Pi^{-1}(t)$ (see Lemmas 5.2.5 and 5.2.7). The trace s is in fact the actual input of the trace-ACA $\mathcal{A}_\varphi$. Therefore, the (Σ, k)-dag t is accepted by $\mathcal{A}'$ iff $s \in \Pi^{-1}(t)$ is accepted by $\mathcal{A}_\varphi$, that is, iff $t \models \varphi$. □

Our methods in particular imply that the monadic theory of $\mathbb{D}_k$ is decidable for any $k \in \mathbb{N}$: Let φ be a monadic sentence. Using Lemma 5.2.8, we can build a monadic sentence ψ that axiomatizes a preimage under Π of the models of φ in $\mathbb{D}_k$. Hence $\neg\psi$ is a tautology iff $\neg\varphi$ is. Since the monadic theory of traces is decidable [EM96], the result follows. There is another, more direct way to prove this decidability: Given $k \in \mathbb{N}$, one can bound the pathwidth (cf. [Bod98] for an overview) of the dags in $\mathbb{D}_k$ by some n. Since $\mathbb{D}_k$ is monadically axiomatizable, and since the monadic theory of the dags of pathwidth at most n is decidable [Cou90], the decidability follows. Anyway, using Theorem 5.1.1, one obtains the following result:

Corollary 5.2.10 *There exist algorithms that solve the following decision problems:*
input: *an alphabet Σ, $k \in \mathbb{N}$ and a Σ-ACA $\mathcal{A}$.*
output: *Is $L(\mathcal{A}) \cap \mathbb{D}_k$ empty?*
Is $L(\mathcal{A})$ contained in $\mathbb{D}_k$?
Does $L(\mathcal{A}_1) \cap \mathbb{D}_k = L(\mathcal{A}_2) \cap \mathbb{D}_k$?

Recall that by Proposition 5.2.2 the expressive power of deterministic Σ-ACAs does not capture that of monadic second order logic relative to $\mathbb{D}_k$. Hence, we get in particular that nondeterministic ACAs are strictly more powerful than deterministic ACAs within the class $\mathbb{D}_k$ for $k \geq 2$ and the same holds for the set of all Σ-dags (which we already knew from Theorem 4.1.10). In this latter case, the set of Σ-ACAs that have an equivalent deterministic Σ-ACA is not recursive (Theorem 4.1.10). It is an open question whether this holds for the class of (Σ, k)-dags, too, i.e. whether there is an algorithm that given a Σ-ACA $\mathcal{A}$ and a positive integer k decides whether there exists a deterministic Σ-ACA $\mathcal{A}_d$ such that $L(\mathcal{A}) \cap \mathbb{D}_k = L(\mathcal{A}_d) \cap \mathbb{D}_k$.

Chapter 6

Other automata models for pomsets

The covering relation of a pomset without autoconcurrency is a Σ-dag. This allows us to speak of the set of pomsets that is accepted by an asychronous cellular automaton: A pomset $(V, \leq, \lambda)$ is accepted by the Σ-ACA $\mathcal{A}$ iff its Hasse-diagram $(V, -\!\!<, \lambda)$ belongs to $L(\mathcal{A})$. Actually, this was the original intention when asynchronous cellular automata were generalized from dependence graphs to more general structures in [DG96] (cf. also [Kus98, DGK00]).

For pomsets, other automata models have been considered in the literature. In particular, Arnold considered P-asynchronous automata [Arn91] and Lodaya and Weil dealt with branching automata [LW98a, LW98b, LW00]. The primary aim of this chapter is to compare the expressive power of these automata with the expressive power of our Σ-ACAs. Somewhat as a byproduct, we obtain results on the relation between these other automata models and monadic second order logic.

6.1 Branching automata by Lodaya and Weil

In several papers, Lodaya and Weil considered branching automata and proved results analogous to Kleene's and to Myhill-Nerod's Theorems [LW98b, LW98a, LW00]. Their automata work on so call series-parallel pomsets, sp-pomsets for short, defined as follows: A labeled partial order $(V, \leq, \lambda)$ is a *sp-pomset* if the partially ordered set $(N, \leq_N)$ cannot be embedded into $(V, \leq)$ (cf. Figure 6.1). To give an alternative description of sp-pomsets (that also explains the name) we need some more notation: Let $t_1 = (V_1, \leq_1, \lambda_1)$ and $t_2 = (V_2, \leq_2, \lambda_2)$ be labeled partial orders with $V_1 \cap V_2 = \emptyset$. The *serial product* $t_1 \cdot t_2$ of them is the labeled partial order

$$(V_1 \cup V_2, \leq_1 \cup \leq_2 \cup V_1 \times V_2, \lambda_1 \cup \lambda_2).$$

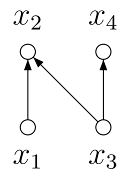


Figure 6.1: The partially ordered set $(N, \leq)$

Thus, in $t_1 \cdot t_2$, the pomset t_2 is put on top of the pomset t_1. On the contrary, the *parallel product* $t_1 \parallel t_2$ is defined to be

$$(V_1 \cup V_2, \leq_1 \cup \leq_2, \lambda_1 \cup \lambda_2),$$

i.e. here the two partial orders are set side by side. Now it is a result in the folklore of order theory that a partially ordered set is series-parallel iff it can be constructed from the one-point partial orders by the application of the operations $\cdot$ and $\parallel$. In other words, the set of all sp-pomsets $\mathrm{SP}(\Sigma)$ over the alphabet Σ is the least class of Σ-labeled partial orders containing the one-point pomsets that is closed under the application of the serial product $\cdot$ and the parallel product $\parallel$. Hence the set $\mathrm{SP}(\Sigma)$ is equipped with two operations, the serial and the parallel product. They can naturally be extended to sets of sp-pomsets by

$$\begin{aligned} S \cdot T &:= \{s \cdot t \mid s \in S, t \in T\} \text{ and} \\ S \parallel T &:= \{s \parallel t \mid s \in S, t \in T\} \end{aligned}$$

for $S, T \subseteq \mathrm{SP}(\Sigma)$. One can even consider the iteration of the two operations defined by

$$\begin{aligned} S^\star &:= \{s_1 \cdot s_2 \cdot s_3 \cdots s_n \mid n \geq 1, s_i \in S\} \text{ and} \\ S^\oplus &:= \{s_1 \parallel s_2 \parallel s_3 \cdots \parallel s_n \mid n \geq 1, s_i \in S\} \end{aligned}$$

for $S \subseteq \mathrm{SP}(\Sigma)$. A set $S \subseteq \mathrm{SP}(\Sigma)$ is *rational* if it can be constructed from the finite subsets of $\mathrm{SP}(\Sigma)$ by the operations $\cup$, $\cdot$, $\parallel$, *, and $^\oplus$. It is *weakly rational* if the operation $^\oplus$ is applied to languages of the form $K \cdot L$, only. Finally it is *series-rational* if it can be constructed from the finite subsets of $\mathrm{SP}(\Sigma)$ by the operations $\cup$, $\cdot$, $\parallel$, and * (i.e. without the parallel iteration). Clearly, any series-rational language is weakly rational and any weakly rational language is rational. These two implications cannot be inverted for $(a \parallel a)^\oplus$ is rational and not weakly rational and $a^\oplus$ is weakly rational but not series-rational. Since in the construction of series-rational languages the parallel iteration cannot appear, for any series-rational language S there exists an $n \in \mathbb{N}$ with $w(s) \leq n$ for any $s \in S$, i.e. any series-rational language is *width-bounded*.

Example 6.1.1 Let a denote the a-labeled one-point pomset for any $a \in \Sigma$. Now let $a, b \in \Sigma$. Then $a \parallel b \parallel b$ is an antichain of three elements two of which are labeled by b and the third by a. Furthermore, $S = (a \parallel a)^{\oplus}$ is a rational language. It consists of all antichains of an even number of a-labeled vertices. In particular, S is not width-bounded and therefore not series-rational. Furthermore, S cannot be monadically axiomatizable since it is impossible to axiomatize the finite sets of even size in this logic relative to all finite sets.

The example above showed that not every rational sp-language can be monadically axiomatized. It is not clear which additional features should be adjoint to MSO to obtain precisely the expressive power of rational sp-languages. Even though there are rational languages that cannot be monadically axiomatized, this does not occur in the context of weakly rational languages:

Proposition 6.1.2 *Let S be a weakly rational language. Then there exists a monadic sentence σ such that $S = \{t \in \mathrm{SP}(\Sigma) \mid t \models \sigma\}$.*

Proof. Clearly, any finite set of sp-pomsets can be monadically axiomatized. Now let S and T be two sets of sp-pomsets axiomatized by the monadic sentences σ and τ, respectively. Then $S \cup T$ is axiomatized by $\sigma \vee \tau$. The set $S \parallel T$ consists of all sp-pomsets satisfying

$$\exists X (\forall x \forall y (x \in X \wedge y \notin X \to x \parallel y) \wedge \sigma \upharpoonright X \wedge \tau \upharpoonright X^{co})$$

where $\sigma \upharpoonright X$ is the restriction of σ to the set X and $\tau \upharpoonright X^{co}$ that of τ to the complement of X. Similarly, $S \cdot T$ is axiomatized by

$$\exists X (\forall x \forall y (x \in X \wedge y \notin X \to x < y) \wedge \sigma \upharpoonright X \wedge \tau \upharpoonright X^{co}).$$

Next we show that $S^{\star}$ can be described by a monadic sentence: The idea of a sentence axiomatizing $S^{\star}$ is to color the vertices of an sp-pomset s by two colors such that the coloring corresponds to a factorization in factors $s = s_1 \cdot s_2 \cdot s_3 \cdots s_n$ where every factor s_i belongs to S. The identification of the S-factors will be provided by the property of being a maximal convex one-colored set. More formally, we define $\varphi = \exists X \exists Y (\varphi_1 \wedge \varphi_X \wedge \varphi_Y)$ where φ_1 asserts that X and Y form a partition of the set of vertices. The formula φ_X states that the maximal subsets of X that are convex satisfy σ, i.e.

$$\begin{aligned} \varphi_X \quad = \forall Z(\quad & Z \subseteq X \wedge Z \text{ is convex} \wedge \\ & \forall Z'(Z \subseteq Z' \subseteq X \wedge Z' \text{ is convex} \to Z = Z') \\ & \to \sigma \upharpoonright Z) \end{aligned}$$

and the formula φ_Y is defined similarly with Y taking the place of X.

Finally, we have to deal with the parallel iteration $\oplus$. Recall that it is applied to languages of the form $S = S_1 \cdot S_2$, only. Hence, any element of the iterated language S (as a partial order) is connected. Thus, the informal sentence

$$\varphi = \forall Z(Z \text{ is connected } \to \sigma \restriction Z)$$

axiomatizes $S^{\oplus}$. Since in monadic second order logic the transitive closure of the comparability relation $\leq \cup \geq$ can be defined, one can express that a set is connected. Hence, $S^{\oplus}$ can be monadically axiomatized whenever S is the product of two monadically axiomatizable languages. □

Thus, rational languages are not necessarily monadically axiomatizable, but weakly rational languages and therefore series-rational languages are monadically axiomatizable. On the other hand, the set of all sp-pomsets is trivially axiomatizable but not weakly rational. At the end of this section, we will see that the only missing property for being series-rational is width-bounded, i.e. that a width-bounded set of sp-pomsets is series-rational iff it is monadically axiomatizable. This answers an open question from [LW00]. As a tool in this proof, we use (which is no surprise in the context of this Habilitationsschrift) Σ-ACAs and branching automata that we introduce next:

Definition 6.1.3 A *branching automaton* is a tuple $\mathcal{B} = (S, T_s, T_f, T_j, I, A)$ where
S is a finite set of states,
I and A are subsets of S of initial and accepting states, respectively,
$T_s \subseteq S \times \Sigma \times S$ is the set of *sequential transitions*,
$T_f \subseteq S \times (S \times S)$ is the set of *fork transitions*, and
$T_j \subseteq (S \times S) \times S$ is the set of *join transitions*.

Branching automata were introduced in [LW98b] in a slightly more general form. But it is easy to see that the definition given here yields the same expressive power. Since branching automata are meant to run on sp-pomsets, their runs can be defined by induction on the construction of sp-pomsets from the one-point pomsets by serial and parallel product. Let $p, r \in S$ be states of the branching automaton $\mathcal{B}$ and let a be the one-point a-labeled pomset. Then there is a run from p to r on a (denoted $p \xrightarrow{a} r$) iff $(p, a, r) \in T_s$. Now let s, t be sp-pomsets. Then there is a run $p \xrightarrow{s \cdot t} r$ iff there exists a state $q \in S$ and runs $p \xrightarrow{s} q$ and $q \xrightarrow{t} r$. Finally, there is a run $p \xrightarrow{s \| t} r$ iff there are states $p_1, p_2, r_1, r_2 \in S$, a fork transition $(p, (p_1, p_2)) \in T_f$, runs $p_1 \xrightarrow{s} r_1$ and $p_2 \xrightarrow{t} r_2$, and a join transition $((r_1, r_2), r) \in T_j$ (this definition is visualized in Figure 6.2, edges that form a fork or a join transition are connected by an angle at their sources or their targets). An sp-pomset s is *accepted* by $\mathcal{B}$ iff there exist $\iota \in I$ and $q \in A$ and a run $\iota \xrightarrow{s} q$. By $L(\mathcal{B})$ we denote the set of sp-pomsets accepted by $\mathcal{B}$. The

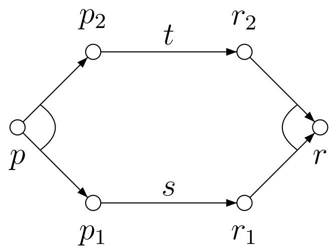


Figure 6.2: A run on $s \parallel t$

concepts of (serial-)rational sets and of accepted language are closely connected as the following theorem shows:

Theorem 6.1.4 ([LW98a, LW00]) *Let S be a set of sp-pomsets. Then S is rational iff it can be accepted by some branching automaton. Moreover, S is serial-rational iff it is width-bounded and can be accepted by some branching automaton.*

In [LW00], Lodaya and Weil give other characterizations of serial-rational languages by algebraic means and by additional requirements on branching automata ("fork-acyclic branching automata"). Since these characterizations are not used in our considerations, the interested reader is referred to the literature.

Remark 6.1.5 Let $\mathrm{SP}_{wa}(\Sigma)$ denote the set of sp-pomsets without autoconcurrency, i.e. the set of sp-pomsets $t = (V, \leq, \lambda)$ for which $\lambda^{-1}(a)$ is a chain for any $a \in \Sigma$. There exists a branching automaton $\mathcal{B}$ with $L(\mathcal{B}) = \mathrm{SP}_{wa}(\Sigma)$.

Proof. Consider the following branching automaton:
the set of states is 2^{Σ},
the sequential transitions are given by $(M, a, M \cup \{a\})$,
the fork transitions are of the form $(M, (M_1, M_2))$ where $M_1 \dot\cup M_2 = M$,
the join transitions are of the form $((M_1, M_2), M)$ where $M_1 \dot\cup M_2 = M$,
the only initial state is $\emptyset$, and any state is accepting.

Inductively, one gets $M \xrightarrow{s} N$ iff $s \in \mathrm{SP}_{wa}(\Sigma)$ and $N = M \cup \lambda(s)$. Hence this branching automaton accepts the sp-pomsets without autoconcurrency. □

6.1.1 From branching automata to ACAs

Let $\mathcal{B}$ be a branching automaton whose accepted language is contained in $\mathrm{SP}_{wa}(\Sigma)$. Next, we want to show that $L(\mathcal{B})$ can be accepted by a Σ-ACA. By Theorem 6.1.4 and Proposition 6.1.2, $L(\mathcal{B})$ is monadically axiomatizable. Thus, it remains to show that an monadically axiomatizable width-bounded language in $\mathrm{SP}_{wa}(\Sigma)$ can be accepted by a Σ-ACA. The following lemma shows that we can use the concept of (Σ, k)-dag that proved suitable when accepting a monadically axiomatized set by a Σ-ACA.

Lemma 6.1.6 *Let* $t \in \mathrm{SP}_{wa}(\Sigma)$. *Then the Hasse-diagram* $\mathrm{Ha}(t)$ *is a* $(\Sigma, |\Sigma|^2)$-*dag.*

Proof. To ease the notions, when speaking of a k-chain covering of the pomset $t = (V, \leq, \lambda)$ we mean a k-chain covering of its Hasse-diagram $\mathrm{Ha}(t)$. A k-chain covering of the Σ-dag t is *saturated* if any vertex $x \in V$ belongs to at least $|\Sigma|$ chains. We show by induction on the construction of the sp-pomset t that it has a saturated $w(t) \cdot |\Sigma|$-chain covering ($w(t)$ it the width of $(V, \leq)$). This is obviously true for the one-point pomset $t = (V, \leq, \lambda)$ setting $C_i = V$ for $1 \leq i \leq |\Sigma|$.

Now let $s = (V_s, \leq_s, \lambda_s)$ and $t = (V_t, \leq_t, \lambda_t)$ be sp-pomsets without autoconcurrency and let $(C_i)_{i \in [w(s) \cdot |\Sigma|]}$ and $(C'_i)_{i \in [w(t) \cdot |\Sigma|]}$ be saturated chain-coverings of s and t, respectively.

Note that $w(s) + w(t) = w(s \parallel t)$ and that

$$\overline{C_i} := \begin{cases} C_i & \text{for } 1 \leq i \leq w(s) \cdot |\Sigma| \\ C'_{i-[w(s)\cdot|\Sigma|} & \text{for } w(s) \cdot |\Sigma| < i \leq (w(s) + w(t)) \cdot |\Sigma| \end{cases}$$

defines a saturated chain covering with $w(s \parallel t) \cdot |\Sigma|$ chains of the parallel product $s \parallel t$.

Now we consider the sequential product. By symmetry, is suffices to consider the case $w(s) \geq w(t)$. Furthermore, we may assume that any chain C_i contains a maximal vertex in s and any chain C'_i contains a minimal vertex in t. Let $\max(s) = \{x_1, x_2, \ldots, x_m\}$ denote the maximal vertices in s. We renumber the chains of the saturated chain covering of s such that $\{C_{i,j} \mid 1 \leq j \leq m_i\}$ for $1 \leq i \leq m$ are those chains that contain x_i. Then $m_i \geq |\Sigma|$ since the chain covering of s is saturated. Similarly, let $\min(t) = \{y_1, y_2, \ldots, y_n\}$ be the minimal vertices of t and let $\{C'_{i,j} \mid 1 \leq j \leq n_i\}$ be the set of chains containing y_i for $1 \leq i \leq n$. Again, $n_i \geq |\Sigma|$. Furthermore, $m \leq w(s)$ and $n \leq w(t)$ since $\max(s)$ and $\min(t)$ are antichains.

To define the chain covering of $s \cdot t$, let $1 \leq i \leq m$ and $1 \leq j \leq n$. Then $i \leq m \leq |\Sigma| \leq n_j$ and similarly $j \leq n \leq |\Sigma| \leq m_i$. Hence the chains $C'_{j,i}$ and $C_{i,j}$ contain y_j and x_i, respectively. Set $\overline{C_{i,j}} := C_{i,j} \cup C'_{j,i}$. Note that $\sum_{i=1}^{m} m_i$ is the number of chains in the chain covering of s, i.e. this sum equals $w(s) \cdot |\Sigma|$. Similarly, $\sum_{i=1}^{n} n_i = w(t) \cdot |\Sigma|$. Furthermore, $\{(i,j) \mid 1 \leq i \leq m, n < j \leq n_i\}$ is the set of indices of chains in s that are not coupled with a chain in t so far. Since $w(s) \geq w((t)$, there is a surjective mapping η from $\{(i,j) \mid 1 \leq i \leq m, n < j \leq n_i\}$ onto $\{(j,i) \mid 1 \leq j \leq n, m < i \leq m_j\}$. Now define $\overline{C_{i,j}} := C_{i,j} \cup C'_{\eta(i,j)}$ for $1 \leq i \leq m$ and $n < j \leq n_i$. Then the chains $\overline{C_{i,j}}$ with $1 \leq i \leq n$ and $1 \leq j \leq n_i$

form a saturated chain covering of $s \cdot t$ with $w(s) \cdot |\Sigma| = w(s \cdot t) \cdot |\Sigma|$ chains. $\square$

Theorem 6.1.7 *Let $\mathcal{B}$ be a branching automaton. Then there exists a Σ-ACA $\mathcal{A}$ such that* $\text{Ha}(L(\mathcal{B}) \cap \text{SP}_{wa}(\Sigma)) = L(\mathcal{A})$.

Proof. By Remark 6.1.5, the set $\text{SP}_{wa}(\Sigma)$ can be accepted by a branching automaton. By [LW00, Theorem 4.6], the set of languages in $\text{SP}_{wa}(\Sigma)$ that can be accepted by a branching automaton is closed under intersection. Hence $L = L(\mathcal{B}) \cap \text{SP}_{wa}(\Sigma)$ is acceptable by a branching automaton. In addition, L is width-bounded since any $t \in \text{SP}_{wa}(\Sigma)$ has width at most $|\Sigma|$. Thus, by Theorem 6.1.4, L is serial-rational and therefore monadically axiomatizable relative to all sp-pomsets by Proposition 6.1.2. Hence the set $\text{Ha}(L) = \{\text{Ha}(t) \mid t \in L\}$ is monadically axiomatizable relative to all Σ-dags. Since, by Lemma 6.1.6, $\text{Ha}(L) \subseteq \mathbb{D}_{|\Sigma|^2}$, we can apply Theorem 5.2.9 and obtain the existence of the required Σ-ACA. $\square$

6.1.2 From ACAs to branching automata

Next, we want to show the inverse implication of the theorem above. This proof is direct, i.e. we construct from a Σ-ACA $\mathcal{A} = ((Q_a)_{a\in\Sigma}, (\delta_{a,J})_{a\in\Sigma, J\subseteq\Sigma}, F)$ a branching automaton $\mathcal{B} = (S, T_s, T_f, T_j, I, A)$ such that $\text{Ha}(L(\mathcal{B})) = L(\mathcal{A}) \cap \text{Ha}(\text{SP}_{wa}(\Sigma))$:

- The state space S of $\mathcal{B}$ is given by $S = \bigcup_{J\subseteq\Sigma} \prod_{b\in J} Q_b$. Note that the empty tuple, denoted () is contained in S. Furthermore, for $s = (s_b)_{b\in J} \in S$ we define $\text{dom}(s) = J$.

- For any $s, s' \in S$ and $a \in \Sigma$, we define a sequential transition (s, a, s') if $\text{dom}(s') = \{a\}$ and $s'(a) \in \delta_{a,\text{dom}(s)}(s)$.

- For $s \in S$, there is a fork transition $s \to (s, s)$, i.e. the branching automaton can always duplicate its states and these are the only fork transitions.

- For $s_1, s_2, s \in S$, there is a join-transition $(s_1, s_2) \to s$ if $\text{dom}(s_1)$ and $\text{dom}(s_2)$ are disjoint and nonempty and $s = s_1 \cup s_2$.

- The set of initial states is given by $I = \{()\}$, i.e. the empty tuple is the only initial state.

- The set of final states A equals F.

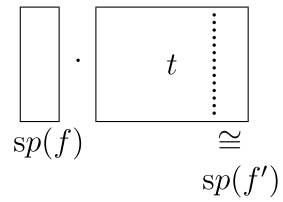


Figure 6.3: cf. Lemma 6.1.8

For a tuple $f \in S$, let sp(f) denote the sp-pomset $(\mathrm{dom}(f), \Delta_{\mathrm{dom}(f)}, \mathrm{id}_{\mathrm{dom}(f)})$, i.e. sp$(f)$ is an antichain of size dom(f) and the nodes are labeled by the elements of dom(f). Now we can show a correspondence between the runs of the branching automaton $\mathcal{B}$ and the Σ-ACA $\mathcal{A}$:

Lemma 6.1.8 *Let $t = (V, \leq, \lambda) \in \mathrm{SP}_{wa}(\Sigma)$ and $f, f' \in S$. Then the following are equivalent:*
(i) There is a run of $\mathcal{B}$ on t from f to f'.
(ii) There is a mapping $r : \mathrm{dom}(f) \,\dot\cup\, V \to Q$ such that
(a) r extends f,
(b) r satisfies the run condition relative to Ha(sp$(f) \cdot t$) *for vertices in V,*
(c) $\mathrm{dom}(f') = \lambda \circ \max(t)$*, and*
(d) $r(\max \lambda^{-1}(b)) = f'(b)$ *for $b \in \mathrm{dom}(f')$.*

To ease the understanding of the proof of this lemma, we first try to visualize statement (ii) (cf. Figure 6.3): The first box denotes sp(f). The nodes in this box are mutually incomparable since sp(f) is an antichain. The second box depicts the sp-pomset t. All of its nodes are larger than all nodes from the first box since we consider the sequential product of these two. The maximal nodes in the sp-pomset t (i.e. those behind the dotted line) correspond to sp(f') by (c). This correspondence is given by $b \mapsto \max(\lambda^{-1}(b))$ for any vertex b in sp(f'). Now the second statement asserts the existence of a map $r : \mathrm{dom}(f)\dot\cup V \to S$ that coincides with f in the first box, with f' in the maximal elements of t, and that satisfies the run-condition for nodes in the big box. Having this visualization in mind, the following proof is straightforward.

Proof. Since we deal with sp-pomsets, we can prove the lemma by induction on the construction of t from the one-point sp-pomsets. So let $a \in \Sigma$ and consider the one-point sp-pomset $t = (\{1\}, \Delta_{\{1\}}, \{(1, a)\})$. If there is a run $f \xrightarrow{t} f'$ of $\mathcal{B}$, there is a corresponding a-labeled transition (f, a, f') in $\mathcal{B}$, i.e. $\mathrm{dom}(f') = \{a\}$ and $f'(a) \in \delta_{a,\mathrm{dom}(f)}(f)$. Now define $r : \mathrm{dom}(f) \cup \{1\} \to S$ by $r \restriction \mathrm{dom}(f) = f$ and $r(1) = f'(a)$. Then (a) is trivial and (b) holds for the vertex 1. Furthermore,

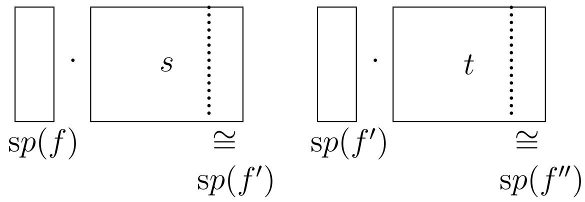


Figure 6.4: The case $s \cdot t$ in the proof of Lemma 6.1.8

$\mathrm{dom}(f') = \{a\} = \{\lambda(1)\} = \lambda \circ \max(t)$ ensures (c). Finally, $r(\max \lambda^{-1}(a)) = r(1) = f'(a)$ finishes the proof of (i)⇒(ii) for one-point pomsets.

Conversely, assume (ii) holds for the one-point pomset t. Then in particular $r \restriction \mathrm{dom}(f) = f$ and $r(1) = f'(a)$. Since the reading domain of 1 in the Hasse diagram $\mathrm{Ha}(\mathrm{sp}(f) \cdot t)$ is $\mathrm{dom}(f)$, and since r satisfies the run-condition for 1, we obtain $r(1) \in \delta_{a,\mathrm{dom}(f)}(f)$. Hence there is an a-labeled transition (f, a, f').

Now we come to the induction argument. So let $s = (V_s, \leq_s, \lambda_s)$ and $t = (V_t, \leq_t, \lambda_t)$ be sp-pomsets without autoconcurrency that satisfy the equivalence of (i) and (ii) for any $f, f' \in S$.

The case $s \cdot t$. Let $f, f'' \in S$ be states of $\mathcal{B}$ and suppose there is a run of $\mathcal{B}$ on $s \cdot t$ from f to f''. Then there exists $f' \in S$ such that $f \xrightarrow{s} f'$ and $f' \xrightarrow{s} f''$. Hence there are mappings r_s and r_t that satisfy (a)-(d). The situation is depicted in Figure 6.4. Note that the sp-pomset $\mathrm{sp}(f) \cdot s \cdot t$ is obtained from $\mathrm{sp}(f) \cdot s$ and $\mathrm{sp}(f') \cdot t$ by identifying a vertex $b \in \mathrm{dom}(f')$ with the maximal b-labeled vertex of s (which belongs to $\max(s)$ by (c)). By (d), we have $r_s(\max \lambda_s^{-1}(b)) = f'(b) = r_t(b)$, i.e. this identification behaves well with respect to the mappings r_s and r_t. Now it is immediate that $r = r_s \cup (r_t \restriction V_t)$ satisfies the conditions (a)-(d) with respect to the sp-pomset $\mathrm{sp}(f) \cdot s \cdot t$, i.e. one obtains (ii).

Conversely, assume (ii) holds, i.e. there is a mapping $r : \mathrm{dom}(f) \cup V_s \cup V_t \to S$ such that r, f, $s \cdot t$ and f'' satisfy (a)-(d). Then define $\mathrm{dom}(f') := \lambda_s \circ \max(s)$ and $f'(b) = r(\max \lambda_s^{-1}(b))$ for $b \in \mathrm{dom}(f')$. Furthermore, let $r_s = r \restriction \mathrm{dom}(f) \cup V_s$. To define the mapping $r_t : \mathrm{dom}(f') \cup V_t$, let $r_t \restriction V_t = r \restriction V_t$ and, for $b \in \mathrm{dom}(f') = \max(s)$, set $r_t(b) = f'(b) = r(\max \lambda_s^{-1}(b))$. Now we have (a)-(d) for r_s, f, s and f'. Hence by the induction hypothesis $f \xrightarrow{s} f'$. Since $\mathrm{dom}(f') = \lambda_s \circ \max(s)$ and $f'(b) = r_s(\max \lambda_s^{-1}(b))$ for $b \in \mathrm{dom}(f')$, the same holds for r_t, f', t and f'' implying $f' \xrightarrow{t} f''$. Thus, $f \xrightarrow{s \cdot t} f''$, i.e. we showed the equivalence of (i) and (ii) for the sequential product $s \cdot t$.

The case $s \parallel t$. Since $s \parallel t$ has no autoconcurrency, the two sp-pomsets carry disjoint labels, i.e. $\lambda_s(V_s) \cap \lambda_t(V_t) = \emptyset$. Now let $f, f' \in S$ be such that $f \xrightarrow{s \parallel t} f'$ in $\mathcal{B}$. Then this run consists of a fork transition $f \to (f, f)$ followed by two parallel runs $f \xrightarrow{s} f_1$ and $f \xrightarrow{t} f_2$ that are joint at the end in a join transition $(f_1, f_2) \to$

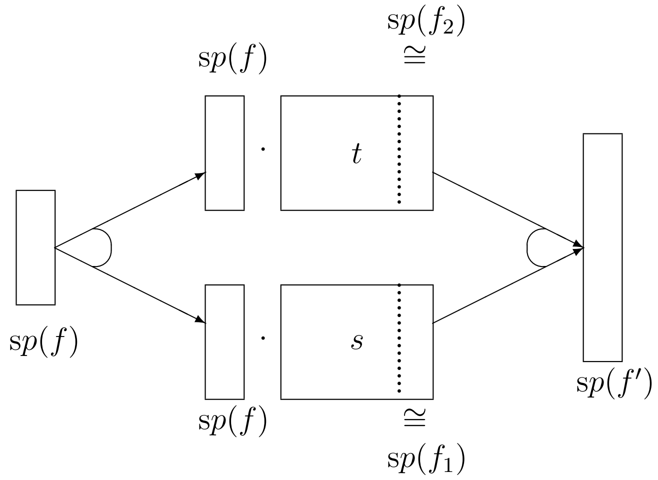


Figure 6.5: The case $s \parallel t$ in the proof of Lemma 6.1.8

f' (cf. Figure 6.5). Applying the induction hypothesis to the two parallel runs, we get two mappings r_s and r_t such that r_s, f, s and f_1 satisfy (a)-(d) and similarly r_t, f, t and f_2. Now define $r := r_s \cup r_t$. Note that $\mathrm{dom}(r_s) \cap \mathrm{dom}(r_t) = \mathrm{dom}(f)$ where r_s and r_t coincide with f. Hence $r : \mathrm{dom}(f) \cup V_s \cup V_t \to S$ is well defined and we obtain (a) and (b) for r, f and $s \parallel t$ immediately. To verify (c), note that $\mathrm{dom}(f') = \mathrm{dom}(f_1) \cup \mathrm{dom}(f_2)$ and therefore $\mathrm{dom}(f') = \lambda \circ \max(s \parallel t)$. Now (d) follows since $\mathrm{dom}(f_1)$ and $\mathrm{dom}(f_2)$ are disjoint. Thus, we showed the implication (i)⇒(ii).

For the other implication assume r, f, $s \parallel t$ and f' satisfy (a)-(d). Then $r \restriction \mathrm{dom}(f) \cup V_s$, f, s and $f' \restriction \lambda_s \circ \max(s) =: f_1$ satisfy (a)-(d). Similarly, we get (a)-(d) for $r \restriction \mathrm{dom}(f) \cup V_t$, f, t and $f' \restriction \lambda_t \circ \max(t) =: f_2$. Hence there are runs $f \xrightarrow{s} f_1$ and $f \xrightarrow{t} f_2$ in $\mathcal{B}$. Starting with the fork transition $f \to (f, f)$, then performing the two runs in parallel and finally joining them by the join transition $(f_1, f_2) \to f'$ gives a run $f \xrightarrow{s\parallel t} f'$ as required. □

Recall that for a run r on $t = (V, E, \lambda) \in \mathbb{D}$ of a Σ-ACA, the states at the maximal a-labeled nodes $(\max \lambda^{-1}(a))_{a \in \lambda(V)}$ decide whether it is successful or not. On the contrary, in the lemma above, we have an assertion on the states at the maximal nodes $\max(t)$, only. Call a run r on t *weakly successful* if the states at the maximal nodes are accepting, i.e. if $(r(x))_{x \in \max(t)} \in F$. Let $L_w(\mathcal{A})$ denote the set of Σ-dags that admit a weakly successful run of $\mathcal{A}$. We show that for any Σ-ACA $\mathcal{A}$ there exists another Σ-ACA $\mathcal{A}'$ such that $L(\mathcal{A}) = L_w(\mathcal{A}')$: The Σ-ACA $\mathcal{A}'$ simulates the run of $\mathcal{A}$ on some Σ-dag t and additionally guesses the maximal a-labeled node of t for each letter a that occurs in t. To do this, along a weakly successful run of $\mathcal{A}'$ on t, any local process a nondeterministically picks one and only one node x with $\lambda(x) = a$. This node sends a signal (End_a, p_a) upwards where p_a is the local state at the guessed node. This signal is forwarded upwards by all transitions. The Σ-ACA $\mathcal{A}'$ stops if an a-labeled node receives the signal End_a since this means that the guessed a-labeled node was not maximal in its chain $\lambda^{-1}(a)$. By reading the states at the maximal nodes from $\max(t)$, we can now recover the final state of each process a and accept or reject the Σ-dag according to the acceptance condition of $\mathcal{A}$.

Theorem 6.1.9 *Let $\mathcal{A}$ be a Σ-ACA. Then there exists a branching automaton $\overline{\mathcal{B}}$ such that* $\mathrm{Ha}(L(\overline{\mathcal{B}})) = L(\mathcal{A}) \cap \mathrm{Ha}(\mathrm{SP}(\Sigma))$.

Proof. By the consideration above, it suffices to show that $L_w(\mathcal{A}) \cap \mathrm{Ha}(\mathrm{SP}(\Sigma))$ can be accepted by a branching automaton $\overline{\mathcal{B}}$. So let $\mathcal{B}$ be the branching automaton we constructed above. Furthermore, let $t = (V, \leq, \lambda) \in \mathrm{SP}_{wa}(\Sigma)$. Then $t \in L(\mathcal{B})$ iff there is a run $() \xrightarrow{t} f$ for some $f \in F$. Note that $sp(()) = \emptyset$. Hence the existence of the run $() \xrightarrow{t} f$ of $\mathcal{B}$ is by Lemma 6.1.8 equivalent to the existence of a run r on $\emptyset \cdot t = t$ of $\mathcal{A}$ satisfying $\mathrm{dom}(f) = \lambda \circ \max(t)$ and $r(\max \lambda^{-1}(b)) = f(b)$ for $b \in \mathrm{dom}(f)$, i.e. to the existence of a weakly successful run of $\mathcal{A}$. Thus, we showed

$$\mathrm{Ha}\left(L(\mathcal{B}) \cap \mathrm{SP}_{wa}(\Sigma)\right) = L_w(\mathcal{A}) \cap \mathrm{Ha}(\mathrm{SP}_{wa}(\Sigma)) = L_w(\mathcal{A}) \cap \mathrm{Ha}(\mathrm{SP}(\Sigma)).$$

By Remark 6.1.5, the set $\mathrm{SP}_{wa}(\Sigma)$ can be accepted by a branching automaton. Since, by [LW00, Theorem 4.6], the set of sp-languages acceptable by branching automata is closed under intersection, there exists a branching automaton $\overline{\mathcal{B}}$ as required. □

By Theorems 6.1.7 and 6.1.9, branching and asynchronous cellular automata have the same expressive power when restricted to sp-pomsets without autoconcurrency. Since the Hasse-diagrams of these pomsets are $(\Sigma, |\Sigma|^2)$-dags, we can conclude that the expressive power of branching automata coincides with that of MSO when restricted to sp-pomsets without autoconcurrency. We finish the consideration of branching automata showing that this latter result holds for all width-bounded sp-languages thereby answering an open question from [LW00]:

Theorem 6.1.10 *Let $L \subseteq \mathrm{SP}(\Sigma)$ be a width-bounded sp-language. Then L can be accepted by a branching automaton iff it is monadically axiomatizable.*

Proof. By Theorem 6.1.4, a width-bounded language can be accepted by a branching automaton iff it is serial-rational. Since any serial-rational language can be monadically axiomatized by Proposition 6.1.2, we get the implication

$\Rightarrow$. For the other implication, let n be a bound for the width of elements of L and define $\Sigma' := \Sigma \times 2^{[n]}$. The mapping $(a, M) \mapsto a$ naturally induces a projection π from $\mathrm{SP}(\Sigma')$ onto $\mathrm{SP}(\Sigma)$. Now let φ be a monadic sentence over Σ that axiomatizes L. In φ, replace any subformula $\lambda(x) = a$ by $\bigvee_{\emptyset \neq M \subseteq [n]} \lambda(x) = (a, M)$ and denote the resulting sentence by φ'. Then a sp-pomset t over Σ' satisfies φ' iff $\pi(t) \models \varphi$, i.e. iff $\pi(t) \in L$. Since any $s \in L$ has width at most n, there exists even an sp-pomset without autoconcurrency $t \in \mathrm{SP}_{wa}(\Sigma')$ with $\pi(t) = s$. So let L' denote the set of all sp-pomsets without autoconcurrency $t \in \mathrm{SP}_{wa}(\Sigma')$ with $\pi(t) \in L$. Then $\pi(L') = L$ and L' is monadically axiomatized relative to $\mathrm{SP}_{wa}(\Sigma')$ by φ'. Hence, by Lemma 6.1.6 and Theorem 5.2.9, $\mathrm{Ha}(L')$ can be accepted by a Σ'-ACA. Using Theorem 6.1.9, we obtain a branching automaton over Σ' that accepts L'. By [LW00, Theorem 4.6], this implies that $L = \pi(L')$ can be accepted by a branching automaton over Σ since the set of languages acceptable by branching automata is closed with respect to projections. $\square$

Recall that any serial-rational language is weakly rational and any weakly rational language is monadically axiomatizable. Furthermore, there are rational languages that are not monadically axiomatizable (consider $(a \parallel a)^{\oplus}$) but I do not know any weakly rational language that cannot be monadically axiomatized. Hence it seems reasonable to try whether the expressive power of MSO captures that of weakly rational expressions. To capture all rational languages, one presumably needs the ability to count modulo: The language $(a \parallel a)^{\oplus}$ consists of all antichains of *even* size. Such an ability is contained in Courcelle's logic [Cou90]. But there, he also allows to quantify over sets of edges which seems not to be appropriate when one tries to capture rational sp-languages.

6.2 P-asynchronous automata by Arnold

Generalizing the asynchronous automata for traces, Arnold defined P-asynchronous automata that are meant to accept Σ-labeled pomsets without autoconcurrency [Arn91]. In this section, we present in a condensed form some of his definitions and then show that the accepting power of P-asynchronous automata is captured by that of Σ-ACAs.

The tuple $\mathcal{B} = ((S_i)_{i \in I}, (\delta_{a,J})_{a \in \Sigma, J \subseteq I}, \iota, F, I, (D_a)_{a \in \Sigma})$ is a *P-asynchronous automaton over the alphabet* Σ provided

1. I is a finite set of *indices* with $\Sigma \subseteq I$,
2. S_i for $i \in I$ is a finite set of local states of process i,
3. $\delta_{a,J} : \prod_{j \in J} S_j \to \prod_{j \in J} S_j$ is a local transition function for $a \in \Sigma$ and $\emptyset \neq J \subseteq I$,

4. $D_a : S_a \to 2^I \setminus \{\emptyset\}$ is a mapping for $a \in \Sigma$,

5. $\iota \in \prod_{i\in I} S_i$ is the *initial state* and $F \subseteq \prod_{i\in I} S_i$ is the set of *accepting states*.

Above, I said that P-asynchronous automata are meant to accept pomsets. But the way they do this is more involved than for ACAs. First, from an P-asynchronous automaton, one defines a sequential automaton over Σ as follows: The set of states is the direct product of the local state spaces $S = \prod_{i\in I} S_i$. The transition function is defined in two steps: Let $a \in \Sigma$ and $s = (s_i)_{i\in I} \in S$. Then $J := D_a(s)$ is a subset of I. Let $(s'_j)_{j\in J} := \delta_{a,J}((s_j)_{j\in J})$ and, for $i \in I \setminus J$ $s'_i := s_i$. Then $\delta(s, a) := (s'_i)_{i\in I}$. In other words, the transition function $\delta : S \times \Sigma \to S$ changes only some components of its state space. The function D_a decides, which components are changed according to which local transition function $\delta_{a,J}$. Then the tuple (S, δ, ι, F) is a classical sequential automaton over the alphabet Σ. Now let $w = a_1a_2 \ldots a_n \in \Sigma^\star$ be some word over Σ. Since the sequential automaton derived from $\mathcal{B}$ is total and deterministic, there is an initial computation path of the form

$$\iota = (s_i^0)_{i\in I} \xrightarrow{a_1} (s_i^1)_{i\in I} \xrightarrow{a_2} (s_i^2)_{i\in I} \xrightarrow{a_3} \cdots \xrightarrow{a_n} (s_i^n)_{i\in I}.$$

For $1 \le \ell \le n$, let $J_\ell := D_{a_\ell}(s_{a_\ell}^{\ell-1}) \ne \emptyset$, i.e. J_ℓ is the set of components of I that are changed in the ℓth computation step. Then $\sigma(w) := (a_1, J_1)(a_2, J_2) \ldots (a_n, J_n)$ is a word over $\Gamma := \Sigma \times (2^I \setminus \{\emptyset\})$. On the alphabet Γ, we again consider the dependence relation

$$D = \{((a, M), (b, N)) \in \Gamma^2 \mid M \cap N \ne \emptyset \text{ or } a = b\}.$$

Let $[\sigma(w)] = (V, \le, \lambda_\Gamma)$ denote the trace from $\mathbb{M}(\Gamma, D)$ associated to the word $\sigma(w)$. Finally, let $\pi_1 : \Gamma \to \Sigma$ be the projection to the first component of a letter from Γ. Then $[w]_\mathcal{B} := (V, \le, \pi_1 \circ \lambda_\Gamma)$ is a Σ-labeled partially ordered set. Note that it is completely determined by the word $w \in \Sigma^\star$, i.e. the P-asynchronous automaton $\mathcal{B}$ defines a mapping from $\Sigma^\star$ into the set of Σ-labeled partial orders. The image of this mapping is denoted by $P(\mathcal{B}) = \{[w]_\mathcal{B} \mid w \in \Sigma^\star\}$. A set of Σ-labeled pomsets P is *a-regular* if there exists a P-asynchronous automaton $\mathcal{B}$ with $P(\mathcal{B}) = P$.

Let $w \in \Sigma^\star$ and let $(V, \le, \lambda) = [w]_\mathcal{B}$ denote the associated partial order. Since in (Γ, D) pairs with the same letter from Σ are dependent, i.e. $((a, M), (a, N)) \in D$ for any M, N, this partial order has no autoconcurrency. Furthermore, the Σ-dag $(V, -\!\!<, \lambda)$ admits a $|D|$-chain covering since the Hasse-diagram of the trace $[\sigma(w)]$ is a $(\Gamma, |D|)$-dag by Example 5.2.1.

Now let again $\mathcal{B}$ be a P-asynchronous automaton and let (S, δ, ι, F) denote the sequential automaton derived from $\mathcal{B}$. We write $W(\mathcal{B})$ for the set of *words* that are accepted by (S, δ, ι, F) and call this set the *word language accepted by* $\mathcal{B}$. The *pomset-language accepted by* $\mathcal{B}$ is defined by

$$L(\mathcal{B}) := \{[w]_\mathcal{B} \mid w \in W(\mathcal{B})\}.$$

Note that any set of Σ-labeled pomsets that can be accepted by a P-asynchronous automaton is contained in some a-regular set since $L(\mathcal{B}) \subseteq P(\mathcal{B})$. In particular, for any P-asynchronous automaton $\mathcal{B}$, the set $\mathrm{Ha}(L(\mathcal{B}))$ of Hasse-diagrams of accepted pomsets consists of (Σ, k)-dags for some k. On the other hand, for $k > 1$, there is no P-asynchronous automaton $\mathcal{B}$ with $\mathrm{Ha}(L(\mathcal{B})) = \mathbb{D}_k$. In particular, P-asynchronous automata cannot accept $\mathbb{D}_k$ relative to $\mathbb{D}_{k+1}$. Since this is possible by a Σ-ACA (cf. Corollary 5.2.6), the expressive power of Σ-ACAs is not captured by that of P-asynchronous automata. But, on the contrary, any P-asynchronous automaton can be simulated by a Σ-ACA:

Theorem 6.2.1 *Let $\mathcal{B}$ be a P-asynchronous automaton over Σ. Then there exists a Σ-ACA $\mathcal{A}$ with* $\mathrm{Ha}(L(\mathcal{B})) = L(\mathcal{A})$.

Proof. The word language $\sigma(W(\mathcal{B})) = \{\sigma(w) \mid w \in W(\mathcal{B})\}$ is recognizable in $\Gamma^\star$. By [Arn91, Lemma 5.1], $\sigma(W(\mathcal{B}))$ is closed with respect to the trace equivalence, i.e. if $w' \in \sigma(W(\mathcal{B}))$ and $v' \in \Gamma^\star$ with $[w'] = [v']$, then $v' \in \sigma(W(\mathcal{B}))$. Hence the trace language $\{[\sigma(w)] \mid w \in W(\mathcal{B})\} \subseteq \mathbb{M}(\Gamma, D)$ is recognizable. By [EM96], it can be monadically axiomatized, say, by the sentence φ. In φ, replace any subformula of the form $\lambda_\Gamma(x) = (a, M)$ by $\lambda(x) = a \wedge \{i \in I \mid x \in C_i\} = M$ and denote the resulting formula by φ'. Now consider the following sentence ψ:

$$
\begin{aligned}
\exists_{i\in I} C_i : \quad (\quad & C_i \text{ is a chain for } i \in I \\
& \wedge \forall x, y : (x \longrightarrow\!\!\!< y \rightarrow (\lambda(x) = \lambda(y) \vee x, y \in C_i \text{ for some } i \in I)) \\
& \wedge \forall x, y : (x \parallel y \rightarrow \lambda(x) \neq \lambda(y)) \\
& \wedge \varphi' \\
)\quad &
\end{aligned}
$$

Then ψ axiomatizes $L(\mathcal{B})$. Since the order relation $\leq$ can be monadically defined from the covering relation, we get that $\mathrm{Ha}(L(\mathcal{B}))$ can be monadically axiomatized. This is a set of Σ-dags and, even more, it is contained in $\mathbb{D}_k$ with $k = |D|$. Hence, by Theorem 5.2.9, it can be accepted by some Σ-ACA $\mathcal{A}$. Thus, we showed $\mathrm{Ha}(L(\mathcal{B})) = L(\mathcal{A})$ for some Σ-ACA $\mathcal{A}$. □

Thus, the advantage of P-asynchronous automata is that they are deterministic and therefore can easily be complemented. But this complementation always refers to the set $P(\mathcal{B})$, i.e. the complemented P-asynchronous automaton accepts $P(\mathcal{B}) \setminus L(\mathcal{B})$. On the other hand, the expressive power of P-asynchronous automata is strictly weaker than that of asynchronous cellular automata.

Part II

Divisibility monoids

Chapter 7

Preliminaries

In the introduction, I explained that traces can be defined in two different ways: either as combinatorial structures (dependence graphs) or as elements of a free partially commutative monoid. The first part of the present work generalized the first approach considering Σ-dags. Now we are going to deal with a generalization of the second approach, i.e. we consider divisibility monoids. It might not be clear at first glance that they indeed generalize trace monoids since the divisibility monoids are defined in a different spirit than trace monoids, but from Theorem 8.2.10 it follows immediately that any trace monoid is a divisibility monoid. It is the aim of this part to carry over large parts of the theory of recognizable languages in the trace monoid to our setting of divisibility monoid.

This chapter starts with some simple monoid-theoretic preliminaries. Then, we introduce left divisibility monoids and show some of their basic properties that will be useful in our further considerations. These definitions as well as the results in the first two sections are taken from [DK99, DK01]. In the last Section 7.3 of this chapter, a Foata Normal Form for the elements of a divisibility monoid is defined and considered. This Foata Normal Form, besides the fact that it stresses the connection with trace monoids, will be useful later in Chapter 10 where we will characterize when a divisibility monoid satisfies Kleene's Theorem.

7.1 Monoid-theoretic preliminaries

A triple $(M, \cdot, 1)$ is a *monoid* if M is a set, $\cdot : M \times M \to M$ is an associative operation and $1 \in M$ is the *unit element* satisfying $1 \cdot x = x \cdot 1 = x$ for any $x \in M$. Let $(M, \cdot, 1)$ be a monoid and $X \subseteq M$. Then, by $\langle X \rangle$ we denote the submonoid of M generated by X, i.e. the intersection of all submonoids of M that contain X. If $\langle X \rangle = M$, X is a *set of generators of M*. The monoid M is *finitely generated* if it has a finite set of generators. Let X be a set. Then $X^\star$ denotes the set of all words over X. With the usual concatenation of words and the empty word as unit element, this becomes a *free monoid generated by X*.

Let $M = (M, \cdot, 1)$ be a monoid. We call M *cancellative* if $x \cdot y \cdot z = x \cdot y' \cdot z$ implies $y = y'$ for any $x, y, y', z \in M$. This in particular ensures that M does not contain a zero element (i.e. an element z such that $z \cdot x = x \cdot z = z$ for any $x \in M$). Now let $x \cdot y = z$. Then, in a cancellative monoid y is uniquely determined. We denote it by $x^{-1}z$. For $x, y \in M$, *x is a left divisor of y* (denoted $x \leq y$) if there is $z \in M$ such that $x \cdot z = y$. In general, the relation $\leq$ is not antisymmetric.

Lemma 7.1.1 *Let $(M, \cdot, 1)$ be a cancellative monoid and $a \in M$. Then the mapping $a : (M, \leq) \to (a \cdot M, \leq)$ defined by $a(x) := a \cdot x$ is a preorder isomorphism.*

Proof. Since the monoid M is cancellative, the mapping a is bijective. Now let $b, c \in M$. If $ab \leq ac$, we find $d \in M$ such that $abd = ac$. Now $b \leq c$ follows by cancellation. The other implication is trivial. □

Let $T := (M \setminus \{1\}) \setminus (M \setminus \{1\})^2$. The set T consists of those elements of M that do not have a proper divisor, its elements are called *irreducible*. Note that T has to be contained in any set generating M.

The set of *rational sets* in a monoid $(M, \cdot, 1)$ is the least class $\mathfrak{C} \subseteq 2^M$ such that

- all finite subsets of M belong to $\mathfrak{C}$,
- $X \cdot Y = \{x \cdot y \mid x \in X, y \in Y\}$ and $X \cup Y$ belong to $\mathfrak{C}$ whenever $X, Y \in \mathfrak{C}$, and
- $\langle X \rangle$ belongs to $\mathfrak{C}$ whenever $X \in \mathfrak{C}$.

A set $L \subseteq M$ is *recognizable* iff there exists a finite monoid $(S, \cdot, 1)$ and a homomorphism $\eta : M \to S$ such that $L = \eta^{-1}\eta(L)$. Recognizable sets are sometimes called recognizable *languages*. It is easily verified that the set of recognizable languages in a monoid $(M, \cdot, 1)$ is closed under the usual set-theoretic operations like union, intersection and complementation. Furthermore, in any monoid the empty set as well as the whole set are recognizable.

In general, the sets of recognizable and of rational subsets of a monoid are different and even incomparable. For finitely generated monoids, it is known that any recognizable set is rational (and that this property characterizes the finitely generated monoids). The other implication holds in particular in finitely generated free monoids:

Kleene's Theorem ([Kle56]). *Let T be a finite set. Then a set $L \subseteq T^\star$ is rational iff it is recognizable.*

Now let T be a finite set and $L \subseteq M := T^\star$. Since the set of recognizable languages is closed under the usual set-theoretic operations, the set of rational languages in the free monoid $T^\star$ enjoys these closure properties.

For $x \in T^\star$, let $\alpha(x)$ denote the alphabet of x comprising all letters of T that occur in x. Then $L_B := \langle B \rangle \cap L \setminus (\bigcup_{A \subset B} \langle A \rangle)$ with $B \subseteq T$ is the set of elements x of L with $\alpha(x) = B$. If L is rational, the language L_B is rational, too. The language L is *monoalphabetic* if $L = L_B$ for some $B \subseteq T$. The class of *monoalphabetic-rational languages* (m-rational for short) in $T^\star$ is the smallest class $\mathfrak{C} \subseteq 2^{T^\star}$ satisfying

- all finite subsets of $T^\star$ belong to $\mathfrak{C}$,
- $X \cdot Y$ and $X \cup Y$ belong to $\mathfrak{C}$ whenever $X, Y \in \mathfrak{C}$, and
- $\langle X \rangle$ belongs to $\mathfrak{C}$ whenever $X \in \mathfrak{C}$ is monoalphabetic.

The following lemma seems to be folklore but we could not find an explicit reference.

Lemma 7.1.2 *Let T be a finite set. Then a language in $T^\star$ is rational iff it is m-rational.*

Proof. The implication $\Leftarrow$ is trivial. For the other implication, one shows by induction on the size of $B \subseteq T$ that $\langle L \rangle_B$ is m-rational for any rational language L. Then the result follows since $\langle L \rangle$ is the union of these languages.

For $B = \emptyset$, the statement is trivial. Now observe that

$$\langle L \rangle_B = \bigcup_{X,Y \subset B} L_X \cdot \left\langle L_B \cup (L \cdot L)_B \cup \bigcup L_C \langle L \rangle_D L_E \right\rangle \cdot L_Y$$

where the inner union is taken for all $C, D, E \subseteq B$ with $C \cup D \cup E = B$ and $C \cup D \neq B \neq D \cup E$. Hence by the induction hypothesis for $D \subset B$, the languages $\langle L \rangle_D$ are m-rational and therefore $\langle L \rangle_B$ is so, too. □

7.2 Definition and basic properties of divisibility monoids

Definition 7.2.1 A monoid $(M, \cdot, 1)$ is called a *left divisibility monoid* provided the following hold

1. M is cancellative and its irreducible elements form a finite set of generators of M,
2. $x \wedge y$ exists for any $x, y \in M$, and
3. $(\downarrow x, \leq)$ is a distributive lattice for any $x \in M$.

Note that by the third axiom the prefix relation in a left divisibility monoid is a partial order relation. Since, by Lemma 7.1.1, $y \leq z$ implies $x \cdot y \leq x \cdot z$, a left divisibility monoid is a left ordered monoid. Ordered monoids where the order relation is the intersection of the prefix and the suffix relation were investigated e.g. in [Bir73] under the name "divisibility monoid". Despite that we require more than just the fact that $(M, \cdot, \leq)$ be a left ordered monoid this might explain why we call the monoids defined above "left divisibility monoid". Since Birkhoff's divisibility monoids will not appear in our investigations any more, we will simply speak of "divisibility monoids" as an abbreviation for "left divisibility monoid".

Let $(M, \cdot, 1)$ be a divisibility monoid and let $x, y \in M$ with $x \cdot y = 1$. Then $1 \leq x \leq 1$ implies $x = 1$ since by the third axiom $\leq$ is a partial order. Hence we have $y = x \cdot y = 1$, i.e. there are no proper divisors of the unit element.

Example 7.2.2 It is easily seen that any (finitely generated) trace monoid is a left divisibility monoid. Now let $\Sigma = \{a, b, c, d\}$ be an alphabet. Let $\sim^1$ be the least congruence on the free monoid $\Sigma^\star$ that identifies the words ab and cd. In a trace monoid, the equality $ab = cd$ implies $\{a, b\} = \{c, d\}$ for any generators a, b, c, d. Hence the quotient monoid $\Sigma^\star/{\sim^1}$ is not a trace monoid. But, as we will see later, it is a divisibility monoid. Similarly, let $\sim^2$ identify aa and bb. Again, $\Sigma^\star/{\sim^2}$ is no trace but a divisibility monoid. Finally, identifying aa and bc again results in a divisibility monoid. The proof that these three monoids are indeed divisibility monoids is delayed to Chapter 8 where we will give a finite representation for divisibility monoids (cf. Theorem 8.2.10).

Since a divisibility monoid $(M, \cdot, 1)$ is generated by the set T of its irreducible elements, there is a natural epimorphism $\mathrm{nat} : T^\star \to M$. Now let $A \subseteq M$ and $x \in M$ with $A \leq x$. Since $(\downarrow x, \leq)$ is a lattice, the supremum y of A *in this lattice* exists. Now let $z \in M$ be an upper bound of A in $(M, \leq)$ which is not necessarily in the lattice $\downarrow x$. Then y and z have an infimum $y \wedge z$ in $(M, \leq)$. This infimum is an upper bound of A dominated by y. Thus $y = y \wedge z \leq z$. Hence y is even the supremum of A in the partially ordered set $(M, \leq)$. Thus we showed that any set, bounded above, has a supremum in $(M, \leq)$ (Lemma 7.2.3 below will show that bounded sets are finite). This supremum of A can be viewed as the least common multiple of A, whereas the infimum of A is the greatest common (left-)divisor of A. Note that $(M, \leq)$ is not necessarily a lattice since it may contain unbounded pairs of elements. By Lemma 7.1.1, multiplication in a left divisibility monoid $(M, \cdot, 1)$ from the left (but not from the right) distributes over infima and suprema, i.e. $a \cdot (b \wedge c) = ab \wedge ac$ for any $b, c \in M$ and $a \cdot (b \vee c) = ab \vee ac$ provided $\{b, c\}$ (or, equivalently, $\{ab, ac\}$) is bounded above. This is essential in the following proof that shows that any element of a divisibility monoid has only finitely many left divisors:

Lemma 7.2.3 *Let $(M,\cdot,1)$ be a divisibility monoid and $m \in M$. Then $\downarrow m$ is finite.*

Proof. Let $T \subseteq M$ be the set of irreducible elements of M. Then T is a finite set of generators of M. By contradiction, assume $n \in \mathbb{N}$ minimal such that there exist $x_1,\dots,x_n \in T$ with $\downarrow(x_1 \cdot x_2 \dots x_n)$ infinite. Then $n \geq 2$. Let $m := x_1 \cdot x_2 \dots x_n$. Since the set $\downarrow m$ is an infinite distributive lattice, it contains an infinite chain C. By Lemma 7.1.1, $(\{y \in M \mid x_1 \leq y \leq m\},\leq) \cong (\downarrow(x_2 \cdot x_3 \dots x_n),\leq)$. Since n is minimal, the sets $\downarrow x_1$ and $\{y \in M \mid x_1 \leq y \leq m\}$ are finite. Hence there exist $x,y \in C$ such that $x < y$, $x \vee x_1 = y \vee x_1$ and $x \wedge x_1 = y \wedge x_1$. Since $\downarrow m$ is distributive and complements in a distributive lattice are unique [Bir73, Corollary to Theorem II.13], this implies $x = y$ contradicting $x < y$. $\square$

Thus, for an element m of a divisibility monoid, $(\downarrow m,\leq)$ is a finite distributive lattice. Let $|m|$ denote the length of this lattice which equals the size of any maximal chain deduced by 1. It is easily checked that $x \longrightarrow\!\!\!< y$ iff there exists $t \in T$ with $x \cdot \mathrm{nat}(t) = y$ for any $x,y \in M$. Hence the maximal chains in $\downarrow m$ correspond to the words $w \in T^\star$ with $\mathrm{nat}(w) = m$. This implies that any two such words have the same length which equals $|m|$.

By the second requirement on divisibility monoids, the partial order $(M,\leq)$ can be seen as the set of compacts of a Scott-domain. The lemma above ensures that it is even the set of compacts of a dI-domain (cf. [Ber78, Win87]). Thus, we have in particular $(x \vee y) \wedge z = (x \wedge z) \vee (y \wedge z)$ whenever the left hand side is defined.

Let M again be a divisibility monoid. Two elements x and y are *independent* (denoted by $x\,\mathcal{I}\,y$) if $x \wedge y = 1$ and $\{x,y\}$ is bounded above. In this case the supremum $x \vee y$ exists in M. Since M is cancellative, there is a unique element z such that $y \cdot z = x \vee y$. This element z is called *the residuum of x after y* and denoted by $x \uparrow y$. Note that it is defined for independent elements x and y only. Hence $x \uparrow y$ is defined iff $y \uparrow x$ is defined and in this case $x(y \uparrow x) = y(x \uparrow y)$. Fixing $x \in M$, we get a unary partial function c_x from M to M by $\mathrm{dom}(c_x) := \{y \in M \mid x\,\mathcal{I}\,y\}$ and $c_x(y) := y \uparrow x$. The function c_x will be called the *commutation behavior of x*. We may turn it into a total function on M by introducing an additional element. Therefore an equation $c_x(y) = c_z(y')$ means "$c_x(y)$ is defined iff $c_z(y')$ is defined and in this case they are equal".

As an example, consider the trace monoid over the dependence alphabet (Σ,D). Then the infimum of two traces s and t is trivial whenever no letter occurs in s as well as in t. In this case the set $\{s,t\}$ is bounded iff any letter from s is independent from any letter from t. Hence s and t are independent according to our definition above if and only if they are independent in the sense of trace theory. Now assume s and t to be independent traces. Then their supremum equals $st = ts$. Hence the residuum of s after t equals s, i.e. the commutation behavior c_t is contained in the identity function $\mathrm{id}_{\mathbb{M}(\Sigma,D)}$ and is completely given by the set of letters that occur in t. In other words, the notion of a "commutation behavior" in a trace monoid is pretty trivial, but in the context of divisibility monoids it turns out to be of great importance as most of our proofs rely on this concept.

Lemma 7.2.4 *Let* $(M, \cdot, 1)$ *be a divisibility monoid and* $x, y \in M$ *with* $x \mathcal{I} y$*. The commutation behavior* c_x *is injective on its domain. Furthermore,* $|y| = |c_x(y)|$*.*

Proof. Let $y, y' \in \mathrm{dom}(c_x)$ with $c_x(y) = c_x(y')$. Then $x \wedge y = x \wedge y' = 1$. By the definitions of $\uparrow$ and c_x we get $x \vee y = x(y \uparrow x) = x(y' \uparrow x) = x \vee y'$. Hence y and y' are complements of x in the lattice $\downarrow(x \vee y)$. Since complements in a distributive lattice are unique [Bir73], this implies $y = y'$.

To show that c_x is length preserving, let $z = x \vee y = x \cdot c_x(y)$. Then x is the complement of y in the distributive lattice $\downarrow z$. Hence the height of z is the sum of the heights of x and y in this lattice [Bir73]. □

Next we show some formulas satisfied by the commutation behaviors that will be useful in our subsequent considerations.

Lemma 7.2.5 *Let* $(M, \cdot, 1)$ *be a divisibility monoid and* $x, x', y, z \in M$*.*

1. $x \mathcal{I} yz$ *iff* $x \mathcal{I} y$ *and* $c_y(x) \mathcal{I} z$*.*
2. $c_{yz}(x) = c_z(c_y(x))$*; in other words* $x \uparrow (yz) = (x \uparrow y) \uparrow z$*.*
3. $c_x(yz) = c_x(y) \cdot c_{c_y(x)}(z)$*; equivalently* $yz \uparrow x = (y \uparrow x) \cdot (z \uparrow (x \uparrow y))$*.*
4. *If* $c_x = c_{x'}$ *and* $y \mathcal{I} x$ *then* $c_{c_y(x)} = c_{c_y(x')}$*.*

Proof. 1. First we show the implication $\Rightarrow$. Therefore, let $x \mathcal{I} yz$. Then $x \wedge yz = 1$ and the set $\{x, yz\}$ is bounded above. Hence $x \wedge y = 1$ and $\{x, y\}$ is bounded above since $y \leq yz$, i.e. $x \mathcal{I} y$. To show that $c_y(x) \mathcal{I} z$, note that $c_y(x)$ is defined since x and y are independent. Then $yc_y(x) = x \vee y$ by the definition of c_y. Note that $\{x, yz\}$ and therefore $\{x, y, yz\}$ is bounded above. Hence $\{x \vee y, yz\}$ is bounded above. Now the boundedness of $\{c_y(x), z\}$ follows by $x \vee y = yc_y(x)$ and Lemma 7.1.1. Furthermore,

$$\begin{aligned} y \cdot (c_y(x) \wedge z) &= yc_y(x) \wedge yz && \text{by Lemma 7.1.1} \\ &= (x \vee y) \wedge yz \\ &= (x \wedge z) \vee (y \wedge yz) && \text{by distributivity} \\ &= 1 \vee y = y && \text{since } x \mathcal{I} yz \text{ and } y \leq yz \end{aligned}$$

Since M is cancellative, we showed $c_y(x) \wedge z = 1$.

To show the inverse implication, let $x \mathcal{I} y$ and $c_y(x) \mathcal{I} z$. Then $x \wedge yz \leq (x \vee y) \wedge yz = yc_y(x) \wedge yz$. By Lemma 7.1.1 this equals $y(c_y(x) \wedge z)$. Since $c_y(x) \mathcal{I} z$ we get $x \wedge yz \leq y$. Hence $x \wedge yz = x \wedge y \wedge yz = 1 \wedge yz = 1$ since $x \mathcal{I} y$. To show that $\{x, yz\}$ is bounded, note that $\{yc_y(x), yz\}$ is bounded by $c_y(x) \mathcal{I} z$ and Lemma 7.1.1. Hence $\{x \vee y, yz\}$ and therefore $\{x, yz\}$ are bounded which finishes the proof of the first statement.
2. Since M is cancellative, the following equation implies $c_{yz}(x) = c_z(c_y(x))$:

$$\begin{aligned} yzc_{yz}(x) &= x \vee yz && \text{by the definition of the function } c_{yz} \\ &= (x \vee y) \vee yz && \text{since } y \leq yz \\ &= yc_y(x) \vee yz \\ &= y(c_y(x) \vee z) && \text{by Lemma 7.1.1} \\ &= yzc_z(c_y(x)). \end{aligned}$$

3. Similarly, $c_x(yz) = c_x(y)c_{c_y(x)}(z)$ follows from the following equation since we can cancel x from the left:

$$\begin{aligned} xc_x(yz) &= x \vee yz = x \vee y \vee yz && \text{since } y \leq yz \\ &= yc_y(x) \vee yz && \text{since } x \vee y = yc_y(x) \\ &= y(c_y(x) \vee z) && \text{by Lemma 7.1.1} \\ &= yc_y(x)c_{c_y(x)}(z) = (x \vee y)c_{c_y(x)}(z) \\ &= xc_x(y)c_{c_y(x)}(z). \end{aligned}$$

4. Let $z \in \text{dom}(c_{c_y(x)})$. By the second statement and $c_x = c_{x'}$, we obtain $c_x(y)c_{c_y(x)}(z) = c_x(yz) = c_{x'}(yz) = c_{x'}(y)c_{c_y(x')}(z) = c_x(y)c_{c_y(x')}(z)$. Now we can conclude $c_{c_y(x)}(z) = c_{c_y(x')}(z)$ by cancelling $c_x(y)$ from the left. □

Note that the second statement implies $c_z \circ c_y = c_{yz}$ where $\circ$ is the usual concatenation of partial functions. Hence the set $\mathbb{C}_M = \{c_x \mid x \in M\}$ is closed under concatenation. Since c_1 is the identity function on M, $(\mathbb{C}_M, \circ, c_1)$ is a monoid, *the monoid of commutation behaviors of* M. The function $c : M \to \mathbb{C}_M$ with $c(x) := c_x$ is a monoid antihomomorphism. We say that M *has finite commutation behavior* if $\mathbb{C}_M$ is finite, i.e. there are only finitely many different functions c_x ($x \in M$). Actually, it is not clear whether this is a definite restriction since we do not know any divisibility monoid with infinite commuation behavior. But, on the other hand, we did not succeed in proving that any divisibility monoid has finite commutation behavior.

Later, we will need the following

Lemma 7.2.6 *Let M be a divisibility monoid and $x, y \in M$ with $x \mathcal{I} y$ and $c_x \subseteq \mathrm{id}_M$. Then $c_{c_y(x)} \subseteq \mathrm{id}_M$.*

Proof. Let $z \in \mathrm{dom}(c_{c_y(x)})$, i.e. $z \mathcal{I} c_y(x)$. Then (by Lemma 7.2.5(1)) $x \mathcal{I} yz$ implying $c_x(yz) = yz$. But $c_x(yz) = yz \uparrow x = (y \uparrow x) \cdot (z \uparrow (x \uparrow y)) = y \cdot c_{c_y(x)}(z)$ by Lemma 7.2.5(3). Since we can cancel y from the left, we get $z = c_{c_y(x)}(z)$, i.e. $c_{c_y(x)} \subseteq \mathrm{id}_M$. □

Recall that nat : $T^\star \to M$ is a homomorphism. Thus, one can easily define functions $d_x : T^\star \to T^\star$ for $x \in M$ such that $\mathrm{nat} \circ d_x = c_x \circ \mathrm{nat}$. E.g. one could choose any normal form function NF: $M \to T^\star$ with $\mathrm{nat} \circ \mathrm{NF}(x) = x$ for $x \in M$ and then define $d_x(w) := \mathrm{NF} \circ c_x \circ \mathrm{nat}(w)$. But then $\mathrm{im}(d_x)$ consists of normal forms, only. Thus, this partial function was not injective on its domain. In addition, an equation similar to Lemma 7.2.5(3) was very unlikely to hold. Therefore, we follow another way: Recall that for $t \in T$ and $x \in M$ with $x \mathcal{I} t$ we have $|t| = |c_x(t)|$ by Lemma 7.2.4 and therefore $c_x(t) \in T$. Hence $d_x(t) := c_x(t)$ (if $t \mathcal{I} x$) is a partial function mapping T to T. We extend it to a partial function from $T^\star$ to $T^\star$ by $d_x(tu) := d_x(t) d_{c_t(x)}(u)$. Then one can easily check that

$$d_x(t_1 t_2 \ldots t_n) = d_x(t_1) d_{c_{\mathrm{nat}(t_1)}(x)}(t_2) d_{c_{\mathrm{nat}(t_1 t_2)}(x)}(t_3) \ldots d_{c_{\mathrm{nat}(t_1 t_2 \ldots t_{n-1})}(x)}(t_n)$$

and therefore

$$d_x(uv) = d_x(u) d_{c_{\mathrm{nat}(u)}(x)}(v). \tag{7.1}$$

Now let $x, y \in M$, $t \in T$ and $u \in T^\star$. We get immediately $d_{xy}(t) = c_{xy}(t) = c_y(c_x(t)) = d_y(d_x(t))$ since c_x and c_y are length preserving. Now we can conclude

$$\begin{array}{rcll} d_y(d_x(tu)) & = & d_y(d_x(t) d_{c_t(x)}(u)) & \text{by (7.1)} \\ & = & d_y(d_x(t)) \cdot d_{c_{d_x(t)}(y)}(d_{c_t(x)}(u)) & \text{by (7.1)} \\ & = & d_{xy}(t) \cdot d_{c_t(x) c_{d_x(t)}(y)}(u) & \\ & = & d_{xy}(t) \cdot d_{c_t(xy)}(u) & \\ & = & d_{xy}(tu). & \end{array}$$

Now let $v \in T^\star$ be a word over T. Then we define $d_v := d_{\mathrm{nat}(v)}$. Thus, $\mathrm{dom}(d_v) = \{u \in T^\star \mid \mathrm{nat}(u) \mathcal{I} \mathrm{nat}(v)\}$. We write $u \mathcal{I} v$ for $u \in \mathrm{dom}(d_v)$. Note that, similarly to c_x, we have the following

Lemma 7.2.7 *Let $(M, \cdot, 1)$ be a divisibility monoid and $u, v, w \in T^\star$.*

$$\begin{array}{rcl} d_{vw}(u) & = & d_w(d_v(u)), \\ d_u(vw) & = & d_u(v)\ d_{d_v(u)}(w), \text{ and} \\ \mathrm{nat}(d_v(u)) & = & c_{\mathrm{nat}(v)}(\mathrm{nat}(u)). \end{array}$$

Furthermore, d_u is injective on its domain and length preserving.

Proof. Immediate by Lemmas 7.2.5 and 7.2.4. □

Let $\mathbb{D}_M = \{d_u \mid u \in T^\star\}$ be the set of all commutation behaviors of words over T. Then $(\mathbb{D}_M, \circ, d_\varepsilon)$ is a monoid and $d : T^\star \to \mathbb{D}_M : u \mapsto d_u$ is a monoid antihomomorphism by Lemma 7.2.7. Since $\mathrm{nat}(t) = t$ for $t \in T$, in this case the third equation can be written as $d_v(t) = c_{\mathrm{nat}(v)}(t)$. Using the first and the third equation from Lemma 7.2.7, the mapping $d_u \mapsto c_{\mathrm{nat}(u)}$ turns out to be a monoid homomorphism from $(\mathbb{D}_M, \circ, d_\varepsilon)$ onto $(\mathbb{C}_M, \circ, c_1)$. The following lemma shows that it is injective, i.e. that it is even an isomorphism.

Lemma 7.2.8 *Let $u, v \in T^\star$. Then $d_u = d_v$ iff $c_{\mathrm{nat}(u)} = c_{\mathrm{nat}(v)}$.*

Proof. The implication $\Rightarrow$ is immediate by the third equation from Lemma 7.2.7. Now let $c_{\mathrm{nat}(u)} = c_{\mathrm{nat}(v)}$. If $t \in T$ and $d_u(t)$ is defined, then $c_{\mathrm{nat}(u)}(t) = c_{\mathrm{nat}(v)}(t)$ and therefore $d_v(t)$ is defined. Furthermore, $d_u(t) = c_{\mathrm{nat}(u)}(t) = d_v(t)$ proving the claim for arguments from T. Now let $w \in T^\star$. Then $d_u(tw) = d_u(t)\ d_{c_t(\mathrm{nat}(u))}(w)$. By the above argument, $d_u(t) = d_v(t)$. Furthermore, by Lemma 7.2.5(4), $c_{c_t(\mathrm{nat}(u))} = c_{c_t(\mathrm{nat}(v))}$. Now $d_{c_t(\mathrm{nat}(u))}(w) = d_{c_t(\mathrm{nat}(v))}(w)$ follows from the induction hypothesis. Hence $d_u(tw) = d_v(t)\ d_{c_t(\mathrm{nat}(v))}(w) = d_v(tw)$. □

Let $u, u', v, w \in T^\star$. If $d_w(u) = v$ and $\mathrm{nat}(u) = \mathrm{nat}(v)$, by the third equation in Lemma 7.2.7, it holds $\mathrm{nat}(d_w(u')) = \mathrm{nat}(v)$. Conversely, assume $\mathrm{nat}(d_w(u')) = \mathrm{nat}(v)$. Then the following lemma shows that there exists $u \in T^\star$ with $d_w(u) = v$ (by the injectivity of $c_{\mathrm{nat}(w)}$ in addition $\mathrm{nat}(u) = \mathrm{nat}(u')$).

Lemma 7.2.9 *Let $x, y \in M$ and $t_i \in T$ for $1 \le i \le n$ such that $c_x(y) = \mathrm{nat}(t_1 t_2 \dots t_n)$. Then there exist $s_i \in T$ for $1 \le i \le n$ such that $d_x(s_1 s_2 \dots s_n) = t_1 t_2 \dots t_n$. These elements s_i of T are unique.*

Proof. Since $x \wedge y = 1$, the intervals $[1, y]$ and $[x, x \vee y]$ are transposed and therefore isomorphic by [Bir73, Theorem I.13] and an isomorphism is given by $a \mapsto a \vee x$ for $a \in {\downarrow} y$. Inductively, define s_i to be the unique element in M with $\mathrm{nat}(s_1 s_2 \dots s_i) \vee x = x \cdot \mathrm{nat}(t_1 t_2 \dots t_i)$. Then one can easily show that s_i does not have a proper divisor. In addition, $t_{i+1} = s_{i+1} \uparrow (x \uparrow \mathrm{nat}(s_1 s_2 \dots s_i))$ and therefore $d_x(s_1 s_2 \dots s_n) = t_1 t_2 \dots t_n$. The uniqueness is immediate by the proof. □

7.3 A Foata Normal Form

Throughout this section, let $(M, \cdot, 1)$ be a fixed divisibility monoid and let T denote the set of its irreducible elements. For simplicity, let $\mathbb{J}(x)$ denote the join-irreducible elements of the distributive lattice ${\downarrow}x$ for any $x \in M$. We define the set of *cliques* $\mathcal{C}\ell$ to consist of all nonempty subsets of T that are bounded above. Since any subset of M that is bounded above has a supremum, we have $\mathcal{C}\ell = \{A \subseteq T \mid \emptyset \neq A \text{ and } \sup(A) \text{ exists}\}$. Let $A \in \mathcal{C}\ell$. Then any two distinct elements $s, t \in A$ are bounded above. Furthermore, since s is an atom in the partially ordered set $(M, \leq)$, the infimum of s and t belongs to $\{1, s\}$. But s and t are incomparable. Hence we showed that any two distinct elements of A are independent. But this property does not characterize the cliques. The reason is that even if any two elements of $A \subseteq T$ are bounded above, the set A need not be bounded.

Next we define the set FNF consisting of words over $\mathcal{C}\ell$ as

$$\{A_1 A_2 \ldots A_n \in \mathcal{C}\ell^\star \mid \forall t \in A_{i+1} \forall B \in \mathcal{C}\ell : \sup B \neq (\sup A_i) \cdot t \text{ for } 1 \leq i < n\}.$$

Since the condition that constitutes membership in FNF is local, FNF is a rational language in $\mathcal{C}\ell^\star$. In addition, FNF is closed under cancellation from the left and from the right, i.e. $U, V, W \in \mathcal{C}\ell^\star$ with $U\,V\,W \in \text{FNF}$ implies $V \in \text{FNF}$. Let $\alpha' : \mathcal{C}\ell \to M$ denote the mapping that associates with any clique $A \in \mathcal{C}\ell$ its supremum $\sup A$ in M. This mapping can be extended uniquely to a monoid homomorphism α from $\mathcal{C}\ell^\star$ to M. Then $\alpha(A_1 A_2 \ldots A_n) = (\sup A_1) \cdot (\sup A_2) \cdots (\sup A_n)$. This mapping is surjective since $\alpha(\{t_1\}\{t_2\} \ldots \{t_n\}) = t_1 \cdot t_2 \cdots t_n$ for any $t_i \in T$ and T generates M. On the other hand, it is easily seen not to be injective. The set FNF is particularly useful since it provides normal forms for the elements of M, i.e. since the restriction of α to FNF is a bijection (cf. Lemma 7.3.3). But before we can prove this lemma, we need some more order theory:

Let $(L, \leq)$ be a distributive lattice and $x \in L$. Then the set ${\uparrow}x$ together with the partial order $\leq \cap ({\uparrow}x \times {\uparrow}x)$ is a distributive lattice with join-irreducible elements $\mathbb{J}({\uparrow}x)$. Note that in general $\mathbb{J}({\uparrow}x) \neq \mathbb{J}(L) \cap {\uparrow}x$. The following lemma relates the join-irreducible elements of L and those of $({\uparrow}x, \leq)$.

Lemma 7.3.1 *Let $(L, \leq)$ be a finite distributive lattice and $x \in \mathbb{J}(L)$. The mapping $f : \mathbb{J}(L) \setminus {\downarrow}x \to \mathbb{J}({\uparrow}x)$ with $f(y) = x \vee y$ is an order isomorphism.*

Proof. Let $y \in \mathbb{J}(L) \setminus {\downarrow}x$. First we show that $f(y) \in \mathbb{J}({\uparrow}x)$: Let $a, b \in L$ with $x \leq \{a, b\}$ and $a \vee b = x \vee y$. Then we have $y = y \wedge (x \vee y) = y \wedge (a \vee b) = (y \wedge a) \vee (y \wedge b)$. Since y is join-irreducible in $(L, \leq)$, this implies (without loss of generality) $y = y \wedge a$, i.e. $y \leq a$. Thus, we have $\{x, y\} \leq a \leq x \vee y$. Hence, $a = x \vee y$ proving that $x \vee y$ is join-irreducible in the distributive lattice $(x{\uparrow}, \leq)$.

To show that f is order preserving and reflecting, let $y_1, y_2 \in \mathbb{J}(L)$. Clearly, $y_1 \leq y_2$ implies $x \vee y_1 \leq x \vee y_2$. Suppose conversely $x \vee y_1 \leq x \vee y_2$. Then

$y_1 \leq x \vee y_2$. Since $y_1 \not\leq x$, we obtain $y_1 \leq y_2$ from the fact that y_1 is prime in $(L, \leq)$. □

Lemma 7.3.2 *Let $W = A_1 A_2 \dots A_n \in \mathrm{FNF}$ and $x := \alpha(W) \in M$. Then we have $A_1 = \{t \in T \mid t \leq x\}$, and $\alpha(A_1 A_2 \dots A_i) = \sup\{y \in \mathbb{J}(x) \mid h(y, \mathbb{J}(x)) < i\}$ for $1 \leq i \leq n$.*

Proof. Since $t \leq \alpha(A_1) \leq \alpha(W) = x$ for $t \in A_1$, the inclusion "$\subseteq$" is immediate. For simplicity, let $A = \{t \in T \mid t \leq x\}$. Then $A \in \mathcal{C}\ell$ and $({\downarrow}(\sup A), \leq)$ is isomorphic to the power set of A, ordered by inclusion. If $A_1 \neq A$, there is $t \in T$ with $\sup(A_1) \cdot t \leq \alpha(A) \leq x = \alpha(A_1)\alpha(A_2 A_3 \dots A_n)$. By cancellation, we get $t \leq \alpha(A_2 A_3 \dots A_n)$. Inductively, this implies $t \in A_2$ since $A_2 A_3 \dots A_n \in \mathrm{FNF}$. Hence we found $t \in A_2$ and a clique $A \in \mathcal{C}\ell$ such that $\sup(A_1) \cdot t \leq \sup(A)$. Since $({\downarrow}\sup(A), \leq)$ is isomorphic to the powerset of A, there is $B \subseteq A$ with $\sup(A_1) \cdot t = \sup(B)$, contradicting $A_1 A_2 \in \mathrm{FNF}$.

Note that $\{y \in \mathbb{J}(x) \mid h(y, \mathbb{J}(x)) < 1\} = \{t \in T \mid t \leq x\}$. Hence the second statement holds for $i = 1$. Now assume

$$a := \alpha(A_1 A_2 \dots A_{i-1}) = \sup\{y \in \mathbb{J}(x) \mid h(y, \mathbb{J}(x)) < i - 1\}.$$

Then $a \cdot z = x$ with $z = \alpha(A_i A_{i+1} \dots A_n)$. Since $A_i A_{i+1} \dots A_n \in \mathrm{FNF}$, by the first statement, $A_i = \{t \in T \mid t \leq z\}$ follows. Thus $\alpha(A_1 A_2 \dots A_i) = \alpha(A_1 A_2 \dots A_{i-1}) \cdot \alpha(A_i) = a \cdot \sup\{t \in T \mid t \leq z\}$. Then $\alpha(A_1 A_2 \dots A_i) = a \vee \sup\{at \mid t \in T, t \leq z\}$ by Lemma 7.1.1. Note that $\{at \mid t \in T, t \leq z\}$ is the set of elements of the distributive lattice $([a, az], \leq)$ of height 1 in this lattice. Hence it is the set of elements of height 0 in the set $(\mathbb{J}([a, az]), \leq)$ of join-irreducibles. Now Lemma 7.3.1 implies

$$\begin{aligned} \{at \mid t \in T, t \leq z\} &= \{y \in \mathbb{J}([a, az]) \mid h(y, \mathbb{J}([a, az])) = 0\} \\ &= \{a \vee y' \mid y' \in \mathbb{J}(az) \setminus {\downarrow}a \text{ and } h(y', \mathbb{J}(az) \setminus {\downarrow}a) = 0\}. \end{aligned}$$

Since ${\downarrow}a \cap \mathbb{J}(az) = \{y' \in \mathbb{J}(az) \mid h(y', \mathbb{J}(az)) < i - 1\}$, we get

$$\{at \mid t \in T, t \leq z\} = \{a \vee y' \mid y' \in \mathbb{J}(az) \text{ and } h(y', \mathbb{J}(az)) = i - 1\}$$

and therefore

$$\begin{aligned} \alpha(A_1 A_2 \dots A_i) &= a \vee \sup\{a \vee y' \mid y' \in \mathbb{J}(az) \text{ and } h(y', \mathbb{J}(az)) = i - 1\} \\ &= a \vee \sup(y \in \mathbb{J}(az) \mid h(y, \mathbb{J}(az) = i - 1\} \\ &= \sup\{y' \in \mathbb{J}(az) \mid h(y', \mathbb{J}(az)) < i\}. \end{aligned}$$

□

Now the bijectivity of $\alpha \restriction \mathrm{FNF}$ follows:

Lemma 7.3.3 *The mapping* $\alpha \restriction \mathrm{FNF} : \mathrm{FNF} \to M$ *is bijective.*

Proof. The injectivity follows inductively from the first statement of Lemma 7.3.2. To show surjectivity, let $x, y \in M \setminus \{1\}$, $A = \{t \in T \mid t \leq x\}$, $a = \sup(A)$, $a \cdot y = x$ and $B = \{t \in T \mid t \leq y\}$. It is sufficient to show that $AB \in \mathrm{FNF}$, i.e. that $A, B \in \mathcal{C}\ell$ and that $\sup(C) \neq \sup(A) \cdot t$ for any $t \in B$ and $C \in \mathcal{C}\ell$. But A and B are nonempty since $x \neq 1 \neq y$, and A and B have suprema since they are bounded by x and y, respectively. Thus, $A, B \in \mathcal{C}\ell$. Now assume $t \in B$ and $C \in \mathcal{C}\ell$ with $\sup(C) = \sup(A) \cdot t$. Then, for any $s \in C$: $s \leq \sup(A) \cdot t \leq x$ implies $s \in A$, i.e. $C \subseteq A$. But this contradicts $\sup(C) > \sup(A)$. □

Thus, for any $x \in M$, the set $\alpha^{-1}(x) \cap \mathrm{FNF}$ is a singleton. We denote the unique preimage of x in FNF by $\mathrm{fnf}(x)$ and call it the *Foata Normal Form* of x. An immediate consequence of the second statement of Lemma 7.3.2 is

Corollary 7.3.4 *Let* $x \in M$*. Then* $|\mathrm{fnf}(x)|$ *exceeds the length of the partially ordered set* $(\mathbb{J}(x), \leq)$ *by* 1*.*

Next we show that the Foata Normal Form of $\mathrm{nat}(w)$ can be computed from the word $w \in T^\star$ by an automaton. In general, this automaton has infinitely many states. But for "width-bounded divisibility monoids" (cf. Section 10.2) it will be shown to be finite. This finiteness will be the basis for our proof that "width-bounded divisibility monoids" are rational and therefore satisfy Kleene's Theorem.

An *automaton over a monoid* M is a quintuple $\mathcal{A} = (Q, M, E, I, F)$ where

1. Q is a set of *states*,
2. $E \subseteq Q \times M \times Q$ is a set of *transitions*, and
3. $I, F \subseteq Q$ are the sets of *initial and final states*, respectively.

The automaton $\mathcal{A}$ is *finite* if E is. We will write $p \xrightarrow{a} q$ for $(p, a, q) \in E$. A *computation* in $\mathcal{A}$ is a finite sequence of transitions:

$$p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} p_2 \cdots \xrightarrow{a_n} p_n.$$

It is *successful* if $p_0 \in I$ and $p_n \in F$. The *label* of the computation is the element $a_1 \cdot a_2 \cdots a_n$ of the monoid M. For a computation with first state p_0, last state p_n and label a, we will usually write $p_0 \xrightarrow{a} p_n$ without mentioning the intermediate states. The *behavior of* $\mathcal{A}$ is the subset $|\mathcal{A}|$ of M consisting of labels of successful computations in $\mathcal{A}$.

If the monoid M is a direct product $M_1 \times M_2$ of two monoids, it is convenient to think of M_1 as the input and of M_2 as the output of the automaton. Then the automaton computes from an input in M_1 an output from M_2. In our context, the input will be in the free monoid $T^\star$ and the output in the free monoid $\mathcal{C}\ell^\star$ (actually, in the recognizable language $\mathrm{FNF} \subseteq \mathcal{C}\ell^\star$). Therefore, we will construct an automaton $\mathcal{A}_M$ over the monoid $T^\star \times \mathcal{C}\ell^\star$ as follows. The state set is the direct product of M and $\mathcal{C}\ell_\varepsilon := \mathcal{C}\ell \cup \{\varepsilon\}$, the only initial state is $(1, \varepsilon)$ and the set of final states is $\{1\} \times \mathcal{C}\ell_\varepsilon$. Now let $(x, A), (z, C) \in M \times \mathcal{C}\ell_\varepsilon$ and $(t, B) \in T \times \mathcal{C}\ell_\varepsilon$. Then $(x, A) \overset{(t,B)}{\to} (z, C)$ iff

1. $t \leq x$, $B = \varepsilon$, $t \cdot z = x$ and $C = A$, or

2. $t \,\mathcal{I}\, x$, $B = C \neq \varepsilon$, $AB \in \mathrm{FNF}$, and $t \cdot z = x \cdot (\sup B)$.

Since the transition relation is defined for labels from $T \times \mathcal{C}\ell_\varepsilon$, only, the length of a computation equals that of the input word from $T^\star$. Furthermore, the transition relation E in this automaton is deterministic since, for any state (x, A) and any $(t, B) \in T \times \mathcal{C}\ell_\varepsilon$ at most one of the conditions $t \leq x$ or $t \,\mathcal{I}\, x$ can be satisfied. Thus, the starting state and the label of a computation determine its last state completely. This final state is described in the following lemma.

Lemma 7.3.5 *Let $w \in T^\star$, $B_1 B_2 \ldots B_m \in \mathcal{C}\ell^\star$, $z \in M$ and $C \in \mathcal{C}\ell_\varepsilon$. Then in the automaton $\mathcal{A}_M$, $(1, \varepsilon) \overset{(w, B_1 B_2 \ldots B_m)}{\longrightarrow} (z, C)$ iff*

(i) $|\mathrm{fnf} \circ \mathrm{nat}(w)| = m$,

(ii) $\mathrm{fnf}(\mathrm{nat}(w) \cdot z) = B_1 B_2 \ldots B_m$, *and*

(iii) $C = B_m$.

Proof. We prove the lemma by induction on the length of the input word w. Since the only computation with input word $\varepsilon \in T^\star$, starting in $(1, \varepsilon)$, is $(1, \varepsilon) \overset{(\varepsilon,\varepsilon)}{\to} (1, \varepsilon)$, the statement is obvious for $|w| = 0$. Now assume that the statement holds whenever $|w| < n$.

Now, let $v \in T^\star$ and $t \in T$ with $vt = w$ and $|w| = n$ and assume that $(1, \varepsilon) \overset{(w, B_1 B_2 \ldots B_m)}{\longrightarrow} (z, C)$ holds. First, consider the case that the last transition in this computation is of the first kind, i.e. that $(1, \varepsilon) \overset{(v, B_1 B_2 \ldots B_m)}{\longrightarrow} (t \cdot z, C) \overset{(t,\varepsilon)}{\to} (z, C)$. Then $\mathrm{fnf}(\mathrm{nat}(w) z) = \mathrm{fnf}(\mathrm{nat}(v) \cdot tz)$ which equals $B_1 B_2 \ldots B_m$ by the induction hypothesis. Thus, (ii) holds. To show (i), note that $m = |\mathrm{fnf}(\mathrm{nat}(v))| \leq |\mathrm{fnf}(\mathrm{nat}(vt) \cdot z)| = m$, i.e. $|\mathrm{fnf}(\mathrm{nat}(vt))| = m$. Finally, (iii) holds since by the induction hypothesis $C = B_m$. Now assume that the last transition is of the second kind, i.e. there is a state (x, A) such that $(1, \varepsilon) \overset{(v, B_1 B_2 \ldots B_{m-1})}{\longrightarrow} (x, A) \overset{(t, B_m)}{\to} (z, C)$ with $t \,\mathcal{I}\, x$, $B_m = C \neq \varepsilon$, $AB_m \in \mathrm{FNF}$, and $tz = x(\sup B_m)$. By the induction hypothesis, $|\mathrm{fnf}(\mathrm{nat}(v))| = m - 1$, $\mathrm{fnf}(\mathrm{nat}(v) \cdot x) = B_1 B_2 \ldots B_{m-1}$ and $A = B_{m-1}$. Then $\mathrm{nat}(vt) \cdot z = \mathrm{nat}(v) \cdot x \cdot \sup(B_m) = \alpha(B_1 B_2 \ldots B_m)$. Since the words

$B_1B_2\ldots B_{m-1}$ and $B_{m-1}B_m = AB_m$ belong to FNF, we have $B_1B_2\ldots B_m \in$ FNF, i.e. we showed (ii). It remains to show that m is the length of $\mathrm{fnf}(vt)$. Clearly, $m-1 = |\mathrm{fnf}(\mathrm{nat}(v))| \le |\mathrm{fnf}(\mathrm{nat}(vt)\cdot z)| = m$. Now assume $|\mathrm{fnf}(\mathrm{nat}(vt))| = m-1$. Then, by Corollary 7.3.4, the partially ordered set $(\mathbb{J}(\mathrm{nat}(vt)), \le)$ has length $m-2$, i.e. $\mathbb{J}(\mathrm{nat}(vt)) \subseteq \{y \in \mathbb{J}(\mathrm{nat}(w)z) \mid h(y, \mathbb{J}(\mathrm{nat}(w)z)) \le m-2\}$. Hence from Lemma 7.3.2 we get

$$\begin{aligned} \mathrm{nat}(vt) &= \sup \mathbb{J}(\mathrm{nat}(vt)) \\ &\le \sup\{y \in \mathbb{J}(\mathrm{nat}(w)z) \mid h(y, \mathbb{J}(\mathrm{nat}(w)z)) \le m-2\} \\ &= \alpha(B_1B_2\ldots B_{m-1}) = \mathrm{nat}(v)x \end{aligned}$$

by the induction hypothesis. Hence $t \le x$ by cancellation, contradicting $t \,\mathcal{I}\, x$. Thus, we showed $|\mathrm{fnf}(\mathrm{nat}(vt))| = m$, i.e. (iii).

Conversely, let $|\mathrm{fnf}(\mathrm{nat}(vt))| = m$, $\mathrm{fnf}(\mathrm{nat}(vt)\cdot z) = B_1B_2\ldots B_m$ and $C = B_m$. We want to show $(1,\varepsilon) \stackrel{(vt,B_1B_2\ldots B_m)}{\longrightarrow} (z, C)$. First, assume $|\mathrm{fnf}(\mathrm{nat}(v))| = m$. Then $(t\cdot z, C) \stackrel{(t,\varepsilon)}{\to} (z, C)$. Since $|\mathrm{fnf}(\mathrm{nat}(v))| = m$, $\mathrm{fnf}(\mathrm{nat}(v)\cdot t\cdot z) = B_1B_2\ldots B_m$ and $C = B_m$, we can apply the induction hypothesis and get $(1,\varepsilon) \stackrel{(v,B_1B_2\ldots B_m)}{\longrightarrow} (t\cdot z, C)$. Thus, $(1,\varepsilon) \stackrel{(vt,B_1B_2\ldots B_m)}{\longrightarrow} (z, C)$.

Now consider the case $|\mathrm{fnf}(\mathrm{nat}(v))| < m$. Since $\mathrm{nat}(v) \longrightarrow\!\!\!< \mathrm{nat}(vt)$ in the partially ordered set $(M, \le)$, there is $y \in \mathbb{J}(\mathrm{nat}(vt))$ with $\mathbb{J}(\mathrm{nat}(v)) \dot\cup \{y\} = \mathbb{J}(\mathrm{nat}(vt))$. Hence the length of $(\mathbb{J}(\mathrm{nat}(v)), \le)$ and that of $(\mathbb{J}(\mathrm{nat}(vt)), \le)$ differ at most by one, i.e. $|\mathrm{fnf}(\mathrm{nat}(v))| = m-1$ by Corollary 7.3.4. Therefore, $\mathrm{nat}(v) \le \alpha(B_1B_2\ldots B_{m-1})$ by Corollary 7.3.4. Since the length of the partially ordered set $(\mathbb{J}(\alpha(B_1B_2\ldots B_{m-1}), \le))$ is $m-2$ and that of $(\mathbb{J}(\mathrm{nat}(vt)), \le)$ equals $m-1$, we get $\mathrm{nat}(vt) \not\le \alpha(B_1B_2\ldots B_{m-1})$.

From $\mathrm{nat}(v) \le \alpha(B_1B_2\ldots B_{m-1})$, we deduce the existence of $x \in M$ such that $\mathrm{nat}(v)\cdot x = \alpha(B_1B_2\ldots B_{m-1})$. This implies in particular $\mathrm{fnf}(\mathrm{nat}(v)\cdot x) = B_1B_2\ldots B_{m-1}$. In addition, $\mathrm{nat}(v)\cdot x\cdot \alpha(B_m) = \alpha(B_1B_2\ldots B_{m-1}B_m) = \mathrm{nat}(vt)\cdot z$. Hence $x\alpha(B_m) = t\cdot z$. To show $(x, B_{m-1}) \stackrel{(t,B_m)}{\to} (z, B_m)$, it remains to prove $t\,\mathcal{I}\,x$. Since $\{t, x\} \le t\cdot z$ and $t \in T$, it is sufficient to ensure $t \not\le x$. So assume $t \le x$. Then $\mathbb{J}(\mathrm{nat}(vt)) \subseteq \mathbb{J}(\mathrm{nat}(v)\cdot x)$. Hence the length of $\mathbb{J}(\mathrm{nat}(vt))$ is bounded by that of $(\mathbb{J}(\mathrm{nat}(v)\cdot x), \le)$ which equals $m-2$ since $\mathrm{fnf}(\mathrm{nat}(v)\cdot x) = B_1B_2\ldots B_{m-1}$. But this contradicts $|\mathrm{fnf}(\mathrm{nat}(vt))| = m$. Hence indeed $(x, B_{m-1}) \stackrel{(t,B_m)}{\to} (z, B_m)$.

Recall that $|\mathrm{fnf}(\mathrm{nat}(v))| = m-1$ and $\mathrm{fnf}(\mathrm{nat}(v)\cdot x) = B_1B_2\ldots B_{m-1}$. Hence, we can use the induction hypothesis and obtain $(1,\varepsilon) \stackrel{(v,B_1B_2\ldots B_{m-1})}{\longrightarrow} (x, B_{m-1})$. But this implies $(1,\varepsilon) \stackrel{(vt,B_1B_2\ldots B_m)}{\longrightarrow} (z, B_m)$ since $(x, B_{m-1}) \stackrel{(t,B_m)}{\to} (z, B_m)$. □

Now we can show that the automaton $\mathcal{A}_M$ computes for any input word $w \in T^\star$ the Foata Normal Form $\mathrm{fnf} \circ \mathrm{nat}(w)$ of the associated element of the divisibility monoid M:

Theorem 7.3.6 *Let M be a divisibility monoid. Then the behavior $|\mathcal{A}_M|$ of the automaton $\mathcal{A}_M$ is the relation $\{(w, \mathrm{fnf}(\mathrm{nat}(w)) \mid w \in T^\star\}$ in $T^\star \times \mathcal{C}\ell^\star$, i.e. the automaton computes the function* $\mathrm{fnf} \circ \mathrm{nat} : T^\star \to \mathcal{C}\ell^\star$.

Proof. By Lemma 7.3.5, an element (w, W) of $T^\star \times \mathcal{C}\ell^\star$ is the label of a successful computation, i.e. of a computation that starts in $(1, \varepsilon)$ and ends in $\{1\} \times \mathcal{C}\ell_\varepsilon$, if and only if $\mathrm{fnf}(\mathrm{nat}(w) \cdot 1) = W$. □

Chapter 8

A finite representation

By definition, trace monoids M are finitely presented, i.e. there exists a finite set of equations of the form $ab = ba$ with $a, b \in \Sigma$ such that M is isomorphic to $\Sigma^\star / \langle ab = ba \mid (a,b) \in I \rangle$. Later, an algebraic characterization of trace monoids was found [Dub86]. Differently, divisibility monoids are defined by their algebraic properties. In this chapter, we show that they can be finitely presented (cf. Theorem 8.2.10). Not only will we show that this is possible in general, but we will give a concrete representation for any divisibility monoid (cf. Lemma 8.2.1). Finally, we give a decidable class of finite presentations that give rise to all divisibility monoids. But first, we prove two order-theoretic results that we will need in this context.

8.1 Order-theoretic preliminaries

Lemma 8.1.1 *Let $(M, \leq)$ be a partially ordered set with least element such that*

1. *$\downarrow x$ is finite for any $x \in M$, and*
2. *for any x, y_1, y_2, z with $x \multimap y_1, y_2$ and $\{y_1, y_2\} \leq z$, the least upper bound $y_1 \vee y_2$ in $(M, \leq)$ exists.*

Then any two elements of M that are bounded above have a least upper bound in $(M, \leq)$.

Proof. Let $y_1, y_2, z \in M$ with $y_1, y_2 \leq z$. We have to show that y_1 and y_2 admit a supremum. It can be assumed that y_1 and y_2 are incomparable for otherwise we were done. Since $(M, \leq)$ has a least element there is $x \in M$ with $x \leq y_1, y_2$. Since $\downarrow z$ is finite, the size of the chains in $[x, z]$ is bounded. We will prove the existence of $y_1 \vee y_2$ by induction on the size of these chains.

If any chain in $[x, z]$ has size at most 2, we have $x \multimap y_1, y_2$. Hence $y_1 \vee y_2$ exists. Now assume that any chain in $[x, z]$ has at most $n + 1$ elements. Let

C_i for $i = 1, 2$ be maximal chains in $[x, z]$ containing y_i. Since C_i is maximal, it contains x. Let x_i be the least element of $C_i \setminus \{x\}$. Since y_1 and y_2 are incomparable, this implies $x \prec x_i \leq y_i$ for $i = 1, 2$. Hence the supremum $x_1 \vee x_2 =: a$ exists. Note that x_i is a lower bound of y_i and a. Since $x \prec x_i \leq z$, the chains in the interval $[x_i, z]$ contain at most n elements. Hence by the induction hypothesis $y_i' := y_i \vee a$ exist for $i = 1, 2$. Since $x \prec x_1 \leq a$, the size of the chains in the interval $[a, z]$ is bounded by n. Hence we can apply the induction hypothesis to y_1' and y_2' and obtain the existence of their supremum $y_1' \vee y_2'$.

We show that $b := y_1' \vee y_2'$ is the supremum of y_1 and y_2: Since $y_i \leq y_i'$, we obtain $y_i \leq b$ for $i = 1, 2$. Now let c be an upper bound of y_1 and y_2. Then it is an upper bound of x_1 and x_2 and therefore of a, too. Hence $y_i' \leq c$ for $i = 1, 2$ and therefore $b \leq c$. □

By the Vilhelm-Šik-Jakubik Theorem (cf. [Ste91, Theorem 4.14]), any finite semimodular lattice that is not modular contains a non-modular interval of length 3. Next, we prove a similar result that distinguishes modular from distributive lattices[1]

Lemma 8.1.2 *Let $(L, \leq)$ be a finite modular but non-distributive lattice. Then it contains a non-distributive interval of length 2.*

Proof. Let $[a, b]$ be a minimal non-distributive interval. Since $[a, b]$ is modular and non-distributive, there are mutually distinct elements $y_1, y_2, y_3 \in [a, b]$ with $y_i \wedge y_j = a$ and $y_i \vee y_j = b$ for $1 \leq i < j \leq 3$ by [Bir73, Theorem II.13]. Hence the intervals $[a, y_i]$ and $[y_j, b]$ are transposed for $i \neq j$. Since the lattice $(L, \leq)$ is modular, all these intervals are mutually isomorphic [Bir73, Theorem I.13].

Let $a \prec a' \leq y_1$. Let $y_2' := y_2 \vee a'$ and $y_3' := y_3 \vee a'$. Then y_2' and y_3' belong to the interval $[a', b]$ which is distributive since it is a proper subinterval of $[a, b]$ (cf. Figure 8.1). Hence we have

$$
\begin{aligned}
b &= b \wedge b \\
&= (y_1 \vee y_3') \wedge (y_2' \vee y_3') && \text{since } y_i \leq y_i' \leq b \\
&= (y_1 \wedge y_2') \vee y_3' && \text{since } y_1, y_2', y_3' \in [a', b] \\
& && \quad \text{and this interval is distributive} \\
&= (y_1 \wedge (y_2 \vee a')) \vee y_3' \\
&= ((y_1 \wedge y_2) \vee a') \vee y_3' && \text{since } a' \leq y_1 \text{ and } [a, b] \text{ is modular} \\
&= y_3' && \text{since } (y_1 \wedge y_2) = a \leq a' \leq y_3'.
\end{aligned}
$$

Since $a = a' \wedge y_3$ and $b = y_3' = a' \vee y_3$, the intervals $[a, a']$ and $[y_3, b]$ are transposed and therefore isomorphic. Hence $y_3 \prec b$. Since the intervals $[a, y_i]$

[1] We give the proof although, by [FGL90, p. 270], it "is a well known result in the folklore of lattice theory".

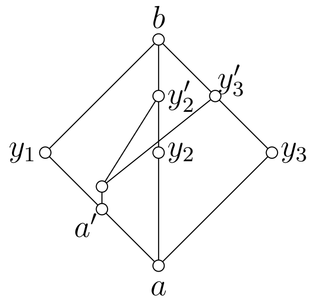


Figure 8.1: The elements from the proof of Lemma 8.1.2

and $[y_j, b]$ are mutually isomorphic, we therefore get $a \longrightarrow\!\!< y_1 \longrightarrow\!\!< b$, i.e. the interval $[a, b]$ has length 2. □

8.2 The finite presentation

Since a divisibility monoid M is generated by the set T of its irreducible elements, there is a congruence $\sim$ on the free monoid $T^\star$ such that the quotient $T^\star/\sim$ is isomorphic to M. The following result that was shown in [DK99, DK01] states that this congruence is quite natural and that the monoid M is finitely presentable.

Lemma 8.2.1 *Let M be a divisibility monoid and T the set of its irreducible elements. Let $\sim$ denote the least congruence on the free monoid $T^\star$ containing $\{(ab, cd) \mid a, b, c, d \in T$ and $a \cdot b = c \cdot d\}$. Then $\sim$ is the kernel of the natural epimorphism* $\mathrm{nat} : T^\star \to M$. *In particular, $M \cong T^\star/\sim$.*

Proof. Throughout this proof, we denote the product of $a, b \in T$ in the monoid M by $a \cdot b$, while the word is denoted by ab. Thus, $a \cdot b = \mathrm{nat}(ab) \in M$ and $ab \in T^\star$.

Note that $u \sim v$ implies that u and v have the same length.

Clearly, the kernel of nat contains $\sim$ since $a \cdot b = c \cdot d$ implies $\mathrm{nat}(ab) = \mathrm{nat}(cd)$ for $a, b, c, d \in T$. For the converse, let $u, v \in T^\star$ with $\mathrm{nat}(u) = \mathrm{nat}(v)$. We show $u \sim v$ by induction on the length of u: If $|u| = 0$, clearly, $\mathrm{nat}(u) = \mathrm{nat}(v) = 1$, so $u = v = \varepsilon$. If $|u| = 1$ then $u = v \in T$.

Now let $u = u_1u_2\dots u_n$ and $v = v_1v_2\dots v_n$ with $u_i, v_i \in T$ and $n \geq 2$. If $u_1 = v_1$, we get $\mathrm{nat}(u_2u_3\dots u_n) = \mathrm{nat}(v_2v_3\dots v_n)$ by cancellation. By the induction hypothesis, this implies $u_2u_3\dots u_n \sim v_2v_3\dots v_n$ and therefore $u_1u_2\dots u_n \sim v_1v_2\dots v_n$. Thus assume $u_1 \neq v_1$. Then u_1, v_1 are elements of the finite distributive lattice $\downarrow \mathrm{nat}(u)$. Hence there exist $b, d \in M$ such that $u_1 \cdot b = v_1 \cdot d = u_1 \vee v_1$. One even has $b, d \in T$ since the distributive lattice $\downarrow \mathrm{nat}(u)$ is semimodular [Bir73]. Note that $u_1b \sim v_1d$ by the definition of $\sim$. Since $u_1 \vee v_1 \leq \mathrm{nat}(u)$, we find $y \in T^\star$ such that $(u_1 \vee v_1) \cdot \mathrm{nat}(y) = \mathrm{nat}(u)$. Hence $\mathrm{nat}(u_1u_2\dots u_n) = \mathrm{nat}(u_1by)$ and therefore $\mathrm{nat}(u_2\dots u_n) = \mathrm{nat}(by)$ by cancellation. The induction hypothesis ensures $u_2\dots u_n \sim by$. Now we have $\mathrm{nat}(v_1dy) = (u_1 \vee v_1) \cdot \mathrm{nat}(y) = \mathrm{nat}(u) = \mathrm{nat}(v)$ implying $\mathrm{nat}(dy) = \mathrm{nat}(v_2v_3\dots v_n)$. Now $dy \sim v_2v_3\dots v_n$ follows from the induction hypothesis since the length of $v_2v_3\dots v_n$ equals $n-1$. Thus, we have $u_1u_2\dots u_n \sim u_1by \sim v_1dy \sim v_1v_2\dots v_n$. □

In particular, Lemma 8.2.1 states that any divisibility monoid is (up to isomorphism) given by the equations $a \cdot b = c \cdot d$ for irreducible elements a, b, c, d that hold in M. Next, we want to characterize, which sets of equations of this form give rise to divisibility monoids.

For the rest of this section, let T be a finite set and E a set of word equations over T of the form $ab = cd$ for $a, b, c, d \in T$. Let $\sim$ denote the least congruence on the free monoid $T^\star$ that contains E. In addition, let $M := T^\star/\!\sim$ be the quotient of the free monoid with respect to $\sim$. Furthermore, we require that the following hold in the monoid M for any $a, b, c, a', b', c' \in T$:

(i) $(\downarrow(a \cdot b \cdot c), \leq)$ is a distributive lattice,

(ii) $a \cdot b \cdot c = a \cdot b' \cdot c'$ or $b \cdot c \cdot a = b' \cdot c' \cdot a$ implies $b \cdot c = b' \cdot c'$, and

(iii) $a \cdot b = a' \cdot b'$, $a \cdot c = a' \cdot c'$ and $a \neq a'$ imply $b = c$.

We will show that M is a divisibility monoid.

Remark 8.2.2 Let $(\overline{M}, \cdot, 1)$ be a divisibility monoid. Let $\overline{T}$ be the set of irreducible generators of $\overline{M}$ and let $\overline{E}$ consist of all equations of the form $a \cdot b = c \cdot d$ with $a, b, c, d \in \overline{T}$ that hold in $\overline{M}$. Then by Lemma 8.2.1, $\overline{M} \cong \overline{T}^\star / \langle \overline{E} \rangle$. Furthermore, the distributivity in (i) is trivial since it holds for any $x \in \overline{M}$. Similarly, (ii) is a special instance of the cancellation property in $\overline{M}$. To show (iii) assume $a \cdot b = a' \cdot b'$, $a \cdot c = a' \cdot c'$ and $a \neq a'$. Then $ab \wedge ac$ exists. Note that a and a' are distinct lower bounds of $\{ab, ac\}$. Hence the infimum of ab and ac lies above a and a' and below ab. But since a and a' are direct predecessors of ab, this implies $ab \wedge ac = ab$. Hence $ab = ac$ implying $b = c$ by cancellation. Thus, any divisibility monoid can be obtained this way.

Example 8.2.3 As an example, consider the monoid $M = T^\star / \langle (ab, cd), (de, ed) \rangle$ where a, b, c, d, e are mutually different elements of the finite set T. Properties (i) and (ii) are easily checked by considering all possible situations. The third property is trivially satisfied. Let $\eta : (\mathbb{N} \times \mathbb{N}, +, (0, 0)) \to (M, \cdot, 1)$ be the monoid homomorphism defined by $\eta(1, 0) = d$ and $\eta(0, 1) = e$. Then the preimage of the rational language $L = \langle (d \cdot e) \rangle \subseteq M$ in $\mathbb{N} \times \mathbb{N}$ is $\{(n, n) \mid n \in \mathbb{N}\}$. Since this set is not recognizable, L is not recognizable in M. Hence Kleene's Theorem does not hold in M. Furthermore, in any trace monoid $ab = cd$ for irreducible elements a, b, c and d implies $\{a, b\} = \{c, d\}$. Since this is not satisfied by $(M, \cdot, 1)$, this monoid is not free partially commutative.

Next, consider $M_1 = T^\star / \langle (ab, cc) \rangle$ and $M_2 = T^\star / \langle (aa, bb) \rangle$ where a, b, c are pairwise different elements of the finite set T. Again, these two monoids satisfy the conditions (i),(ii) and (iii). They are no trace monoids by the same argument as above. We only mention for the sake of completeness that these two are neither concurrency monoids as considered in [Dro95, Dro96, DK96, DK98] (where we extend the multiplication freely whenever it was the null element), since in concurrency monoids $ab = cc$ implies $a = b = c$.

Lemma 8.2.4 *Let $a, b, c \in T$. Then $ab \sim ac$ or $ba \sim ca$ implies $b = c$.*

Proof. First let $ab \sim ac$. Then $abb \sim acb$ and $abc \sim acc$. By (ii), this implies $bb \sim cb$ and $bc \sim cc$. Now (iii) ensures $b = c$. Now let $ba \sim ca$. Then $bba \sim bca$ implying by (ii) $bb \sim bc$. By what we saw before, this implies $b = c$. □

If v and w are words over T satisfying $v \sim w$, then w is obtained from v by a finite sequence of transformations according to the set of equations E. We call two words strongly equivalent if this sequence has length 1. More formally, v and w are *strongly equivalent* ($v \approx w$) if there are words $x, y \in T^\star$ and an equation $ab = cd$ in E such that $v = xaby$ and $w = xcdy$. Thus, w can be obtained from v by replacing two consecutive letters by equivalent ones (according to the set of equations E). To recall the position where this change has been made, we sometimes write it as an index to $\approx$, i.e. with the symbols from above, $v \approx_{|x|+1} w$. In the same spirit, let $\approx_{>i} = \bigcup_{j>i} \approx_j$. Then $v \approx_{>i} w$ denotes that one change has been made to obtain w from v and that this change occurred at a position behind i. Then $\sim$ is the least equivalence on the set $T^\star$ that contains the relation $\approx$.

For a word $w \neq \varepsilon$, let w^h denote the first letter (the "head") and w^t the remaining word (the "tail"), i.e. $w^h \in T$ and $w = w^h w^t$. For a sequence $(w_0, w_1, \ldots w_k)$ of nonempty words, we will consider the number of changes in the first position, i.e.

$$\text{changes}(w_0, w_1, \ldots w_k) := |\{i \mid 0 \leq i < k \text{ and } w_i^h \neq w_{i+1}^h\}|.$$

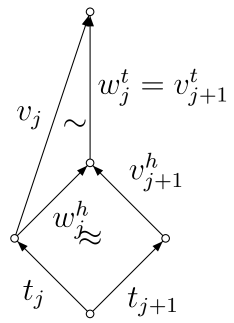


Figure 8.2: Condition (2) and (3) from Lemma 8.2.5

Let $v, w \in T^\star$ with $v \sim w$. The distance $d(v, w)$ denotes the minimal number of changes at the first position in a sequence that transforms v to w. More formally, it is the minimum over all integers changes$(w_0, w_1, \ldots, w_k)$ where w_i are words over T with $v = w_0$, $w_i \approx w_{i+1}$ and $w_k = w$.

Before showing that the distance is bounded, we give an alternative definition:

Lemma 8.2.5 *Let $v, w \in T^\star$ with $v \sim w$. Then $d(v, w)$ is the least integer m such that there exist $t_j \in T$ and $v_j, w_j \in T^\star$ for $0 \leq j \leq m$ with*

(1) $v = t_0 v_0$,

(2) $v_j \sim w_j$ *for* $0 \leq j \leq m$,

(3) $t_j w_j^h \approx t_{j+1} v_{j+1}^h$, $w_j^t = v_{j+1}^t$, $t_j \neq t_{j+1}$ *for* $0 \leq j < m$, *and*

(4) $t_m w_m = w$.

The second and third statement are visualized by Figure 8.2.

Proof. By the definition of $d(v, w)$, it is sufficient to prove that the existence of $x_i \in T^\star$ with $v = x_0$, $x_i \approx x_{i+1}$, $x_n = w$ and $m =$ changes$(x_0, \ldots, x_n)$ is equivalent to the existence of $t_i \in T$ and $v_i, w_i \in T^\star$ satisfying (1)-(4).

First, let $x_i \in T^\star$ $(0 \leq i \leq n)$ with $v = x_0$, $x_i \approx x_{i+1}$, $x_n = w$, and changes$(x_0, \ldots, x_n) = m$. There exist $0 = i_0 < i_1 < \cdots < i_m < i_{m+1} = n + 1$ with

(a) $x_{i_j - 1}^h \neq x_{i_j}^h$ for $0 < j \leq m$ and

(b) $x_{i_j}^h = x_k^h$ for $0 \leq j \leq m$ and $i_j \leq k < i_{j+1}$.

Let $t_j := x^h_{i_j}$, $v_j := x^t_{i_j}$ and $w_j := x^t_{i_{j+1}-1}$ for $0 \leq j \leq m$. Then $v = x_0 = x^h_0 x^t_0 = t_0 v_0$ ensures property (1). Since $i_m < i_{m+1} = n+1$, we get $i_m \leq n < i_{m+1}$. Now (b) with $j = m$ and $k = n$ implies $t_m = x^h_{i_m} = x^h_n$. This implies $t_m w_m = x^h_n x^t_n = x_n = w$ which in turn proves property (4). To show (2), let $0 \leq j \leq m$. Then $x^h_k = t_j$ for $i_j \leq k < i_{j+1}$ and $t_j v_j = x_{i_j} \approx x_{i_j+1} \approx x_{i_j+2} \cdots \approx x_{i_{j+1}-1} = x^h_{i_{j+1}-1} x^t_{i_{j+1}-1} = t_j w_j$. Hence, by Lemma 8.2.4, the strong equivalence $x_k \approx x_{k+1}$ is induced by some change at a higher position, i.e. $x_k \approx_{>1} x_{k+1}$. But this implies $x^t_k \approx x^t_{k+1}$ and therefore $v_j \sim w_j$. To show (3), note that $t_j \neq t_{j+1}$ holds by (a). By (b) and the definition of w_j, we have $t_j w_j = x^h_{i_{j+1}-1} x^t_{i_{j+1}-1} = x_{i_{j+1}-1}$. Similarly, $t_{j+1} v_{j+1} = x^h_{i_{j+1}} x^t_{i_{j+1}} = x_{i_{j+1}}$ by the definition of t_{j+1} and v_{j+1}. Since $x_{i_{j+1}-1} \approx x_{i_{j+1}}$ we therefore get $t_j w_j \approx t_{j+1} v_{j+1}$. From (a), we obtain $t_j = x^h_{i_{j+1}-1} \neq x^h_{i_{j+1}} = t_{j+1}$. Hence $t_j w_j \approx_1 t_{j+1} v_{j+1}$ and therefore (3).

Conversely, let t_j, v_j and w_j satisfy (1)-(4) and consider the sequence

$$(t_0 v_0, t_0 w_0, t_1 v_0, t_1 w_1, \ldots, t_n v_n, t_n w_n).$$

Since $v_j \sim w_j$, we can put additional words between $t_j v_j$ and $t_j w_j$ all of which start with t_j such that the resulting sequence has the desired form. $\square$

Now we can show that the distance is bounded by 1.

Lemma 8.2.6 *Let $v, w \in T^\star$ with $v \sim w$. Then*

(1) $d(v,w) \leq 1$ and

(2) if $v^h = w^h$, then $v^t \sim w^t$.

Proof. First of all suppose $d(v,w) \leq 1$ and $v^h = w^h$. Since v and w have the same first letter, $d(v,w) \neq 1$, i.e. $d(v,w) = 0$. Hence there is a sequence of words transforming v to w that leaves the first letter unchanged. This implies $v^t \sim w^t$. Thus, (1) implies (2) and we have to show the first statement, only. This is done by induction on the length of v which equals that of w. If $|v| \leq 1$, we get $d(v,w) = 0$. Next we consider the case $|v| = 2$. Then $(\downarrow[v^h v^t v^t], \leq)$ is a distributive lattice by (i). Using (ii), one gets that $(\downarrow[v], \leq)$ is a sublattice. Since it is of length 2, it contains at most 2 elements different from 1 and from $[v]$. Hence $d(v,w) \leq 1$.

By induction, we assume that (1) and (2) hold for any $v', w' \in T^\star$ with $v' \sim w'$ and $|v'| < |v|$.

Assume $n := d(v,w) > 1$. By Lemma 8.2.5, there are $t_i \in T$ and $v_i, w_i \in T^\star$ for $0 \leq i \leq n$ such that

$v = t_0 v_0$,
$v_i \sim w_i$ for $0 \leq i \leq n$,
$t_i w^h_i \approx t_{i+1} v^h_{i+1}$, $w^t_i = v^t_{i+1}$, $t_i \neq t_{i+1}$ for $0 \leq i < n$, and
$t_n w_n = w$.

We prove that there exist words $x, y \in T^\star$ satisfying $w_0 \sim x$, $y \sim v_2$ and $t_0x^h \approx t_2y^h$ or $t_0x^h = t_2y^h$. Once we will have found them, we can conclude $t_0w_0 \sim_{>1} t_0x(\approx_1 \cup =)t_2y \sim_{>1} t_2v_2$ which decreases the number of changes at the first position, contradicting $n = d(v, w)$.

First we consider the case $t_0 = t_2$ and show that $x = w_0$ and $y = v_2$ are the desired elements: By $t_0w_0^h \approx t_1v_1^h$ and $t_0v_2^h = t_2v_2^h \approx t_1w_1^h$, (iii) implies $v_1^h = w_1^h$ since $t_0 \neq t_1$. Thus we have $t_0x^h = t_0w_0^h \approx t_1v_1^h = t_1w_1^h \approx t_2v_2^h = t_0y^h$. Applying Lemma 8.2.4 to t_0x^h and t_0y^h, we get $x^h = y^h$, i.e. $t_0x^h = t_2y^h$ as required.

Now let $t_0 \neq t_2$. If $v_1^h = w_1^h$, we had $t_0w_0^h \sim t_1v_1^h = t_1w_1^h \sim t_2v_2^h$. As we saw above, $\downarrow[t_0w_0^h]$ contains at most 2 elements different from 1 and from $[t_0w_0^h]$. Since t_0, t_1 and t_2 are three elements of this set, we derived a contradiction. Hence we showed $v_1^h \neq w_1^h$. By the induction hypothesis for v_1 and w_1, we get $d(v_1, w_1) \leq 1$. Since they start with different letters, their distance is 1, i.e. there are in particular $a, b \in T$ and $z \in T^\star$ such that $v_1 \sim_{>1} v_1^haz \approx_1 w_1^hbz \sim_{>1} w_1$. This ensures

$$t_0w_0^ha \approx_1 t_1v_1^ha \approx_2 t_1w_1^hb \approx_1 t_2v_2^hb.$$

Hence t_0 and t_2 are different elements of the distributive lattice $(\downarrow[t_0w_0^ha], \leq)$. Therefore, there are $c, d, e \in T$ with $t_0c \sim t_2d$ and $t_0ce \sim t_0w_0^ha$. The latter in particular implies $ce \sim w_0^ha$ by (ii). Thus we have

$$t_2v_2^hb \sim t_0w_0^ha \sim t_0ce \approx_1 t_2de$$

which implies $v_2^hb \sim de$ by (ii) again. From the induction hypothesis (2), applied to the equivalence $v_1^haz \sim v_1 = v_1^hv_1^t$, the equivalence $az \sim v_1^t = w_0^t$ follows. Similarly, $bz \sim w_1^t = v_2^t$ follows from $w_1^hbz \sim w_1 = w_1^hw_1^t$. Hence we have

- $w_0 = w_0^hw_0^t \sim w_0^haz \approx_1 cez =: x$,
- $v_2 = v_2^hv_2^t \sim v_2^hbz \approx_1 dez =: y$, and
- $t_0x^h = t_0c \approx t_2d = t_2y^h$.

This proves that $x = cez$ and $y = dez$ satisfy the desired properties. □

Corollary 8.2.7 $(M, \cdot, 1)$ *is left cancellative and hence the left divisor relation* $\leq$ *is a partial order on* M.

Proof. Cancellation is immediate by Lemma 8.2.6 (2). To prove the antisymmetry of $\leq$, one uses the simple observation that $1 = [\varepsilon]$ has no left divisor. □

Lemma 8.2.8 *Let $x \in M$ and $s, t \in T$ with $s \neq t$ such that $\{xs, xt\}$ is bounded above in $(M, \leq)$. Then there exists $a \in T$ such that xsa is the least upper bound of xs and xt in $(M, \leq)$.*

Proof. Since by Lemma 7.1.1 the function $y \mapsto xy$ is an order isomorphism, it is sufficient to consider the case $x = 1$. Let $y \in M$ with $s, t \leq y$. By Lemma 8.2.6, there are $a_y, b_y \in T$ with $sa_y \approx ta_y$ and $[sa_y] \leq y$. Now let $z \in M$ be some upper bound of s and t. Then, as for y, we obtain $a_z, b_z \in T$ with $sa_z \approx ta_z$ and $[sa_z] \leq z$. Now (iii) implies $a_y = a_z$. Hence xsa_y is the supremum of xs and xt in the partially ordered set $(M, \leq)$. □

Lemma 8.2.9 *For $x, y \in M$, $(\downarrow x, \leq)$ is a distributive lattice and $x \wedge y$ exists.*

Proof. Clearly, $\downarrow x \subseteq \{[v] \mid v \in T^\star, |v| \leq |x|\}$ is finite. By Lemma 8.2.8, we can apply Lemma 8.1.1. Hence $(\downarrow x, \leq)$ is a lattice since any two elements of $\downarrow x$ are bounded above. It is even semimodular by Lemma 8.2.8. To show that it is modular, consider some interval $[y, yabc]$ of $\downarrow x$ with $y \in M$ and $a, b, c \in T$. By left-cancellation (Corollary 8.2.7), it is sufficient to deal with the case $y = 1$. But then $[1, abc] = \downarrow(abc)$ which is distributive by (i) and therefore in particular modular. Hence by the Vilhelm-Šik-Jakubik Theorem [Ste91, Theorem 4.14], $\downarrow x$ is modular. To show distributivity, we consider some interval of length 2 and argue similarly using Lemma 8.1.2.

The set $\downarrow x \cap \downarrow y$ is finite and bounded. Hence, by Lemma 8.1.1, it has a least upper bound which is the maximal lower bound of x and y, i.e. $x \wedge y$ exists. □

Now we can prove the main theorem of this chapter.

Theorem 8.2.10 *Let T be a finite set and E a set of equations of the form $ab = cd$ with $a, b, c, d \in T$. Let $\sim$ be the least congruence on $T^\star$ containing E. Then $M := T^\star/\!\sim$ is a divisibility monoid if and only if (i)-(iii) hold for any $a, b, c, b', c' \in T$:*

(i) $(\downarrow(a \cdot b \cdot c), \leq)$ is a distributive lattice,

(ii) $a \cdot b \cdot c = a \cdot b' \cdot c'$ or $b \cdot c \cdot a = b' \cdot c' \cdot a$ implies $b \cdot c = b' \cdot c'$, and

(iii) $a \cdot b = a' \cdot b'$, $a \cdot c = a' \cdot c'$ and $a \neq a'$ imply $b = c$.

Furthermore, each divisibility monoid arises this way.

Proof. By Remark 8.2.2, it remains to show that T and E satisfying (i)-(iii) define a divisibility monoid. By Corollary 8.2.7 and Lemma 8.2.9, it remains to prove that $(M, \cdot, 1)$ is right cancellative. For this, it suffices to show that $xa = ya$ with $x, y \in M$ and $a \in T$ implies $x = y$. By contradiction, assume that $x \neq y$.

Since the lattice $\downarrow xa$ is distributive, $z := x \wedge y \longrightarrow\!\!\!< x, y$, i.e. there are $b, c \in T$ with $x = zb$ and $y = zc$. Hence $zba = zca$. Now $ba = ca$ follows from Corollary 8.2.7. Lemma 8.2.4 ensures $b = c$ and therefore $x = y$. □

Let (Σ, D) be a dependence alphabet. Let E denote the set of all equations $ab = ba$ for $(a, b) \in \Sigma^2 \setminus D$. Then $\mathbb{M}(\Sigma, D) = \Sigma^\star / \langle E \rangle$. One can easily check that the three properties (i), (ii) and (iii) of the theorem above hold. Hence a trace monoid is indeed a divisibility monoid.

Chapter 9

An Ochmański-type theorem

Kleene's Theorem on recognizable languages of finite words has been generalized in several directions, e.g. to formal power series [Sch61] and to infinite words [Büc60]. More recently, rational monoids were investigated [Sak87], in which the recognizable languages coincide with the rational ones. Building on results from [CP85, CM88, Mét86], a complete characterization of the recognizable languages in a trace monoid by c-rational sets was obtained in [Och85]. A further generalization of Kleene's and Ochmański's results to concurrency monoids was given in [Dro95]. In this chapter, we derive such a result for divisibility monoids. The proofs by Ochmański [Och85] and by Droste [Dro95] rely on the *internal* structure of the elements of the monoids. Here, we do not use the internal representation of the monoid elements, but algebraic properties of the monoid itself. The results presented in this chapter were obtained together with Manfred Droste. They appeared in [DK99] and the presentation follows [DK01].

9.1 Commutation grids and the rank

In trace theory, the generalized Levi Lemma (cf. [DM97]) plays an important role. It was extended to concurrency monoids in [Dro95]. Here, we develop a further generalization to divisibility monoids using commutation grids. This enables us to obtain the concept of "rank" of a language for these monoids, similar to the one given by Hashigushi [Has91] for trace monoids. Let M be a divisibility monoid and $x, y \in M$. Recall that $c_x(y) = y \uparrow x$. Sometimes (for instance in the following definition), it is more convenient to use this notation for the functions d_u, too. Therefore, we define $v \uparrow u := d_u(v)$ whenever the latter is defined for $u, v \in T^\star$.

Definition 9.1.1 For $0 \leq i \leq j \leq n$ let $x_j^i, y_i^j \in M$ ($\in T^\star$, respectively). The tuple $(x_j^i, y_i^j)_{0 \leq i \leq j \leq n}$ is a *commutation grid* in M (in $T^\star$, respectively) provided the following holds for any $0 \leq i < j \leq n$:

$$x_j^i \,\mathcal{I}\, y_i^{j-1}, \quad x_j^i \uparrow y_i^{j-1} = x_j^{i+1}, \quad \text{and} \quad y_i^{j-1} \uparrow x_j^i = y_i^j.$$

A commutation grid can be depicted as in Figure 9.1. There, edges depict elements from M ($T^\star$, resp.) and an angle denotes that the two edges correspond to independent elements. Note that in any of the small squares in Figure 9.1, the lower left corner is marked by an angle. This indicates that $x_j^i y_i^j = y_i^{j-1} x_j^{i+1}$ because of $x_j^i y_i^j = x_j^i(y_i^{j-1} \uparrow x_j^i) = x_j^i \vee y_i^{j-1} = y_i^{j-1}(x_j^i \uparrow y_i^{j-1}) = y_i^{j-1} x_j^{i+1}$. By Lemma 7.2.5 (1)-(3), for any rectangle in the grid $(x_j^i, y_i^j)_{0\le i\le j\le n}$ the bottom and the left side are independent and their residuum is the top (the right) side, respectively. By induction, it is easy to show that

$$(x_1^0 x_2^0 \dots x_n^0) \cdot (y_0^n y_1^n \dots y_n^n) = (x_0^0 y_0^0)(x_1^1 y_1^1)(x_2^2 y_2^2) \dots (x_n^n y_n^n).$$

The right hand side of this equation is the diagonal border of the grid in Figure 9.1.

Let $(M, \cdot, 1)$ be a divisibility monoid and $(x_j^i, y_i^j)_{0\le i\le j\le n}$ a commutation grid in M or in $T^\star$. For a sequence $0 = i_0 < i_1 < \dots < i_{m+1} = n$, we construct a subgrid $(a_l^k, b_k^l)_{0\le k\le l\le m}$ as follows: Define $a_l^k := x_{i_l}^{i_k} x_{i_l+1}^{i_k} x_{i_l+2}^{i_k} \dots x_{i_{l+1}-1}^{i_k}$ and $b_k^l := y_{i_k}^{i_{l+1}-1} y_{i_k+1}^{i_{l+1}-1} \dots y_{i_{k+1}-1}^{i_{l+1}-1}$ (for $m = 4$ and $\vec{\imath} = (0, 1, 5, 7, 9)$, this grid is marked by thick lines in Figure 9.1). Then $(a_l^k, b_k^l)_{0\le k\le l\le m}$ is a commutation grid in M or $T^\star$. We call it the *subgrid generated by the sequence* $(i_k)_{0\le k\le m}$.

Let $(x_j^i, y_i^j)_{0\le i\le j\le n}$ be a commutation grid in $T^\star$. Then it is immediate that $(\mathrm{nat}(x_j^i), \mathrm{nat}(y_i^j))_{0\le i\le j\le n}$ is a commutation grid in M. The following lemma deals with the converse implication. More precisely, let a commutation grid in M be given and suppose that $u_j^0, v_j^n \in T^\star$ are representatives of monoid elements at the left and the upper border of the commutation grid. Then the lemma states that this tuple of words can be extended to a commutation grid in $T^\star$ that is compatible with the commutation grid in M we started with.

Lemma 9.1.2 *Let $(x_j^i, y_i^j)_{0\le i\le j\le n}$ be a commutation grid in M and let u_j^0, v_j^n be words from $T^\star$ that satisfy $\mathrm{nat}(u_j^0) = x_j^0$ and $\mathrm{nat}(v_n^j) = y_n^j$ for $0 \le j \le n$. Then there exists a commutation grid $(u_j^i, v_i^j)_{0\le i\le j\le n}$ with $\mathrm{nat}(u_j^i) = x_j^i$ and $\mathrm{nat}(v_i^j) = y_i^j$ for $0 \le i \le j \le n$.*

Proof. For $1 \le i \le j \le n$ let $u_j^i := d_{y_i^{j-1}}(u_j^{i-1})$. Using Lemma 7.2.7, one can check that $\mathrm{nat}(u_j^i) = x_j^i$ and that therefore $u_j^{i-1} \in \mathrm{dom}(d_{y_i^{j-1}})$. To construct the elements v_i^j, we use Lemma 7.2.9: Let v_i^j be the unique word over T such that $v_i^j \uparrow u_{j+1}^i = v_i^{j+1}$. Then $\mathrm{nat}(v_i^j) = y_i^j$ is immediate since $y_i^j \in M$ is the unique complement of x_{j+1}^i in the distributive lattice $[x_{j+1}^i, x_{j+1}^i \vee y_i^j]$. It is clear that $(u_j^i, v_i^j)_{0\le i\le j\le n}$ is a commutation grid because of $\mathrm{nat}(u_j^i) = x_j^i$ and $\mathrm{nat}(v_i^j) = y_i^j$ for $0 \le i \le j \le n$. □

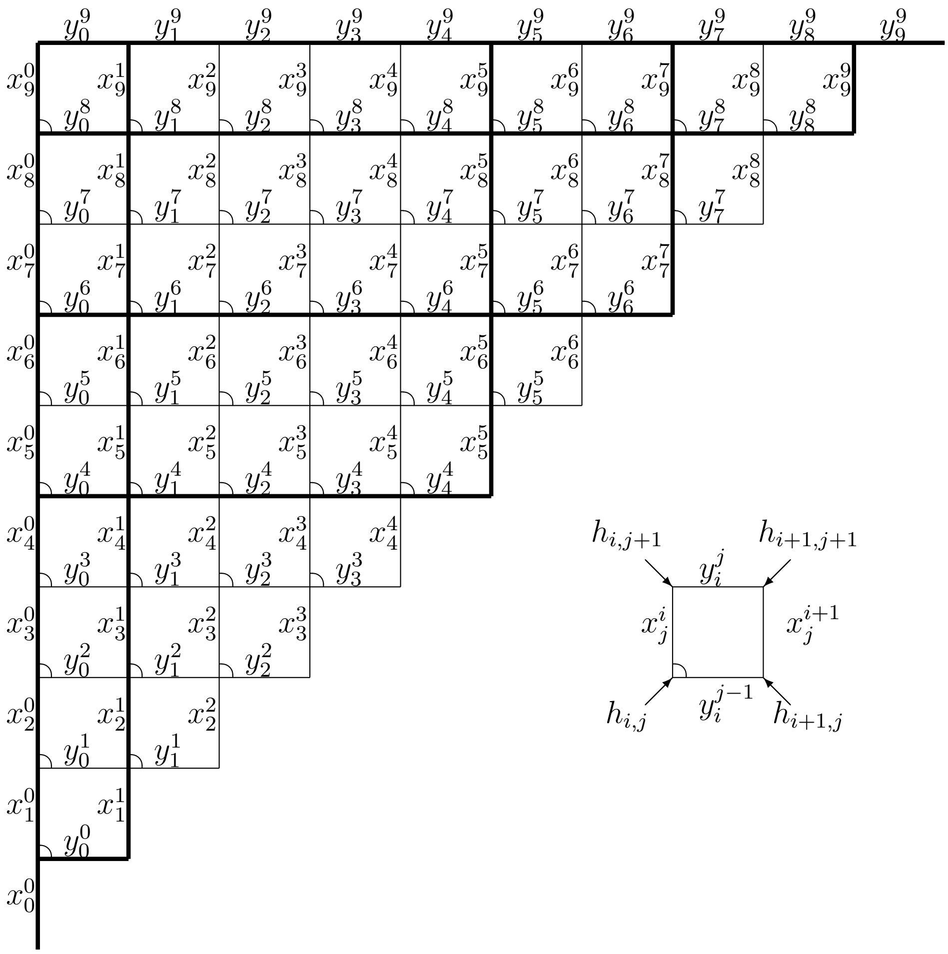


Figure 9.1: A commutation grid

The elements $h_{i,j}$ from M marking the corners of the small square in Figure 9.1 are defined by

$$h_{i,j} := (x_0^0 x_1^0 \dots x_{j-1}^0) \cdot (y_0^{j-1} y_1^{j-1} \dots y_{i-1}^{j-1}).$$

Then $h_{i,j+1} = h_{i,j} \cdot x_j^i$ and $h_{i+1,j} = h_{i,j} \cdot y_i^{j-1}$. Hence by Lemma 7.1.1, $h_{i,j} = h_{i,j+1} \wedge h_{i+1,j}$ and $h_{i+1,j+1} = h_{i,j+1} \vee h_{i+1,j}$. These relations will be used in the proof of the following lemma. It can be read as the converse of the equality $(x_0^0 x_1^0 \dots x_n^0) \cdot (y_0^n y_1^n \dots y_n^n) = (x_0^0 y_0^0)(x_1^1 y_1^1)(x_2^2 y_2^2) \dots (x_n^n y_n^n)$: Whenever the product of two elements of M equals the product of finitely many elements, there exists a corresponding commutation grid. This lemma is the announced extension of Levi's Lemma from trace theory into our setting of divisibility monoids.

Lemma 9.1.3 *Let $z_0, z_1, \dots, z_n, x, y \in M$ with $x \cdot y = z_0 \cdot z_1 \cdots z_n$. Then there* *exists a commutation grid* $(x^i_j, y^j_i)_{0\le i\le j\le n}$ *in* M *such that*

- $x = x^0_0 x^0_1 \dots x^0_n$,
- $y = y^n_0 y^n_1 \dots y^n_n$, *and*
- $z_i = x^i_i y^i_i$ *for* $i = 0, 1, \dots, n$.

Proof. Let $h_{0,j} := x \wedge z_0 z_1 \dots z_{j-1}$ and $h_{j,j} = z_0 \cdot z_1 \dots z_{j-1}$ (for $0 \le j \le n+1$). Note that $h_{0,j} \le h_{0,j+1}$ and $h_{j,j} \le h_{j+1,j+1}$. Furthermore, let $h_{i,j} := h_{0,j} \vee h_{i,i}$ for $0 < i < j \le n+1$ (this supremum exists since $\{h_{0,j}, h_{i,i}\}$ is bounded by xy). Then $h_{i+1,j+1} = h_{0,j+1} \vee h_{i+1,i+1} = h_{0,j+1} \vee h_{0,j} \vee h_{i+1,i+1} \vee h_{i,i} = h_{i,j+1} \vee h_{i+1,j}$.

Now let $0 \le i < j \le n+1$. Then $h_{0,j+1} \wedge h_{i+1,i+1} = x \wedge z_0 z_1 \cdots z_j \wedge z_0 z_1 \cdots z_i = h_{0,i+1} \le h_{0,j}$, and so

$$\begin{aligned} h_{i,j+1} \wedge h_{i+1,j} &= (h_{0,j+1} \vee h_{i,i}) \wedge (h_{0,j} \vee h_{i+1,i+1}) \\ &= (h_{0,j+1} \wedge h_{0,j}) \vee (h_{0,j+1} \wedge h_{i+1,i+1}) \\ &\quad \vee (h_{i,i} \wedge h_{0,j}) \vee (h_{i,i} \wedge h_{i+1,i+1}) \\ &= h_{0,j} \vee h_{0,i+1} \vee (h_{i,i} \wedge h_{0,j}) \vee h_{i,i} \\ &= h_{0,j} \vee h_{i,i} \qquad \text{(since } h_{0,i+1} \le h_{0,j} \text{ and } h_{i,i} \wedge h_{0,j} \le h_{i,i}) \\ &= h_{i,j}. \end{aligned}$$

Now the elements x^i_j and y^j_i for $0 \le i \le j \le n$ of the commutation grid are the monoid elements uniquely determined by $h_{i,j+1} = h_{i,j} \cdot x^i_j$ and $h_{i+1,j+1} = h_{i,j+1} \cdot y^j_i$.

Since $h_{i,j+1} \wedge h_{i+1,j} = h_{i,j}$ and $h_{i+1,j+1} = h_{i,j+1} \vee h_{i+1,j}$, Lemma 7.1.1 implies $x^i_j \wedge y^{j-1}_i = 1$ and $x^i_j \vee y^{j-1}_i = x^i_j y^j_i = y^{j-1}_i x^{i+1}_j$. Hence $x^i_j \,\mathcal{I}\, y^{j-1}_i$, $x^i_j \uparrow y^{j-1}_i = x^{i+1}_j$ and $y^{j-1}_i \uparrow x^i_j = y^j_i$. Thus, we showed that $(x^i_j, y^j_i)_{0\le i\le j\le n}$ is a commutation grid in M.

Note that $x^0_0 x^0_1 \dots x^0_n = h_{0,n+1} = x \wedge z_0 z_1 \dots z_n = x \wedge xy = x$. Furthermore, $h_{j,j} \cdot z_j = h_{j+1,j+1} = h_{j,j} \cdot x^j_j \cdot y^j_j$ implies $z_j = x^j_j \cdot y^j_j$ since M is cancellative. Hence we have

$$\begin{aligned} x(y^n_0 y^n_1 \dots y^n_n) &= (x^0_0 x^0_1 \dots x^0_n)(y^n_0 y^n_1 \dots y^n_n) \\ &= (x^0_0 y^0_0)(x^1_1 y^1_1)(x^2_2 y^2_2) \dots (x^n_n y^n_n) \\ &= z_0 z_1 \dots z_n = xy. \end{aligned}$$

This equality implies $y = y^n_0 y^n_1 \dots y^n_n$. □

It is reasonable that the elements y^j_i in a commutation grid do not completely determine the elements x^i_j. The following lemma describes the freedom we have in choosing these elements: as long as we keep the commutation behaviors in the first column, we can complete the commutation grid.

Lemma 9.1.4 *Let* $(x^i_j, y^j_i)_{0\le i\le j\le n}$ *be a commutation grid in* M*, and, for* $0 \le j \le n$*, let* $w^0_j \in M$ *with* $c_{x^0_j} = c_{w^0_j}$*. Then there exists a commutation grid* $(w^i_j, y^j_i)_{0\le i\le j\le n}$ *in* M*.*

Proof. By Lemma 7.2.5(1), $x^0_j \,\mathcal{I}\, y^{j-1}_0 y^{j-1}_1 \dots y^{j-1}_{j-1}$ for any $0 \le j \le n$. Since the commutation behaviors of x^0_j and w^0_j coincide, this implies $w^0_j \,\mathcal{I}\, y^{j-1}_0 y^{j-1}_1 \dots y^{j-1}_{i-1}$ for $0 \le i \le j \le n$. Hence $w^i_j := w^0_j \uparrow (y^{j-1}_0 y^{j-1}_1 \dots y^{j-1}_{i-1})$ is defined. Using Lemma 7.2.5(1), we get $w^i_j \,\mathcal{I}\, y^{j-1}_i$. By Lemma 7.2.5(2), $w^{i+1}_j = w^i_j \uparrow y^{j-1}_i$. Since $x^i_j = x^0_j \uparrow y^{j-1}_0 y^{j-1}_1 \dots y^{j-1}_{i-1}$ and $c_{x^0_j} = c_{w^0_j}$, Lemma 7.2.5(4) implies $c_{x^i_j} = c_{w^i_j}$. Hence $y^j_i = c_{x^i_j}(y^{j-1}_i) = c_{w^i_j}(y^{j-1}_i) = y^{j-1}_i \uparrow w^i_j$. □

Recall that $\mathrm{nat}(d_v(u)) = c_{\mathrm{nat}(v)}(\mathrm{nat}(u))$ for any words u, v by Lemma 7.2.7. Using Lemma 9.1.2, one gets as a direct consequence

Corollary 9.1.5 *Let* $(u^i_j, v^j_i)_{0\le i\le j\le n}$ *be a commutation grid in* $T^\star$*, and, for* $0 \le j \le n$*, let* $w^0_j \in T^\star$ *with* $d_{u^0_j} = d_{w^0_j}$*. Then there exists a commutation grid* $(w^i_j, v^j_i)_{0\le i\le j\le n}$ *in* $T^\star$*.*

Similarly as above, a direct consequence of Lemma 9.1.3 is the existence of commutation grids in $T^\star$:

Corollary 9.1.6 *Let* $z_0, z_1, \dots, z_n, u, v \in T^\star$ *with* $\mathrm{nat}(uv) = \mathrm{nat}(z_1 z_2 \dots z_n)$*. Then there exists a commutation grid* $(u^i_j, v^j_i)_{0\le i\le j\le n}$ *in* $T^\star$ *such that*

1. $\mathrm{nat}(u) = \mathrm{nat}(u^0_0 u^0_1 \dots u^0_n)$,
2. $\mathrm{nat}(v) = \mathrm{nat}(v^n_0 v^n_1 \dots v^n_n)$*, and*
3. $\mathrm{nat}(z_i) = \mathrm{nat}(u^i_i v^i_i)$ *for* $i = 0, 1, \dots, n$.

Note that the equations in the corollary above do not hold in the free monoid $T^\star$ but only in the divisibility monoid M. But if the words z_i are actually from $T \cup \{\varepsilon\}$ (i.e. their length is at most 1), we can replace the third statement by

3'. $z_i = u^i_i v^i_i$ for $i = 0, 1, \dots, n$.

Now we can introduce the notion of rank in the present context. For traces, it was defined and shown to be very useful by Hashigushi [Has91], cf. [DR95, Ch. 6] and [DM97]. Recall that $\mathrm{nat}(X) = \{\mathrm{nat}(w) \mid w \in X\}$ for any set $X \subseteq T^\star$ of words over T.

Definition 9.1.7 Let $u, v \in T^\star$ and $X \subseteq T^\star$ such that $\mathrm{nat}(uv) \in \mathrm{nat}(X)$. Let $\mathrm{rk}(u, v, X)$, the *rank of* u *and* v *relative to* X, denote the minimal integer n such that there exists a commutation grid $(u^i_j, v^j_i)_{0\le i\le j\le n}$ in $T^\star$ with

1. $\mathrm{nat}(u) = \mathrm{nat}(u^0_0 u^0_1 \dots u^0_n)$,
2. $\mathrm{nat}(v) = \mathrm{nat}(v^n_0 v^n_1 \dots v^n_n)$,
3. $u^0_0 v^0_0 u^1_1 v^1_1 \dots u^n_n v^n_n \in X$.

Let $u, v \in T^\star$ and $X \subseteq T^\star$ such that $\mathrm{nat}(uv) \in \mathrm{nat}(X)$. Then there exists $z \in X$ with $\mathrm{nat}(z) = \mathrm{nat}(uv)$. Let $z = z_1 z_2 \dots z_n$ with $z_i \in T$. Then $n = |uv|$ and by Corollary 9.1.6 (with 3' instead of 3) we find an appropriate commutation grid. Hence $\mathrm{rk}(u, v, X) \le |uv|$. If not only $\mathrm{nat}(uv) \in \mathrm{nat}(X)$ but even $uv \in X$ we can choose $n = 0$, $u^0_0 = u$ and $v^0_0 = v$ and obtain $\mathrm{rk}(u, v, X) = 0$. We define the *rank* $\mathrm{rk}(X)$ *of* X by

$$\mathrm{rk}(X) := \sup\{\mathrm{rk}(u, v, X) \mid u, v \in T^\star, \mathrm{nat}(uv) \in \mathrm{nat}(X)\} \in \mathbb{N} \cup \{\infty\}.$$

A word language $X \subseteq T^\star$ is *closed* if $\mathrm{nat}(u) \in \mathrm{nat}(X)$ implies $u \in X$ for any $u \in T^\star$. Since $\mathrm{rk}(u, v, X) = 0$ whenever $uv \in X$, the rank of a closed language equals 0.

Theorem 9.1.8 *Let $(M, \cdot, 1)$ be a divisibility monoid with finite commutation behavior (i.e. the monoid $\mathbb{C}_M$ is finite). Let $X \subseteq T^\star$ be a recognizable language of finite rank. Then* $\mathrm{nat}(X)$ *is recognizable in* M.

Proof. Let $n = \mathrm{rk}(X) \in \mathbb{N}$ be the rank of X. Since X is recognizable, there is a finite monoid S and a homomorphism $\eta : T^\star \to S$ that recognizes X. Since the mapping $u \mapsto d_u$ is an antihomomorphism from $T^\star$ into the finite monoid $(\mathbb{D}_M, \circ, d_\varepsilon)$, we may assume that $\eta(u) = \eta(v)$ implies $d_u = d_v$.

For $x \in M$, let $R(x)$ denote the set

$$\{(\eta d(u_0), \eta d(u_1) \dots \eta d(u_n))_{d\in\mathbb{D}_M} \mid u_0, u_1, \dots, u_n \in T^\star \text{ and } x = \mathrm{nat}(u_0 u_1 \dots u_n)\}.$$

Then $R(x)$ is a subset of $(S^{n+1})^{|\mathbb{D}_M|}$. Since $\mathbb{D}_M$ and S are finite, there are only finitely many sets $R(x)$. Once we will have shown

$$R(x) = R(z) \Rightarrow x^{-1}\,\mathrm{nat}(X) = z^{-1}\,\mathrm{nat}(X),$$

we thus have that $\{x^{-1}\,\mathrm{nat}(X) \mid x \in M\}$ is finite. Hence $\mathrm{nat}(X)$ is recognizable.

So let $R(x) = R(z)$ and let $y \in x^{-1}\,\mathrm{nat}(X)$, i.e. $xy \in \mathrm{nat}(X)$. Since $\mathrm{rk}(X) = n$, there exists a commutation grid $(u^i_j, v^j_i)_{0\le i\le j\le n}$ in $T^\star$ such that

- $x = \mathrm{nat}(u^0_0 u^0_1 \dots u^0_n)$,

- $y = \mathrm{nat}(v_0^n v_1^n \dots v_n^n)$, and
- $u_0^0 v_0^0 u_1^1 v_1^1 \dots u_n^n v_n^n \in X$.

Then $(\eta d(u_0^0), \eta d(u_1^0) \dots \eta d(u_n^0))_{d \in \mathbb{D}_M} \in R(x) = R(z)$. Hence there exist words $w_j^0 \in T^\star$ with

(1) $\eta d(w_j^0) = \eta d(u_j^0)$ for each $0 \leq j \leq n$ and $d \in \mathbb{D}_M$, and

(2) $z = \mathrm{nat}(w_0^0 w_1^0 \dots w_n^0)$.

In (1), consider $d = d_\varepsilon$ which equals the identity on $T^\star$. Then $\eta(w_j^0) = \eta(u_j^0)$ and therefore (by our assumption on η) $d_{w_j^0} = d_{u_j^0}$. Hence we can apply Corollary 9.1.5 and obtain a commutation grid $(w_j^i, v_i^j)_{0 \leq i \leq j \leq n}$ in $T^\star$. Now consider $d = d_{v_0^{j-1} v_1^{j-1} \dots v_{j-1}^{j-1}} \in \mathbb{D}_M$. Note that $w_j^j = w_j^0 \uparrow (v_0^{j-1} v_1^{j-1} \dots v_{j-1}^{j-1}) = d(w_j^0)$. Hence $\eta(w_j^j) = \eta d(w_j^0) = \eta d(u_j^0) = \eta(u_j^j)$. Now we can conclude

$$\begin{aligned} \eta(w_0^0 v_0^0 w_1^1 v_1^1 \dots w_n^n v_n^n) &= \eta(w_0^0)\eta(v_0^0)\eta(w_1^1)\eta(v_1^1) \dots \eta(w_n^n)\eta(v_n^n) \\ &= \eta(u_0^0)\eta(v_0^0)\eta(u_1^1)\eta(v_1^1) \dots \eta(u_n^n)\eta(v_n^n) \\ &= \eta(u_0^0 v_0^0 u_1^1 v_1^1 \dots u_n^n v_n^n) \in \eta(X). \end{aligned}$$

Hence $w_0^0 v_0^0 w_1^1 v_1^1 \dots w_n^n v_n^n \in X$. Since $(\mathrm{nat}(w_j^i), \mathrm{nat}(v_i^j))_{0 \leq i \leq j \leq n}$ is a commutation grid in M, we obtain $zy = \mathrm{nat}(w_0^0 v_0^0 w_1^1 v_1^1 \dots w_n^n v_n^n) \in \mathrm{nat}(X)$. Hence $y \in z^{-1} \mathrm{nat}(X)$ and therefore $x^{-1} \mathrm{nat}(X) = z^{-1} \mathrm{nat}(X)$ as claimed above. □

9.2 From c-rational to recognizable languages

In this section, we prove closure properties of the set of recognizable languages in a divisibility monoid. These closure properties correspond to c-rational languages that we introduce first:

Let $(M, \cdot, 1)$ be a divisibility monoid. An element $x \in M$ is *connected* if the distributive lattice $\downarrow x$ does not contain any pair of complementary elements. In other words, there are *no* independent $y, z \in M \setminus \{1\}$ such that $x = y \vee z = yc_y(z)$. A set $L \subseteq M$ is *connected* if all of its elements are connected; a language $X \subseteq T^\star$ is *connected* if $\mathrm{nat}(X) \subseteq M$ is connected.

Let t be a trace over the dependence alphabet (Σ, D). In trace theory, this trace is called "connected" if the letters occurring in it induce a connected subgraph of (Σ, D). One can easily check that this is the case iff t is not the supremum of two independent traces, i.e. iff t is connected in the sense defined above. For rational trace languages, to be recognizable it suffices that the iteration is applied to connected languages, only. In other words, there is a subset of the trace monoid C (the connected traces) such that the iteration is applied to languages included in C, only. Already for concurrency monoids (cf. [Dro95, Dro96]), it is not sufficient to restrict to connected languages. But there, one still has finitely many pairwise disjoint sets C_q such that the iteration can be restricted to subsets of C_q. For divisibility monoids, we did not find such sets in general (for labeled divisibility monoids, they exist – see below). Therefore, we impose an internal condition on those languages that we want to iterate:

A language $X \subseteq T^\star$ is *residually closed* if for any $u \in X$ and $v \in T^\star$ with $u \,\mathcal{I}\, v$ the following holds:

$$v \in X \iff d_u(v) \in X.$$

Thus X is residually closed if it is closed under the application of d_u and d_u^{-1} for elements u of X. Note that this need not hold for all $u \in T^\star$. A language $L \subseteq M$ is *residually closed* iff $\{w \in T^\star \mid \text{nat}(w) \in L\}$ is residually closed.

Now we define c-rational languages: The set of *c-rational sets* in a divisibility monoid M is the least class $\mathfrak{C} \subseteq 2^M$ such that

- all finite subsets of M belong to $\mathfrak{C}$,
- $X \cdot Y$ and $X \cup Y$ belong to $\mathfrak{C}$ whenever $X, Y \in \mathfrak{C}$, and
- $\langle X \rangle$ belongs to $\mathfrak{C}$ whenever $X \in \mathfrak{C}$ is connected and residually closed.

Now we are going to show that the set of recognizable languages is closed under multiplication.

Lemma 9.2.1 *Let* $(M, \cdot, 1)$ *be a divisibility monoid and* $X, Y \subseteq T^\star$ *be closed. Then* $\text{rk}(XY) \leq 1$.

Proof. Let $u, v \in T^\star$ with $\text{nat}(uv) \in \text{nat}(XY)$. Then there exist $z_0 \in X$ and $z_1 \in Y$ such that $\text{nat}(uv) = \text{nat}(z_0 z_1)$. By Corollary 9.1.6, there exists a commutation grid $(u_j^i, v_i^j)_{0 \leq i \leq j \leq 1}$ in $T^\star$ such that

$$\begin{array}{ll} \text{nat}(u) = \text{nat}(u_0^0 u_1^0), & \text{nat}(v) = \text{nat}(v_0^1 v_1^1), \\ \text{nat}(z_0) = \text{nat}(u_0^0 v_0^0), \text{ and} & \text{nat}(z_1) = \text{nat}(u_1^1 v_1^1). \end{array}$$

Since X and Y are closed, this implies $u_0^0 v_0^0 \in X$ and $u_1^1 v_1^1 \in Y$. Hence $\text{rk}(u, v, XY) \leq 1$. □

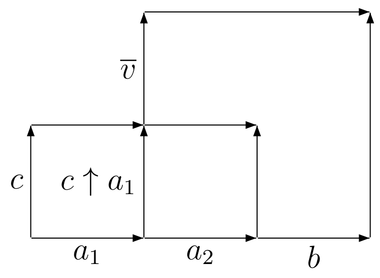


Figure 9.2: The elements from Lemma 9.2.3

Corollary 9.2.2 *Let $(M, \cdot, 1)$ be a divisibility monoid with finite commutation behavior. Let $K, L \subseteq M$ be recognizable. Then $K \cdot L$ is recognizable.*

Proof. Let $X := \{u \in T^\star \mid \mathrm{nat}(u) \in K\}$ and $Y := \{u \in T^\star \mid \mathrm{nat}(u) \in L\}$. Then $K \cdot L = \mathrm{nat}(XY)$, and X and Y are closed and recognizable in $T^\star$. Hence, by Kleene's Theorem, XY is recognizable in $T^\star$. By Lemma 9.2.1, the rank of XY is finite. Hence Theorem 9.1.8 ensures that $\mathrm{nat}(XY)$ is recognizable in M. □

The rest of this section is devoted to the proof that $\langle L \rangle$ is recognizable for any recognizable language $L \subseteq M$ that is connected and residually closed. But first, we prove some technical lemmas that will be used later on.

Lemma 9.2.3 *Let $(M, \cdot, 1)$ be a divisibility monoid with finite commutation behavior. Let $a_1, a_2, b, c, \overline{v} \in T^\star$ such that $d_{a_1a_2} = d_b \subseteq \mathrm{id}_{T^\star}$, $c \,\mathcal{I}\, a_1$, $(c \uparrow a_1)\overline{v} \,\mathcal{I}\, a_2b$ and $a_1 \neq \varepsilon \neq \overline{v}$. Then $\mathrm{nat}((a_1 \uparrow c)\overline{v})$ is not connected.*

Proof. Throughout this proof, we make extensive use of the equations given in Lemma 7.2.7 without mentioning it again.

By $(c{\uparrow}a_1)\overline{v}\mathcal{I}a_2b$, we get in particular $(c{\uparrow}a_1)\overline{v}\mathcal{I}a_2$, and $(c{\uparrow}a_1)\overline{v}{\uparrow}a_2 \in \mathrm{dom}(d_b)$. Furthermore, it implies $c \uparrow a_1 \,\mathcal{I}\, a_2$ and therefore (together with $c \,\mathcal{I}\, a_1$) $c \,\mathcal{I}\, a_1a_2$

and $c \uparrow a_1 a_2 = c$. Now we can conclude (with $\tilde{v} := \overline{v} \uparrow (a_2 \uparrow (c \uparrow a_1)))$

$$\begin{aligned} b\,\mathcal{I}\,(c \uparrow a_1)\overline{v} \uparrow a_2 &= ((c \uparrow a_1) \uparrow a_2)\ (\overline{v} \uparrow (a_2 \uparrow (c \uparrow a_1))) \\ &= (c \uparrow a_1 a_2)\tilde{v} \\ &= c\tilde{v}. \end{aligned}$$

Thus we get $c\tilde{v} = (c \uparrow a_1)\overline{v} \uparrow a_2 \in \mathrm{dom}(d_{a_1 a_2})$ and therefore $c\tilde{v} \uparrow a_1 a_2 = c\tilde{v}$. In particular, we have $a_1\,\mathcal{I}\,c\tilde{v}$ implying $(a_1 \uparrow c)\,\mathcal{I}\,\tilde{v}$.

Hence we have $(c{\uparrow}a_1)\overline{v}{\uparrow}a_2 = c\tilde{v} = (c\tilde{v}{\uparrow}a_1){\uparrow}a_2$. This implies $(c{\uparrow}a_1)\overline{v} = c\tilde{v}{\uparrow}a_1$ since d_{a_2} is injective. Note that $c\tilde{v}{\uparrow}a_1 = (c{\uparrow}a_1)(\tilde{v}{\uparrow}(a_1{\uparrow}c))$. Hence by cancellation we get $\overline{v} = \tilde{v} \uparrow (a_1 \uparrow c)$. This implies $\mathrm{nat}(\tilde{v}) \vee \mathrm{nat}(a_1 \uparrow c) = \mathrm{nat}((a_1 \uparrow c)\overline{v})$.

Since $a_1 \neq \varepsilon$ and d_c is length preserving, we have $a_1{\uparrow}c \neq \varepsilon$, i.e. $\mathrm{nat}(a_1{\uparrow}c) \neq 1$. Similarly, $\overline{v} \neq \varepsilon$ implies $\mathrm{nat}(\tilde{v}) \neq 1$. Finally $(a_1 \uparrow c)\,\mathcal{I}\,\tilde{v}$ proves that $\mathrm{nat}((a_1 \uparrow c)\overline{v})$ is not connected. □

Before proving that the iteration of a residually closed and connected language in M has finite rank, we cite a special case of Ramsey's Theorem (cf. [Cam94] for the general formulation):

Ramsey's Theorem [Ram30] *Let c, r be positive integers. Then there is a positive integer $R_r(c)$ such that for any mapping d of the two-elements subsets of $[R_r(c)]$ into $[c]$ there exists an r-elements subset $A \subseteq [R_r(c)]$ such that we have $d(B) = d(C)$ for any two-elements subsets B and C of A.*

First we use Ramsey's Theorem to show that in a divisibility monoid with finite commutation behavior, for any sufficiently long sequence $u_1, u_2, \ldots u_n$ of elements of $T^\star$, there is a nonempty fragment of this sequence such that the commutation behavior of $u_i u_{i+1} \ldots u_j$ is contained in the identity.

Lemma 9.2.4 *Let M be a divisibility monoid with finite commutation behavior and let $u_i \in T^\star$ for $1 \leq i \leq R_3(|\mathbb{D}_M|)$. Then there exist $1 \leq i < j < k \leq R_3(|\mathbb{D}_M|)$ such that $d_{u_i u_{i+1} \ldots u_{j-1}} = d_{u_j u_{j+1} \ldots u_{k-1}} \subseteq \mathrm{id}_{T^\star}$.*

Proof. For simplicity, let $n = R_3(|\mathbb{D}_M|)$. Consider the mapping d' from the 2-elements subsets of $[n]$ into $\mathbb{D}_M$ with $d'(\{i,j\}) = d_{u_i u_{i+1} \ldots u_{j-1}}$ where $i < j$. By Ramsey's Theorem, there are $1 \leq i < j < k \leq n$ with $d'(\{i,j\}) = d'(\{i,k\}) = d'(\{j,k\}) =: f$. Note that f is an idempotent partial function since $d'(\{i,k\}) = d'(\{j,k\}) \circ d'(\{i,j\})$. In addition, f is injective on its domain by Lemma 7.2.7, implying $f \subseteq \mathrm{id}_{T^\star}$. □

Lemma 9.2.5 *Let M be a divisibility monoid with finite commutation behavior, $(u^i_j, v^j_i)_{1\le i\le j\le n}$ a commutation grid in $T^\star$ with $n \ge R_3(|\mathbb{D}_M|)$ and $u^i_i \neq \varepsilon \neq v^i_i$. Then there exists $1 \le i \le n$ such that* $\mathrm{nat}(u^i_i v^i_i)$ *is not connected.*

Proof. By Lemma 9.2.4, there are $1 \le i < j < j \le n$ with

$$d_{u^0_i u_{i+1}0\ldots u^0_{j-1}} = d_{u^0_j u_{j+1}0\ldots u^0_{k-1}} = d_{u^0_i u_{i+1}0\ldots u^0_{k-1}}.$$

With $a_1 = u^0_i$, $a_2 = u^0_{i+1}u^0_{i+2}\ldots u^0_{j-1}$, $x = u^0_j u_{j+1}0 \ldots u^0_{k-1}$, $y = v^{i-1}_0 v^{i-1}_1 \ldots v^{i-1}_{i-1}$ and $\overline{v} = v^i_i$, the assumptions of Lemma 9.2.3 are satisfied. Hence $\mathrm{nat}(u^i_i v^i_i)$ is not connected. □

Theorem 9.2.6 *Let $(M, \cdot, 1)$ be a divisibility monoid with finite commutation behavior. Let $X \subseteq T^\star$ be closed, connected, and residually closed. Then the rank* $\mathrm{rk}(\langle X\rangle)$ *of X is at most $R_3(|\mathbb{D}_M|) + 1$ and therefore finite.*

Proof. Let $u, v \in T^\star$ with $\mathrm{nat}(uv) \in \mathrm{nat}(\langle X\rangle) = \langle \mathrm{nat}(X)\rangle$. Then there exist $n \in \mathbb{N}$ and $x_0, x_1, \ldots, x_n \in X$ such that $\mathrm{nat}(uv) = \mathrm{nat}(x_0x_1\ldots x_n)$. By Corollary 9.1.6, there exists a commutation grid $(u^i_j, v^j_i)_{0\le i\le j\le n}$ in $T^\star$ such that $\mathrm{nat}(u) = \mathrm{nat}(u^0_0u^0_1\ldots u^0_n)$, $\mathrm{nat}(v) = \mathrm{nat}(v^n_0 v^n_1 \ldots v^n_n)$ and $\mathrm{nat}(x_i) = \mathrm{nat}(u^i_i v^i_i)$. Since X is closed and connected, the latter implies that $u^i_i v^i_i \in X$ is connected.

There exist $m \le n$ and $-1 = i_0 < i_1 < i_2 \cdots < i_m < i_{m+1} = n+1$ such that

- $u^{i_k}_{i_k}$ and $v^{i_k}_{i_k}$ are nonempty for $0 < k < m+1$,
- u^i_i or v^i_i is empty for $0 \le i \le n$ with $i \notin \{i_0, i_1, \ldots, i_{m+1}\}$.

Consider the subgrid $(a^i_j, b^j_i)_{0\le l\le \ell\le m}$ induced by the sequence $(0, i_1, i_2, \ldots, i_m, n)$. Then $a^k_k = u^{i_k}_{i_k} u^{i_k}_{i_k+1} \ldots u^{i_k}_{i_{k+1}-1}$ and $b^k_k = v^{i_{k+1}-1}_{i_k} v^{i_{k+1}-1}_{i_k+1} \ldots v^{i_{k+1}-1}_{i_{k+1}-1}$ are not empty. Hence, by Lemma 9.2.5, $m \le R_3(|\mathbb{D}_M|)$.

Let $0 \le k \le m$ and $i_k < i < i_{k+1}$. Then $u^i_i v^i_i \in X$. Since one of u^i_i and v^i_i is empty, the other belongs to X, i.e. $u^i_i, v^i_i \in X \cup \{\varepsilon\}$.

Now we show $u^i_j, v^j_i \in X \cup \{\varepsilon\}$ by induction on j for $i_k < i \le j < i_{k+1}$:

Assume $i_k + 1 = j$. Then the claim is trivial since $i = j$ follows. Now assume that for any $i_k \le i \le l \le i_{k+1}$ with $l < j$ we have $u^i_l, v^l_i \in X \cup \{\varepsilon\}$. Then $u^i_j {\uparrow} (v^{j-1}_i v^{j-1}_{i+1} \ldots v^{j-1}_{j-1}) = u^j_j \in X \cup \{\varepsilon\}$. Note that the upper index $j-1$ of the v's is properly between i_k and j. Hence by the induction hypothesis $v^{j-1}_{i'} \in X \cup \{\varepsilon\}$ for $i' = i, i+1, \ldots, j-1$. Since X and therefore $X \cup \{\varepsilon\}$ is residually closed, we get $u^i_j \in X \cup \{\varepsilon\}$. On the other hand, $v^j_i = v^{j-1}_i {\uparrow} u^i_j$. By the induction hypothesis, $v^{j-1}_i \in X \cup \{\varepsilon\}$. Hence $v^j_i \in X \cup \{\varepsilon\}$ since u^i_j is an element of this residually closed language.

Now consider the subgrid $(a^i_j, b^i_j)_{0\leq i\leq j\leq 2m+1}$ that is generated by the sequence $(i_1, i_1+1, i_2, i_2+1, \ldots, i_m, i_m+1)$. Then

$$\begin{aligned} a^{2k}_{2k} &= u^{i_k+1}_{i_k+1} u^{i_k+1}_{i_k+2} u^{i_k+1}_{i_k+3} \ldots u^{i_k+1}_{i_{k+1}-1}, \\ b^{2k}_{2k} &= v^{i_{k+1}-1}_{i_k+1} v^{i_{k+1}-1}_{i_k+2} v^{i_{k+1}-1}_{i_k+3} \ldots v^{i_{k+1}-1}_{i_{k+1}-1}, \\ a^{2k+1}_{2k+1} &= u^{i_k}_{i_k}, \text{ and} \\ b^{2k+1}_{2k+1} &= v^{i_k}_{i_k}. \end{aligned}$$

Note that all the factors of a^{2k}_{2k} and of b^{2k}_{2k} belong to $X \cup \{\varepsilon\}$. This implies $a^{2k}_{2k} b^{2k}_{2k} \in \langle X\rangle$. Thus, the commutation grid $(a^i_j, b^j_i)_{0\leq i\leq j\leq 2m+1}$ satisfies

- $\mathrm{nat}(u) = \mathrm{nat}(u^0_0 u^0_1 \ldots u^0_n) = \mathrm{nat}(a^0_0 a^0_1 \ldots a^0_{2m+1})$,
- $\mathrm{nat}(v) = \mathrm{nat}(v^n_0 v^n_1 \ldots v^n_n) = \mathrm{nat}(b^{2m+1}_0 b^{2m+1}_1 \ldots b^{2m+1}_{2m+1})$ and
- $a^i_i b^i_i \in \langle X\rangle$.

Therefore $\mathrm{rk}(u, v, \langle X\rangle) \leq 2m + 1 \leq 2R_3(|\mathbb{D}_M|) + 1$. □

Corollary 9.2.7 *Let $(M, \cdot, 1)$ be a divisibility monoid with finite commutation behavior. Let $L \subseteq M$ be connected, recognizable, and residually closed. Then the iteration $\langle L\rangle$ of L is recognizable.*

Proof. Let $X := \{w \in T^\star \mid \mathrm{nat}(w) \in L\}$. Note that M and X satisfy the assumptions of Theorem 9.2.6. Hence the rank $\mathrm{rk}(X)$ is finite. By the theorem of Kleene, $\langle X\rangle$ is recognizable in $T^\star$. By Theorem 9.1.8, $\langle L\rangle = \mathrm{nat}(\langle X\rangle)$ is recognizable in M. □

Summarizing the results for obtained so far, we can show that any c-rational language is recognizable.

Theorem 9.2.8 *Let $(M, \cdot, 1)$ be a divisibility monoid with finite commutation behavior. Let $L \subseteq M$ be c-rational. Then L is recognizable.*

Proof. By Lemma 7.2.3, $\downarrow x$ is finite for any $x \in M$. Hence finite languages in M are recognizable. By Corollary 9.2.2 and 9.2.7, the set of recognizable languages in M is closed under products and the operation $\langle . \rangle$ applied to connected and residually closed languages. □

9.3 From recognizable to c-rational languages

In this section, we will derive conditions on divisibility monoids M which are sufficient to ensure that all recognizable languages in M are c-rational. Let $(M, \cdot, 1)$ a divisibility monoid. Recall that an equation $\mathrm{nat}(ab) = \mathrm{nat}(cd)$ where a, b, c, d are irreducible generators of M states that the different sequential executions ab and cd give rise to the same effect. If now $a \neq c$, the effect of a in the execution cd has to be resumed by that of d. Therefore, we consider the least equivalence $\equiv$ on the irreducible generators of M identifying a and d that occur in an equation $ab = cd$ with $a \neq c$. To show that any recognizable language is c-rational, we need the property that $\mathrm{nat}(ab) = \mathrm{nat}(cd)$ and $a \equiv c$ imply $a = c$ for any irreducible elements $a, b, c, d \in T$. It is immediate that this is equivalent to the existence of a function $\rho : T \to E$ into some set E satisfying $\rho(s) = \rho(s \uparrow t)$ and $\rho(s) \neq \rho(t)$ for any $s, t \in T$ with $s\mathcal{I}t$. Such a function is called *labeling function*. Since T is finite, we can assume E to be finite, too. A divisibility monoid M together with a labeling function ρ is a *labeled divisibility monoid* (M, ρ). The *label sequence* of a word $u_0u_1 \dots u_n \in T^\star$ is the word $\rho(u_0)\rho(u_1)\dots\rho(u_n) \in E^\star$. We extend the mapping ρ to words over T by $\rho(tw) = \{\rho(t)\} \cup \rho(w)$. Hence ρ is a monoid morphism into the finite monoid $(2^E, \cup, \emptyset)$. By Lemma 8.2.1, $\mathrm{nat}(u) = \mathrm{nat}(v)$ implies $\rho(u) = \rho(v)$ for any $u, v \in T^\star$. Therefore, it is reasonable to define $\rho(\mathrm{nat}(u)) := \rho(u)$, i.e. to extend ρ to a monoid morphism from $(M, \cdot, 1)$ to $(2^E, \cup, \emptyset)$.

A language $L \subseteq M$ is *monoalphabetic* if $\rho(x) = \rho(y)$ for any $x, y \in L$. The class of *mc-rational languages* in the labeled divisibility monoid M is the smallest class $\mathfrak{C} \subseteq 2^M$ satisfying

- any finite subset of M is in $\mathfrak{C}$,
- whenever $L, K \in \mathfrak{C}$ then $L \cup K \in \mathfrak{C}$ and $L \cdot K \in \mathfrak{C}$, and
- whenever $L \in \mathfrak{C}$ is connected and monoalphabetic then $\langle L \rangle \in \mathfrak{C}$.

Note that differently from c-rational languages, here the iteration is restricted to connected and monoalphabetic languages that are not explicitly required to be residually closed. Nonetheless, any mc-rational language is c-rational as Corollary 9.3.2 states.

Lemma 9.3.1 *Let (M, ρ) be a labeled divisibility monoid and $x, y \in M$ with $x\mathcal{I}y$. Then $\rho(x) \cap \rho(y) = \emptyset$, $\rho(y) = \rho(y \uparrow x)$, and $\rho(x) \cup \rho(y) = \rho(x \vee y)$.*

Proof. By contradiction, assume $\rho(x) \cap \rho(y) \neq \emptyset$. Then there exist monoid elements $x_1, x_2, y_1, y_2 \in M$ and $s, t \in T$ such that $x = x_1 s x_2$, $y = y_1 t y_2$ and $\rho(s) = \rho(t)$. Clearly, $x_1 s \mathcal{I} y_1 t$. By Lemma 7.2.5(1) and (3), we have $s\, \mathcal{I}\, (y_1 t {\uparrow} x_1) = (y_1 {\uparrow} x_1)(t {\uparrow} (x_1 {\uparrow} y_1))$. Hence $s' := s {\uparrow} (y_1 {\uparrow} x_1) \mathcal{I} (t {\uparrow} (x_1 {\uparrow} y_1)) =: t'$ by Lemma 7.2.5(1). Furthermore, $\rho(s) = \rho(s')$ and $\rho(t) = \rho(t')$. But this contradicts the definition of a labeling function. Hence the first statement is shown.

By a simple induction on the length of x and y we get $\rho(y \uparrow x) = \rho(y)$, i.e. the second statement. Now the last assertion follows since $\rho(x \vee y) = \rho(x(y \uparrow x)) = \rho(x) \cup \rho(y \uparrow x) = \rho(x) \cup \rho(y)$. □

Corollary 9.3.2 *Let $(M, \cdot, 1, \rho)$ be a labeled divisibility monoid. Then any mc-rational language in M is c-rational.*

Proof. Let $x, y \in M$ with $\rho(x) = \rho(y)$ and $x \mathcal{I} y$. Then Lemma 9.3.1 ensures $x = y = 1$. Hence any monoalphabetic language is residually closed. This implies that mc-rational languages are c-rational. □

Now let $\preceq$ be a linear order on the set E and let $u \in T^\star$. We say that the word u *is in lexicographical normal form* if $\mathrm{nat}(u) = \mathrm{nat}(v)$ implies that the label sequence of v is lexicographically larger than or equal to that of u. Let LNF be the set of all words in lexicographic normal form.

Note that so far there may exist different words u and v in LNF with $\mathrm{nat}(u) = \mathrm{nat}(v)$. We show that this is impossible: So assume that $u, v \in T^\star$ have the same label sequence and satisfy $\mathrm{nat}(u) = \mathrm{nat}(v)$ Let s (t) be the first letter of u (of v, respectively) and suppose $s \neq t$. Then $\mathrm{nat}(s)$ and $\mathrm{nat}(t)$ are incomparable since they have the same length. Thus the infimum of them is properly smaller than $\mathrm{nat}(s)$. Since 1 is the only element which is properly smaller than the irreducible element $\mathrm{nat}(s)$, we get $\mathrm{nat}(s) \wedge \mathrm{nat}(t) = 1$. Since $\mathrm{nat}(s)$ and $\mathrm{nat}(t)$ are bounded above by $\mathrm{nat}(u) = \mathrm{nat}(v)$, we have $s \mathcal{I} t$. Hence $\rho(s) \neq \rho(t)$, contradicting the fact that the label sequences of u and v coincide. Hence $s = t$. Cancelling s and t from the left of u and v, respectively, we can proceed by induction. At the end, we obtain $u = v$. Hence, for any $x \in M$ there exists at most one $u \in \mathrm{LNF}$ such that $x = \mathrm{nat}(u)$. Since, on the other hand, the lexicographical order on $E^\star$ is a well-order, the set of label sequences of words u with $\mathrm{nat}(u) = x$ has a least element. Hence, for any $x \in M$ there exists a unique word $u \in \mathrm{LNF}$ with $x = \mathrm{nat}(u)$. This word is called *the lexicographic normal form of* x. We denote it by $\mathrm{lexNF}(x)$.

Next, we characterize the set of words in lexicographic normal form. This result generalizes the characterization of lexicographic normal forms for trace monoids given in [AK79].

Lemma 9.3.3 *Let* (M, ρ) *be a labeled divisibility monoid and let* $\preceq$ *be a linear order on* T. *Let* $u_i \in T$ *for* $0 \leq i \leq n$. *Then* $u_0 u_1 \dots u_n \in \mathrm{LNF}$ *iff*

$$u_j \in d_{u_i u_{i+1} \dots u_{j-1}}(T) \Rightarrow \rho(u_i) \prec \rho(u_j) \qquad (\star)$$

for $0 \leq i < j \leq n$.

Proof. For simplicity, let $u := u_0 u_1 \dots u_n$. First let $u \in \mathrm{LNF}$ and assume there are i, j with $0 \leq i < j \leq n$ such that $u_j \in d_{u_i u_{i+1} \dots u_{j-1}}(T)$ and $\rho(u_i) \succeq \rho(u_j)$. Then there is $t \in T$ with $u_j = d_{u_i u_{i+1} \dots u_{j-1}}(t)$. Hence

$$\mathrm{nat}(u) = \mathrm{nat}(u_0 \dots u_{i-1} t d_t(u_i \dots u_{j-1}) u_{j+1} \dots u_n)).$$

Since $\rho(u_i) \succeq \rho(u_j) = \rho(t)$, the label sequence of u is larger than or equals that of $u_0 \dots u_{i-1} t d_t(u_i \dots u_{j-1}) u_{j+1} \dots u_n)$. Since u is in lexicographical normal form, this implies in particular $\rho(u_i) = \rho(t)$, contradicting $t \in \mathrm{dom}(d_{u_i})$. Thus, u satisfies the property $(\star)$.

Conversely, let the word u satisfies the property $(\star)$. Let $v \in \mathrm{nat}(u)$ with $u \neq v$. We claim that u is lexicographically smaller than v. Note that any suffix of u satisfies $(\star)$. Hence we may assume that the first letter t of v is different from u_0. Then $\mathrm{nat}(t)$ and $\mathrm{nat}(u_0)$ are bounded above by $\mathrm{nat}(u)$. Since they are different irreducible elements in M, their infimum is trivial. Hence $\mathrm{nat}(t) \mathcal{I} \mathrm{nat}(u_0)$ implying $\rho(t) \neq \rho(u_0)$. Let j be the least integer such that $\rho(t) \in \rho(u_0 u_1 \dots u_j)$. By Lemma 9.3.1, $\mathrm{nat}(t)$ and $\mathrm{nat}(u_0 u_1 \dots u_j)$ are not independent. Since they are bounded by $\mathrm{nat}(u)$, the infimum cannot be 1. Hence $\mathrm{nat}(t) \leq \mathrm{nat}(u_0 u_1 \dots u_j)$. Since on the other hand the infimum of $\mathrm{nat}(t)$ and $\mathrm{nat}(u_0 u_1 \dots u_{j-1})$ is trivial, the supremum of these two equals $\mathrm{nat}(u_0 u_1 \dots u_j)$. Hence $u_j = d_{u_0 u_1 \dots u_{j-1}}(t)$. Since u satisfies $(\star)$, this implies $\rho(u_0) \prec \rho(u_j) = \rho(t)$ and hence our claim. Thus $u \in \mathrm{LNF}$. ☐

Using the lemma above, we show that the set of words in lexicographic normal form is recognizable:

Lemma 9.3.4 *Let* (M, ρ) *be a labeled divisibility monoid. Then* LNF *is recognizable in the free monoid* $T^\star$.

Proof. Recall that $\mathbb{D}_M = \{d_u \mid u \in T^\star\}$ is a monoid consisting of partial functions from $T^\star$ to $T^\star$. These functions are length preserving. In particular, they map elements of T to elements of T. Hence $\mathbb{D}_M \restriction T := \{d_u \restriction T \mid u \in T^\star\}$ is a monoid. It is finite since T is finite. Recall furthermore, that the mapping $T^\star \to \mathbb{D}_M$ defined by $u \mapsto d_u$ is a monoid antihomomorphism. Hence the mapping from $T^\star$ to $\mathbb{D}_M \restriction T$ with $u \mapsto d_u \restriction T$ is a monoid antihomomorphism, too. This implies that the sets $X_d := \{u \in T^\star \mid d_u \restriction T = d\}$ for $d \in \mathbb{D}_M \restriction T$ are recognizable in $T^\star$. Hence they are rational by Kleene's Theorem.

For $S \subseteq T$, $d \in \mathbb{D}_M$ or $d \in \mathbb{D}_M \restriction T$, let $d(S) := \{d(s) \mid s \in S \cap \mathrm{dom}(d)\}$ which is a finite set. Now by Lemma 9.3.3 the set of words over T that are not in lexicographical normal form equals the rational language

$$T^\star \setminus \mathrm{LNF} = \bigcup_{\substack{s\in T\\ d\in\mathbb{D}_M\restriction T}} T^\star\{s\}X_d\ (d_s \circ d)(\{t \in T \mid \rho(t) \preceq \rho(s)\})\ T^\star.$$

Hence LNF is recognizable. □

The crucial point in Ochmański's proof of the c-rationality of recognizable languages in trace monoids is that whenever a square of a word is in lexicographic normal form, it is actually connected. This does not hold any more for labeled divisibility monoids. But we can show that whenever a product of $|E| + 2$ words having the same set of labels is in lexicographic normal form, it is connected (cf. Corollary 9.3.6). This enables us to show that recognizable languages are mc-rational.

For a set $A \subseteq E$ and $u \in T^\star$ let $n_A(u)$ denote the number of maximal factors w of u with $\rho(w) \subseteq A$ or $\rho(w) \cap A = \emptyset$. The number $n_A(u)$ is the number of blocks of elements of A and of $E \setminus A$ in the label sequence of u. For example, let $u = u_1u_2 \dots u_n \in T^\star$ with $u_i \neq \varepsilon$, $\rho(u_{2i}) \subseteq A$ and $\rho(u_{2i+1}) \subseteq E \setminus A$ for all suitable i. Then $n = n_A(u) = n_{E\setminus A}(u)$. Furthermore, we put $n_A(x) := n_A(\mathrm{lexNF}(x))$ for $x \in M$.

Lemma 9.3.5 *Let $(M, \cdot, 1, \rho)$ be a labeled divisibility monoid, $x, y \in M$ and $x\mathcal{I}y$. Then $n_{\rho(x)}(x \vee y) \leq |E| + 1$.*

Proof. If $n_{\rho(x)}(x\vee y) = 1$, the statement is trivial. So let $n_{\rho(x)}(x\vee y) \geq 2$. Since $n_A(x \vee y) = n_{E\setminus A}(x \vee y)$ for any $A \subseteq E$, we may assume that the label sequence of $\mathrm{lexNF}(x \vee y)$ starts with a letter from $A := \rho(x)$.

Hence there exist words $u, v \in T^+$ and $u' \in T^\star$ with $\rho(u) \subseteq A$, $\rho(v) \subseteq E \setminus A$ and $\mathrm{lexNF}(x \vee y) = uvu'$. Now let a be the first letter of u and b the first one of v.

First we show $\rho(\mathrm{nat}(ub) \wedge y) = \rho(b)$: Let $h := \mathrm{nat}(ub) \wedge y$. Then there exist uniquely determined $k, l \in M$ with $\mathrm{nat}(ub) = h\cdot k$ and $y = h\cdot l$. By Lemma 7.1.1, $k\,\mathcal{I}\,l$. Hence $\rho(k) \cap \rho(l) = \emptyset$ by Lemma 9.3.1. We write $\#_e h$ for the number of occurrences of the letter $e \in E$ in the label sequence of any representative of $h \in M$, which is well-defined by Lemma 8.2.1 and the requirements on ρ. So we get on the other hand, $\#_e\,\mathrm{nat}(ub) = \#_e h + \#_e k$ and $\#_e y = \#_e h + \#_e l$ for any $e \in E$. Hence $\#_e h = \min(\#_e\,\mathrm{nat}(ub), \#_e y)$. Note that $\rho(b) \in \rho(y)$. Hence $\rho(b) \in \rho(h)$. Now let $e \in \rho(y) \setminus \rho(b)$. Then $e \notin \rho(x)$ and therefore $\#_e h = 0$. Thus $\rho(\mathrm{nat}(ub) \wedge y) = \rho(b)$.

Hence there exists a word $w \in T^\star$ with $x \vee y = \mathrm{nat}(w)$ such that the label sequence of w starts with $\rho(b)$. Since the label sequence of the lexicographical normal form of $x \vee y$ starts with $\rho(a)$, we get $\rho(a) \prec \rho(b)$.

So let $\mathrm{lexNF}(x \vee y) = u_0 v_0 u_1 v_1 \dots u_n v_n$ with $u_i \neq \varepsilon$ for all $i \leq n$, $v_i \neq \varepsilon$ for $i < n$, $\rho(u_i) \subseteq A$ and $\rho(v_i) \subseteq E \setminus A$. Let a_i (b_i) be the first letter of u_i for $i \leq n$ (v_i for $i < n$, resp.). Using Lemma 7.1.1, we can apply the above result inductively and obtain $\rho(a_i) \prec \rho(b_i) \prec \rho(a_{i+1})$ for each $i < n$. Hence $2n + 1 \leq |E|$. ☐

Corollary 9.3.6 *Let $X \subseteq T^\star$ be a monoalphabetic language. Then* $\mathrm{nat}(w)$ *is connected for any word $w \in X^{|E|+2} \cap \mathrm{LNF}$.*

Proof. Let $n = |E|+1$. Then there exist $x_i \in \mathrm{nat}(X)$ with $\mathrm{nat}(w) = x_0 x_1 \dots x_n$. Now let $x, y \in M$ with $x \mathrel{\mathcal{I}} y$ and $x \vee y = \mathrm{nat}(w)$. Then $\rho(x) \cap \rho(y) = \emptyset$ by Lemma 9.3.1. If $\rho(x_i) \cap \rho(x) \neq \emptyset$ and $\rho(x_i) \cap (E \setminus \rho(x)) \neq \emptyset$ for all $0 \leq i \leq n$, we would obtain $n_{\rho(x)}(\mathrm{nat}(w)) > n = |E| + 1$, contradicting Lemma 9.3.5. Hence there exists $i \in \{0, 1, \dots, n\}$ such that $\rho(x_i) \subseteq \rho(x)$ or $\rho(x_i) \subseteq E \setminus \rho(x)$.

First consider the case $\rho(x_i) \subseteq \rho(x)$. Since X is monoalphabetic, this implies $\rho(x_j) = \rho(x_i) \subseteq \rho(x)$ for all $0 \leq j \leq n$. Now $\rho(y) = \emptyset$ follows from the inclusions $\rho(y) \subseteq \rho(w) \subseteq \rho(x)$ and from $\rho(x) \cap \rho(y) = \emptyset$. Hence $y = 1$.

Now consider the case $\rho(x_i) \subseteq E \setminus \rho(x)$. From Lemma 9.3.1, we obtain $\rho(x) \cup \rho(y) = \rho(\mathrm{nat}(w)) \supseteq \rho(x_i)$ and this implies $\rho(x_i) \subseteq \rho(y)$. Now we can argue as above (with x and y interchanged) and obtain $x = 1$. ☐

Corollary 9.3.7 *Let $(M, \cdot, 1, \rho)$ be a labeled divisibility monoid. Let $L \subseteq M$ be recognizable. Then L is mc-rational.*

Proof. Let $X := \{u \in T^\star \mid \mathrm{nat}(u) \in L\} \cap \mathrm{LNF}$. Since any $x \in M$ has a unique lexicographical normal form, we have $\mathrm{nat}(X) = L$. Then X is recognizable in $T^\star$ and therefore rational. By Lemma 7.1.2, it can be constructed from finite languages in $T^\star$ by the operation $\cdot$, $\cup$ and $\langle . \rangle$ applied to monoalphabetic languages, only. Since $X \subseteq \mathrm{LNF}$, any intermediate language in the construction of X is contained in LNF, too. Let Y be such an intermediate language and suppose that the iteration $\langle . \rangle$ is applied to Y. Hence Y is monoalphabetic. Then $\langle Y \rangle$ is another intermediate language and therefore contained in LNF. Hence by Corollary 9.3.6, $\mathrm{nat}(Y)^{|E|+2}$ is connected. Note that $\langle Y \rangle = \left(\bigcup_{0 \leq i \leq |E|+1} Y^i\right) \left\langle Y^{|E|+2} \right\rangle$. Therefore, we can construct $\mathrm{nat}(X) = L$ as required. ☐

We can summarize our results on recognizable, c-rational and mc-rational languages as follows.

Theorem 9.3.8 *Let $(M, \cdot, 1, \rho)$ be a labeled divisibility monoid with finite commutation behavior. Let $L \subseteq M$. Then the following are equivalent:*

1. *L is recognizable*
2. *L is c-rational*
3. *L is mc-rational.*

Proof. The implications $2 \to 1 \to 3 \to 2$ are Theorem 9.2.8, Corollary 9.3.7 and 9.3.2, respectively. $\square$

Chapter 10

Kleene's Theorem

Theorem 9.3.8 characterizes the recognizable languages in a divisibility monoid with finite commutation behavior using the concept of c-rationality which is a stronger notion than rationality. The aim of this section is to characterize those divisibility monoids that satisfy Kleene's Theorem: A divisibility monoid $(M, \cdot, 1)$ is *width-bounded* provided there exists $n \in \mathbb{N}$ with $w(\downarrow x, \leq) \leq n$ for any $x \in M$. Thus, a divisibility monoid is width-bounded if there is a uniform bound for the width of the lattices $\downarrow x$. Hence in the partial order $(M, \leq)$, bounded antichains have a uniformly bounded size. Note that a free monoid is width-bounded with $n = 1$ and that a direct product of two free monoids is not width-bounded. Hence a trace monoid is width-bounded iff it is free.

10.1 Rational monoids

Rational monoids are the main tool in our proof that any width-bounded divisibility monoid satisfies Kleene's Theorem. This concept was introduced by Sakarovitch [Sak87]. He showed that rational monoids satisfy Kleene's Theorem and considered closure properties of this class of monoids (cf. also [PS90] where the latter topic was extended). In this section, we recall some definitions and results from [Sak87] and prove a first statement concerning divisibility monoids.

Let $(M, \cdot, 1)$ be a monoid. A *generating system* of M is a pair (X, α) where X is a set and $\alpha : X^\star \to M$ is a surjective homomorphism. Then the *kernel of* α, i.e. the binary relation $\ker \alpha = \{(v, w) \in X^\star \times X^\star \mid \alpha(x) = \alpha(y)\}$, is a congruence relation on the free monoid $X^\star$.

An idempotent function $\beta : X^\star \to X^\star$ with $\ker \beta = \ker \alpha$ is a *description of* (X, α). We can think of $\beta(v)$ as a normal form of the word v. Note that (X, α) might have several descriptions. But for any such description β, $M \cong T^\star / \ker \beta$ since $\ker \beta = \ker \alpha$.

Let $(M, \cdot, 1)$ be a divisibility monoid. In Section 7.3, we defined the set $\mathcal{C}\ell$ to consist of all nonempty subsets of T of pairwise independent elements that are bounded in $(M, \leq)$. Furthermore, α was defined to be the extension of the function $A \mapsto \sup(A)$ to a homomorphism from $\mathcal{C}\ell^\star$ onto M. Hence the tuple $(\mathcal{C}\ell, \alpha)$ is a generating system of the divisibility monoid $(M, \cdot, 1)$. Furthermore, we constructed an automaton $\mathcal{A}$ on the monoid $T^\star \times \mathcal{C}\ell^\star$ that computes the function $\mathrm{fnf} \circ \mathrm{nat} : T^\star \to \mathcal{C}\ell^\star$ by Theorem 7.3.6. The following proof uses this function to show that $\beta := \mathrm{fnf} \circ \alpha$ is a description of the generating system $(\mathcal{C}\ell, \alpha)$:

Lemma 10.1.1 *Let $(M, \cdot, 1)$ be a divisibility monoid. Then $\mathrm{fnf} \circ \alpha : \mathcal{C}\ell^\star \to \mathcal{C}\ell^\star$ is a description of $(\mathcal{C}\ell, \alpha)$.*

Proof. For any $A \in \mathcal{C}\ell$, choose some word $w_A \in T^\star$ with $\mathrm{nat}(w_A) = \alpha(A)$. Then there exists a homomorphism $\psi : \mathcal{C}\ell^\star \to T^\star$ that extends the mapping $A \mapsto w_A$. In addition, $\mathrm{nat} \circ \psi : \mathcal{C}\ell^\star \to M$ satisfies

$$\begin{aligned}
\mathrm{nat} \circ \psi(A_1 A_2 \dots A_n) &= \mathrm{nat}(w_{A_1} w_{A_2} \dots w_{A_n}) \\
&= \mathrm{nat}(w_{A_1}) \cdot \mathrm{nat}(w_{A_2}) \cdots \mathrm{nat}(w_{A_n}) \\
&= \alpha(A_1) \cdot \alpha(A_2) \cdots \alpha(A_n) \\
&= \alpha(A_1 A_2 \dots A_n),
\end{aligned}$$

i.e. $\alpha = \mathrm{nat} \circ \psi$. Hence $\beta = \mathrm{fnf} \circ \mathrm{nat} \circ \psi$.

It remains to show that β is idempotent and that $\ker \beta = \ker \alpha$: Since $\mathrm{fnf}(x)$ is the unique word in FNF with $\alpha(\mathrm{fnf}(x)) = x$, we have $\alpha \circ \mathrm{fnf} = \mathrm{id}_M$. Hence $\beta \circ \beta = \mathrm{fnf} \circ \alpha \circ \mathrm{fnf} \circ \alpha = \mathrm{fnf} \circ \mathrm{id}_M \circ \alpha = \beta$, i.e. β is idempotent. Now let $v, w \in \mathcal{C}\ell^\star$ with $\alpha(v) = \alpha(w)$. Then, clearly, $\beta(v) = \mathrm{fnf} \circ \alpha(v) = \mathrm{fnf} \circ \alpha(w) = \beta(w)$, i.e. $\ker(\alpha) \subseteq \ker(\beta)$. Conversely, $\beta(v) = \beta(w)$ implies $\alpha \circ \mathrm{fnf} \circ \alpha(v) = \alpha \circ \mathrm{fnf} \circ \alpha(w)$ and therefore $\ker \beta \subseteq \ker \alpha$ by $\alpha \circ \mathrm{fnf} = \mathrm{id}_M$. □

A function $\beta : M \to N$ mapping one monoid into another can be seen as a subset of $M \times N$. Since this direct product is a monoid, we can speak of rational sets in $M \times N$. In this spirit, a function $\beta : M \to N$ is a *rational function* if it is a rational set in $M \times N$.

A monoid $(M, \cdot, 1)$ is a *rational monoid* if there exists a generating system (X, α) of M that has a rational description. Loosely speaking, a monoid is rational if there is a rational normal form function β that determines M. Let $\beta : X \to X$ be a rational description of the rational monoid M. Since the image of a rational set under a rational function is a rational set, the set $\beta(X^\star)$ is rational in the free monoid $X^\star$. Hence $M = \alpha \circ \beta(X^\star)$ is rational in M. Since any rational set in M is contained in a finitely generated submonoid of M, this implies that a rational monoid is finitely generated.

The key property of rational monoids that will be used in our considerations is that they satisfy Kleene's Theorem:

Theorem 10.1.2 ([Sak87, Theorem 4.1]) *Let M be a rational monoid and $L \subseteq M$. Then L is rational iff it is recognizable.*

Suppose the trace monoid $\mathbb{M}(\Sigma, D)$ is rational. Then it satisfies Kleene's Theorem implying that it is free. Since, conversely, any free monoid is rational, a trace monoid is rational iff it is free.

10.2 Width-bounded divisibility monoids

10.2.1 Width-bounded divisibility monoids are rational

In this section, we will show that the description $\text{fnf} \circ \alpha$ of the generating system $(\mathcal{Cl}, \alpha)$ for a width-bounded divisibility monoid is a rational function. To this purpose, we first show that the function $\text{fnf} \circ \text{nat}$ is rational. This is based on the following theorem that characterizes rational subsets in a monoid.

Theorem 10.2.1 ([EM65]) *Let M be a monoid. A set $L \subseteq M$ is rational iff it is the behavior of a finite automaton over M.*

Recall that an automaton is finite whenever its set of transitions is finite. Since the transitions of the automaton $\mathcal{A}$ from Theorem 7.3.6 are elements of the set $Q \times (T \times \mathcal{Cl}_\varepsilon) \times Q$, and since the set $T \times \mathcal{Cl}_\varepsilon$ is finite, it suffices to show that there are only finitely many reachable states. To this purpose, we show that the length of the monoid elements in reachable states is bounded. But first, we need the following lemma on the lattices $\downarrow x$ for $x \in M$. As known from traces, the width of these lattices is in general unbounded. Here we show that nevertheless the width of the join-irreducible elements is bounded by T:

Lemma 10.2.2 *Let $(M, \cdot, 1)$ be a divisibility monoid and $x \in M$. Then the width of $(\mathbb{J}(x), \leq)$ is at most $|T|$.*

Proof. Let $A \subseteq \mathbb{J}(x)$ be an antichain. Define

$$b := \sup\{y \in \mathbb{J}(x) \mid \neg\exists a \in A : a \leq y\}.$$

Since $\downarrow b \cap \mathbb{J}(x)$ equals $\{y \in \mathbb{J}(x) \mid \neg\exists a \in A : a \leq y\}$ and since A is an antichain, it is the set of minimal elements of the partially ordered set $\mathbb{J}(x) \setminus \downarrow b$. By Lemma 7.3.1, $|A|$ equals the number of minimal elements of $\mathbb{J}([b, x])$. Since $[b, x]$ and $\downarrow b^{-1}x$ are order isomorphic by Lemma 7.1.1, $|A|$ is the number of minimal elements of $\mathbb{J}(b^{-1}x)$, i.e. of elements $t \in T$ with $t \leq b^{-1}x$. Hence $|A| \leq |T|$. ☐

Now we can bound the number of reachable states in the automaton $\mathcal{A}$.

Lemma 10.2.3 *Let $(M, \cdot, 1)$ be a width-bounded divisibility monoid such that $w(\downarrow x, \leq) \leq n$ for any $x \in M$. Let $x, y \in M$ with $|\mathrm{fnf}(xy)| = |\mathrm{fnf}(x)|$. Then $|y| < 2(n+1)|T|$.*

Proof. By contradiction, assume $|y| \geq 2(n+1)|T|$. Since $x \leq xy$, the set $\mathbb{J}(x)$ is an ideal in $(\mathbb{J}(xy), \leq)$. Hence, for $v \in \mathbb{J}(x)$ and $w \in \mathbb{J}(xy)$, it holds $w \not\leq v$. The size of $\mathbb{J}(x)$ equals the length $(\downarrow x, \leq)$ and therefore of x and similarly for xy. Hence $\mathbb{J}(xy) \setminus \mathbb{J}(x)$ contains at least $2(n+1)|T|$ elements. By Lemma 10.2.2, $\mathbb{J}(xy) \setminus \mathbb{J}(x)$ has width at most $|T|$. Hence the elements of $\mathbb{J}(xy) \setminus \mathbb{J}(x)$ occupy at least $2(n+1)$ different heights, i.e. there are natural numbers $0 \leq n_1 < n_2 \cdots < n_{2(n+1)}$ such that there exists $w_i \in \mathbb{J}(xy) \setminus \mathbb{J}(x)$ with $h(w_i, \mathbb{J}(xy)) = n_i$ for $1 \leq i \leq 2(n+1)$. Since $|\mathrm{fnf}(xy)| = |\mathrm{fnf}(x)|$, the partially ordered sets $\mathbb{J}(xy)$ and $\mathbb{J}(x)$ have the same length by Lemma 7.3.2. Hence, for $1 \leq i \leq 2(n+1)$ there exists $v_i \in \mathbb{J}(x)$ with $h(v_i, \mathbb{J}(xy)) = n_i$. Since $h(w_i, \mathbb{J}(xy)) \leq h(v_j, \mathbb{J}(xy))$ for $1 \leq i \leq n \leq j \leq 2(n+1)$, the elements from $\{w_i \mid 1 \leq i \leq n+1\}$ and $\{v_j \mid n+1 \leq j \leq 2(n+1)\}$ are mutually incomparable. Then $I(i,j) := \downarrow\{w_1, w_2, \ldots, w_i, v_{n+1}, v_{n+2}, \ldots, v_{n+1+j}\}$ is a finitely generated ideal in $(\mathbb{J}(xy), \leq)$. Note that $w_j \not\leq w_i$ for $1 \leq i < j \leq n+1$ and similarly $v_j \not\leq v_i$ for $n+1 \leq i < j \leq 2(n+1)$. Hence the ideals $I(i, n-i)$ for $1 \leq i \leq n+1$ are pairwise incomparable, i.e. $(\mathbb{H}(\mathbb{J}(xy), \leq), \subseteq)$ contains an antichain of $n+1$ elements. Since $(\downarrow xy, \leq) \cong (\mathbb{H}(\mathbb{J}(xy), \leq), \subseteq)$, this contradicts our assumption. $\square$

The proof of the following theorem is based on the fact that the description $\mathrm{fnf} \circ \alpha$ of the generating system $(\mathcal{C}\ell, \alpha)$ for a width-bounded divisibility monoid is rational:

Theorem 10.2.4 *Any width-bounded divisibility monoid is a rational monoid.*

Proof. Let M be a width-bounded divisibility monoid. By Theorem 7.3.6, the automaton $\mathcal{A}$ computes the function $\mathrm{fnf} \circ \mathrm{nat} : T^\star \to \mathcal{C}\ell^\star$. To show that this is rational, it remains to prove that the number of reachable states in $\mathcal{A}$ is finite (since the transitions are labeled by the finite set $T \times \mathcal{C}\ell_\varepsilon$). Let (z, C) be a reachable state of $\mathcal{A}$. Then, by Lemma 7.3.5, there exists $x \in M$ with $|\mathrm{fnf}(x)| = |\mathrm{fnf}(xz)|$. Hence, by Lemma 10.2.3, the length of z is bounded by $2(n+1)|T|$ where n is the global bound for the size of bounded antichains in $(M, \leq)$. Since $\mathcal{C}\ell$ is finite, this implies that there are only finitely many reachable states in $\mathcal{A}$.

Recall that $(\mathcal{C}\ell, \alpha)$ is a generating system of M. By Lemma 10.1.1, the function $\mathrm{fnf} \circ \alpha : \mathcal{C}\ell^\star \to \mathcal{C}\ell^\star$ is a description of $(\mathcal{C}\ell, \alpha)$. To show that this description is rational, consider the homomorphism $\psi : \mathcal{C}\ell^\star \to T^\star$ defined in the proof of Lemma 10.1.1, where we also showed $\alpha = \mathrm{nat} \circ \psi$ and therefore $\beta = \mathrm{fnf} \circ \mathrm{nat} \circ \psi$. Since ψ is a homomorphism, it is a rational relation from $\mathcal{C}\ell^\star$ into $T^\star$, i.e. β splits into two rational relations $\mathcal{C}\ell^\star \to T^\star$ and $T^\star \to \mathcal{C}\ell^\star$. Since $T^\star$ is a free monoid, by [EM65] (cf. [Sak87, Proposition A.16]), β is rational. $\square$

Remark. By [Sak87, Theorem 4.1], Kleene's Theorem holds in any rational monoid. Thus, the theorem above implies that in a width-bounded divisibility monoid the rational and the recognizable sets coincide. There is an alternative proof of this weaker result that follows the line of the proof of Theorem 9.3.8: Let $(M, \cdot, 1)$ be a width-bounded divisibility monoid $(M, \cdot, 1)$ with $w(\downarrow x, \leq) \leq n$ for any $x \in M$. Then one shows that its monoid of commutation behaviors $\mathbb{C}_M$ has at most $|T|^{n+1} - 1 + |T|^{(n+1)(|T|^{n+1}-1)}$ elements. Hence any such monoid has finite commutation behavior. The crucial point then is to show that the rank of X is bounded by $2n$ for any $X \subseteq T^\star$.

10.2.2 Rational divisibility monoids are width-bounded

Our next goal is to show that the width-boundedness is not only sufficient but also necessary for Kleene's Theorem to hold. We start with two lattice-theoretic lemmata.

Lemma 10.2.5 *Let $(P, \leq)$ be a partially ordered set, $M, N \subseteq P$ sets with $n-1$ elements each such that any $m \in M$ is incomparable with any $n \in N$. Then there exists a semilattice embedding of $[n-1] \times [n-1]$ into $(\mathbb{H}_f(P), \subseteq)$. If $(P, \leq)$ is finite, this embedding can be chosen to preserve infima, too.*

Proof. Let $M = \{m_1, m_2 \ldots, m_{n-1}\}$ and $N = \{n_1, n_2 \ldots, n_{n-1}\}$ be linear extensions of $(M, \leq)$ and $(N, \leq)$, i.e. $m_i \leq m_j$ or $n_i \leq n_j$ implies $i \leq j$. Then $I(i, j) := \downarrow\{m_1, \ldots, m_i, n_1, \ldots, n_j\}$ is a finitely generated ideal and therefore an element of $\mathbb{H}_f(P, \leq)$. Furthermore, $(\{I(i, j) \mid 1 \leq i \leq n-1, 1 \leq j \leq n-1\}, \subseteq)$ is the desired subposet of $\mathbb{H}_f(P, \leq)$. $\square$

Next, we want to prove that any distributive lattice of sufficient width contains a large grid. Recall that $R_{n+1}(6^n)$ is a Ramsey number (cf. Ramsey's Theorem on page 124).

Theorem 10.2.6 *Let $(L, \leq)$ be a finite distributive lattice with $w(L) \geq R_{n+1}(6^n)$. Then there exists a lattice embedding of $[n-1] \times [n-1]$ into $(L, \leq)$.*

Proof. First we consider the case $w(\mathbb{J}(L)) \geq 2n$. Then there exists an antichain $A \subseteq \mathbb{J}(L)$ containing $2n$ elements. Let M and N be disjoint subsets of A of size n. Then Lemma 10.2.5 implies the statement.

Now assume $w(\mathbb{J}(L)) = k < 2n$. By Dilworth's Theorem [Dil50], there are chains $C_1, C_2, \ldots, C_k \subseteq \mathbb{J}(L)$ with $\mathbb{J}(L) = \bigcup_{i=1,\ldots k} C_i$. For $x \in L$ and $1 \leq \ell \leq k$ let $\partial_\ell(x)$ denote the maximal element of C_ℓ below x if it exists, and $\bot$ otherwise. To ease the notations in this proof, we will consider $\bot$ as an additional element of $\mathbb{J}(L)$ which is minimal. Since L is distributive, $x = \bigvee_{1\leq i\leq k} \partial_i(x)$ for any $x \in L$.

Since $w(L) \geq R_{n+1}(6^n)$, there is an antichain $A = \{x_1, x_2, \ldots, x_m\}$ in L with $m \geq R_{n+1}(6^n)$. Now we define a mapping $g_{i,j} : \{1, 2, \ldots, k\} \to \{<, =, >\}$ for $1 \leq i < j \leq m$ by $g_{i,j}(\ell) = \theta$ iff $\partial_\ell(x_i)\theta\partial_\ell(x_j)$ (with $\theta \in \{<, =, >\}$). For $x_i \neq x_j$, we define $g(x_i, x_j) = g_{\min\{i,j\},\max\{i,j\}}$. Thus, g maps the two-elements subsets of A into $\{<, =, >\}^{\{1,2,\ldots,k\}}$. Since this set contains at most $3^{2n} = 6^n$ elements and since $m > R_{n+1}(6^n)$, we can assume $g(x_i, x_j) = g(x_{i'}, x_{j'}) =: f$ for $i, j, i', j' \in \{1, 2, \ldots, n+1\}$ with $i \neq j$ and $i' \neq j'$. Then $f(\ell_1) \neq$"$=$" for some $1 \leq \ell_1 \leq k$ since otherwise $x_1 = x_2$. Similarly, there is an index $1 \leq \ell_2 \leq k$ with $f(\ell_2) \notin \{=, f(\ell_1)\}$ since otherwise x_1 and x_2 are comparable. Without loss of generality, we assume $f(1) =$"$<$" and $f(2) =$"$>$". Then $\partial_1(x_1) < \partial_1(x_2) < \ldots \partial_1(x_{n+1})$ and $\partial_2(x_1) > \partial_2(x_2) > \ldots \partial_2(x_{n+1})$. Thus, $C_j := \{\partial_j(x_i) \mid 1 < i < n+1\}$ for $j = 1, 2$ is a chain in $\mathbb{J}(L)$ containing $n-1$ elements.

Let $1 < i < n+1$ with $\partial_1(x_i) \geq \partial_2(x_i)$. Then $x_{i+1} \geq \partial_1(x_{i+1}) > \partial_1(x_i) \geq \partial_2(x_i) > \partial_2(x_{i+1})$ and $\partial_2(x_i), \partial_2(x_{i+1}) \in C_2$. But this contradicts the definition of $\partial_2(x_{i+1})$ as the maximal element of C_2 below x_{i+1}. Symmetrically, we can argue if $\partial_1(x_i) \leq \partial_2(x_i)$ (with x_{i-1} in place of x_{i+1}). Thus, $\partial_1(x_i)$ and $\partial_2(x_i)$ are incomparable for $1 < i < n+1$.

Now let $1 < i < j < n+1$ with $\partial_1(x_j) \geq \partial_2(x_i)$. Then $\partial_2(x_i) > \partial_2(x_j)$ since $i < j$, i.e. $\partial_1(x_j) > \partial_2(x_j)$, a contradiction to what we showed above. Similarly, we can argue in the cases $\partial_1(x_j) \leq \partial_2(x_i)$, $\partial_1(x_i) \geq \partial_2(x_j)$ and $\partial_1(x_i) \leq \partial_2(x_j)$.

Thus, we found two chains C_1 and C_2 in $\mathbb{J}(L)$ of size $n-1$ whose elements are mutually incomparable. Now Lemma 10.2.5 implies that $[n-1] \times [n-1]$ can be lattice embedded into $(\mathbb{H}(\mathbb{J}(L)), \subseteq) \cong L$. □

The following lemma implies that the free commutative monoid with two generators can be embedded into a divisibility monoid if the size of bounded antichains is unbounded.

Lemma 10.2.7 *Let $(M, \cdot, 1)$ be a divisibility monoid with finite commutation behavior. Let $z \in M$ with $w(\downarrow z) \geq R_{n+2}(6^{n+1})$ where $n = R_3(|\mathbb{C}_M|) + 1$. Then there exist $x, y \in M \setminus \{1\}$ such that $x \mathcal{I} y$, $c_y(x) = x$ and $c_x(y) = y$.*

Proof. By Theorem 10.2.6, there is a lattice embedding $\eta : [n-1]^2 \to \downarrow z$. By cancellation, we may assume $\eta(0, 0) = 1$. For $0 \leq i \leq n-2$ there is $x_i \in M \setminus \{1\}$ with $\eta(i, 0) \cdot x_i = \eta(i+1, 0)$. By Lemma 9.2.4, there are $0 \leq i < j \leq n-1$ with $c_{x_i x_{i+1} \cdots x_{j-1}} \subseteq \mathrm{id}_M$. Furthermore, there are $y_\ell \in M \setminus \{1\}$ with $\eta(i, \ell) \cdot y_\ell = \eta(i, \ell+1)$. Using Lemma 9.2.4 again, there are $0 \leq k < \ell \leq n-1$ with $c_{y_k y_{k+1} \cdots y_{\ell-1}} \subseteq \mathrm{id}_M$.

Let $x := \eta(i,k)^{-1}\eta(j,k)$ and $y := \eta(i,k)^{-1}\eta(i,\ell) = y_k y_{k+1} \cdots y_{\ell-1}$. Then $c_y \subseteq \mathrm{id}_M$.

To show $c_x \subseteq \mathrm{id}_M$, note that $\eta(i,0) = \eta(i,k) \wedge \eta(j,0)$ since $i < j$. Hence, by Lemma 7.1.1, $1 = \eta(i,0)^{-1}\eta(i,0) = \eta(i,0)^{-1}\eta(i,k) \wedge \eta(i,0)^{-1}\eta(j,0)$, i.e. $\eta(i,0)^{-1}\eta(i,k)$ and $\eta(i,0)^{-1}\eta(j,0)$ are independent.

Similarly, we get $\eta(i,k) \vee \eta(j,0) = \eta(j,k)$ since $i < j$ and therefore

$$\begin{aligned} \eta(i,0)^{-1}\eta(i,k) \vee \eta(i,0)^{-1}\eta(j,0) &= \eta(i,0)^{-1}\eta(j,k) \\ &= \eta(i,0)^{-1}\eta(i,k)\ \eta(i,k)^{-1}\eta(j,k). \end{aligned}$$

Thus $\eta(i,0)^{-1}\eta(j,0) \uparrow \eta(i,0)^{-1}\eta(i,k) = \eta(i,k)^{-1}\eta(j,k) = x$. Since the commutation behavior of $x_i x_{i+1} \cdots x_{j-1} = \eta(i,0)^{-1}\eta(j,0)$ is contained in the identity, Lemma 7.2.6 indeed implies $c_x \subseteq \mathrm{id}_M$.

It remains to show $x \mathrel{\mathcal{I}} y$: Since $\eta(j,k), \eta(i,\ell) \leq \eta(j,\ell)$, the elements x and y are bounded in $(M, \leq)$. Furthermore, $\eta(i,\ell) \wedge \eta(j,\ell) = \eta(i,k)$ implies $x \wedge y = 1$. □

Now we can characterize the divisibility monoids that satisfy Kleene's Theorem.

Theorem 10.2.8 *Let $(M, \cdot, 1)$ be a divisibility monoid with finite commutation behavior. Then the following are equivalent*

1. *M is width-bounded,*
2. *M is rational, and*
3. *any set $L \subseteq M$ is rational iff it is recognizable.*

Proof. The implication $1 \Rightarrow 2$ follows from Theorem 10.2.4, and the implication $2 \Rightarrow 3$ from [Sak87, Theorem 4.1]. Now assume M not to be width-bounded. Then, by Lemma 10.2.7, there are $x, y \in M \setminus \{1\}$ such that $x \mathrel{\mathcal{I}} y$, $c_x(y) = y$ and $c_y(x) = x$. Hence we can embed the monoid $(\mathbb{N} \times \mathbb{N}, +, (1,1))$ into M (extending the mapping $(1,0) \mapsto x$ and $(0,1) \mapsto y$ to a homomorphism). Since $\{(i,i) \mid i \in \mathbb{N}\}$ is rational but not recognizable in $(\mathbb{N} \times \mathbb{N}, +, (0,0))$, its image is so in M. Hence M does not satisfy Kleene's Theorem, i.e. the implication $3 \Rightarrow 1$ is shown. □

Remark. Note that the assumption on M to have finite commutation behavior is necessary for the implication $3 \Rightarrow 1$, only. On the other hand, the implications $1 \Rightarrow 2 \Rightarrow 3$ can be shown without this assumption. It is not clear whether the other implications, in particular that any rational divisibility monoid is width-bounded can be shown without this assumption.

Chapter 11

Monadic second order logic

11.1 Two Büchi-type theorems

Büchi showed that the monadic second order theory of the linearly ordered set $(\omega, \leq)$ is decidable. To achieve this goal, he used automata. In the course of these considerations it was shown that a language in a free finitely generated monoid is recognizable iff it is monadically axiomatizable. In computer science, this latter result and its extension to infinite words are often referred to as "Büchi's Theorem" while in logic it denotes the decidability of the monadic theory of ω. Here, I understand it in this second meaning, i.e. it is the aim of this section to show that certain monadic theories associated to a divisibility monoid are decidable. In particular, it will be shown that the monadic theory $\mathrm{MTh}(\{(\mathbb{J}(\downarrow m), \leq) \mid m \in M\})$ is decidable for any divisibility monoid with finite commutation behavior.

Let $(L, \leq)$ be a finite distributive lattice. Let $x, y \in L$ with $x \longrightarrow\!\!\!< y$. Then there exists a uniquely determined join-irreducible element $z \in \mathbb{J}(L)$ such that $z \leq y$ and z is incomparable with x. We denote this element by $\mathrm{prim}(x, y)$. Then $x \vee \mathrm{prim}(x, y) = y$.

Lemma 11.1.1 *Let $(M, \cdot, 1)$ be a divisibility monoid. Furthermore, let $s, t \in T$ and $u, v \in M$. Then $\mathrm{prim}(u, us)$ and $\mathrm{prim}(usv, usvt)$ are incomparable iff there exist $x_1, x_2 \in M$ and $s' \in T$ such that*

$$sv = x_1 s' x_2, s' = c_{x_1}(s) \text{ and } t \in \mathrm{im}(c_{s'x_2}).$$

The situation of the lemma is depicted by Figure 11.1.

Proof. First, assume $\mathrm{prim}(u, us)$ and $\mathrm{prim}(usv, usvt) =: b$ to be incomparable. Since b is join-irreducible, there is a uniquely determined element $a \in M$ with $a \longrightarrow\!\!\!< b$. Assume $us \leq uva$. Then $\mathrm{prim}(u, us) \leq u \vee \mathrm{prim}(u, us) = us \leq a < b$, contradicting $\mathrm{prim}(u, us) \parallel b$. Hence $us \not\leq u \vee a$. Furthermore, $u \vee a < us \vee a$ for otherwise $us \leq us \vee a = u \vee a$ contradicting to what we just showed. Hence

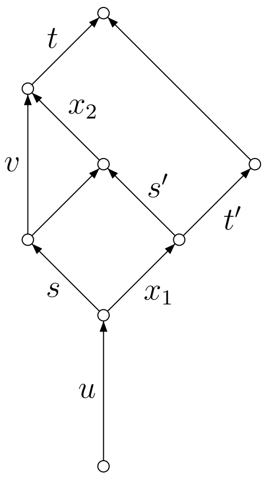


Figure 11.1: cf. Lemma 11.1.1

$u \vee a \lessdot us \vee a$, i.e. there exists $s' \in T$ with $(u \vee a) \cdot s' = us \vee a$. Let $x_1 := u^{-1}(u \vee a)$, i.e. $u \vee a = ux_1$. Then $ux_1 s' = \sup(u, us, a) = ux_1 \vee us$ implying $s' = x_1 \vee s$. On the other hand, $us \not\leq u \vee a = ax_1$, i.e. $s \not\leq x_1$ implies $s \wedge x_1 = 1$ since $s \in T$. Hence $x_1 \, \mathcal{I} \, s$ and therefore $s' = s \uparrow x_1 = c_{x_1}(s)$.

Next we show $u \vee a \lessdot u \vee b$. Clearly, $a \leq b$ implies $u \vee a \leq u \vee b$. Assume them to be equal. Since b is join-irreducible and above a, this implies $b \leq u \leq usv$. But this contradicts $b \vee usv = \mathrm{prim}(usv, usvt) \vee usv = usvt > usv$. Since $a \lessdot b$ we get $u \vee a \lessdot u \vee b$. Hence there exists $t' \in T$ with $u \vee a \cdot t' = u \vee b$.

Let $x_2 \in M$ be given by $(us \vee a) \cdot x_2 = usv$. It remains to show that we have $t = t' \uparrow s'x_2$: First note that $ux_1 t' = u \vee b$ and

$$ux_1 s' x_2 = (u \vee a) s' x_2 = (us \vee a) x_2 = usv.$$

Hence $ux_1 t' \vee ux_1 s' x_2 = u \vee b \vee usv = usvt = ux_1 s' x_2 t$. This implies $t' \vee s'x_2 = s'x_2 t$ and therefore in particular $s'x_2 \lessdot t' \vee s'x_2$. Since $t' \in T$, this implies $t' \, \mathcal{I} \, s'x_2$ and $t = t' \uparrow s'x_2$.

Conversely, let $s', t' \in T$ and $x_1, x_2 \in M$ such that we have $sv = x_1 s' x_2$, $s' = s \uparrow x_1$ and $t = t' \uparrow s'x_2$. Then $s \vee x_1 = x_1 s'$ implying $us \vee ux_1 = ux_1 s'$. Now $ux_1 \vee \mathrm{prim}(u, us) = ux_1 \vee u \vee \mathrm{prim}(u, us) = ux_1 \vee us = ux_1 s'$ follows.

Similarly, $t' \vee s'x_2 = s'x_2 t$ and therefore $ux_1 t' \vee ux_1 s' x_2 = ux_1 s' x_2 t$ or, since $ux_1 t' x_2 = usv$, $ux_1 t' \vee usv = usvt$. On the other hand, we have $usv \vee ux_1 t' = ux_1 s' x_2 \wedge ux_1 t' = ux_1(s'x_2 \wedge t') = ux_1$ since $s'x_2 \mathcal{I} t'$. Hence the two prime intervals $(ux_1, ux_1 t')$ and $(usv, usvt)$ are transposed. Thus we get $ux_1 \vee \mathrm{prim}(usv, usvt) = ux_1 \vee \mathrm{prim}(ux_1, ux_1 t') = ux_1 t'$.

Since t' and $s'x_2$ are independent, t' and s' are in particular incomparable. Hence so are ux_1s' and ux_1t'. Since, as we saw above, $ux_1 \vee \text{prim}(u, us) = ux_1s'$ and $ux_1 \vee \text{prim}(usv, usvt) = ux_1t'$, $\text{prim}(u, us)$ and $\text{prim}(usv, usvt)$ are incomparable. □

Lemma 11.1.2 *Let M be a divisibility monoid with finite commutation behavior $\mathbb{C}_M$. Then there exists a monadic formula* less *over the signature $\{\leq, \lambda\}$ with two free elementary variables such that for any $w \in T^\star$:*

$$(\mathbb{J}(\downarrow[w]), <) \cong (\text{dom}(w), \{(x, y) \in (\text{dom}(w))^2 \mid w \models \text{less}(x, y)\}).$$

Proof. For $c \in \mathbb{C}_M$ let L_c denote the set of all $x \in M$ with $c_x = c$. Then, for $s, t \in T$, we have

$$\begin{aligned} M_{s,t} &:= \{x_1s'x_2 \mid x_1, x_2 \in M, s' \in T \text{ such that } s' = c_{x_1}(s) \text{ and } t \in \text{im}(c_{s'x_2})\} \\ &= \bigcup_{s' \in T} (\bigcup\{L_c \mid c \in \mathbb{C}_M, c(s') = s\} \cdot s' \cdot \bigcup\{L_{c_x} \mid x \in M, t \in \text{im}(c_{s'x})\}). \end{aligned}$$

Since M has finite commutation behavior and $x \mapsto c_x$ is a monoid antihomomorphism, this set is recognizable by Corollary 9.2.2. Hence $\{w \in T^\star \mid [w] \in M_{s,t}\}$ is recognizable in $T^\star$ and therefore axiomatizable by a monadic sentence $\varphi_{s,t}$. Now we define

$$\text{less}(x, y) := \bigwedge_{s,t \in T} (\lambda(x) = s \wedge \lambda(y) = t \wedge x < y \wedge \neg\varphi'_{s,t})$$

where $\varphi'_{s,t}$ is the restriction of $\varphi_{s,t}$ to the positions between x and y, i.e. to the set $\{z \in \text{dom}(w) \mid x \leq z < y\}$. Now the lemma follows easily by the preceding lemma. □

Theorem 11.1.3 *Let $(M, \cdot, 1)$ be a divisibility monoid with finite commutation behavior $\mathbb{C}_M$. Then the monadic theory $\text{MTh}(\{(\mathbb{J}(\downarrow m), <) \mid m \in M\})$ is decidable.*

Proof. Let φ be a monadic sentence over the signature $\{<\}$. In φ, replace any subformula of the form $x < y$ by $\text{less}(x, y)$ and denote the resulting sentence by $\overline{\varphi}$. Then, for any $w \in T^\star$, we have $w \models \overline{\varphi}$ iff $(\mathbb{J}(\downarrow[w]), <) \models \varphi$. Since the monadic theory of the words over T is decidable, the result follows. □

By the theorem above, the monadic theory of $\{(\mathbb{J}(\downarrow m), \leq) \mid m \in M\}$ is decidable. The union of all these sets is $\mathbb{J}(M)$, the set of join-irreducibles in

$(M, \leq)$. The following theorem shows that the monadic theory of this set is not decidable in general:

Theorem 11.1.4 *Let (Σ, D) be a finite dependence alphabet. Then the monadic theory of $(\mathbb{J}(\mathbb{M}(\Sigma, D)), \leq)$ is decidable iff D is transitive.*

Proof. Let D be transitive. Then $\mathbb{J}(\mathbb{M}(\Sigma, D), \leq)$ is the disjoint union of trees of the form $(\{1, 2, \ldots, k\}^\star, \leq)$. Since the monadic theory of these uniformly branching trees is decidable [Rab69], so is the monadic theory of their disjoint union [She75].

On the other hand, suppose D not to be transitive. Then there are $a, b, c \in \Sigma$ with $(a, b), (b, c) \in D$ and $(a, c) \notin D$. We show how to encode an undirected graph (V, E) into two antichains A and B of $\mathbb{J}(\mathbb{M}(\Sigma, D))$: Suppose $V = \{1, 2, \ldots, n\}$. The vertices are represented by the elements of the set $A := \{a^k c^k b \mid 1 \leq k \leq n\}$. Furthermore, the edges of the graph (V, E) correspond to the elements of the antichain $B := \{a^i c^j b \mid (i, j) \in E\}$. Then, for any "vertices" $x, y \in A$, there is an edge in the graph (V, E) iff there exist $x', y' \in \mathbb{J}(\mathbb{M}(\Sigma, D))$ and $z \in B$ such that $x' \multimap x, z$ and $y' \multimap y, z$. Since this can be expressed by an elementary formula, we can easily reduce the elementary theory of graphs to the monadic antichain theory of $(\mathbb{M}(\Sigma, D), \leq)$. $\square$

Again, by Theorem 11.1.3, the monadic theory $\mathrm{MTh}\{\mathbb{J}(\downarrow x, \leq) \mid x \in M\}$ is decidable for any divisibility monoid with finite commutation behavior. This does not imply that the monadic theory $\mathrm{MTh}\{(\downarrow x, \leq) \mid x \in M\}$ is decidable. A counterexample is provided by the free commutative monoid with two generators since this monoid contains, for any $n \in \mathbb{N}$, an element x such that $(\downarrow x, \leq)$ is the grid $([n]^2, \leq)$. We will show that these grids are the only reason for the undecidability.

To this aim, we first show that for a given divisibility monoid $(M, \cdot, 1)$ with finite commutation behavior, the set of lattices $(\downarrow m, \leq)$ for $m \in M$ is finitely axiomatizable in monadic second order logic (Corollary 11.1.7).

Let Σ be a finite alphabet and consider the elementary logic that is appropriate to reason on Σ-labeled partially ordered sets. Furthermore, we deal with pomsets without autoconcurrency, only, i.e. we consider structures $t = (V, \leq, \lambda)$ where $(V, \leq)$ is a finite partially ordered set and $\lambda : V \to \Sigma$ is a mapping such that $\lambda^{-1}(a)$ is linearly ordered in $(V, \leq)$. In this setting, one can write down a formula φ with two free variables x and y such that

$$\varphi^t = \{(x, y) \in E^2 \mid t \models \varphi(x, y)\}$$

is a linear extension of $\leq$. For traces over (Σ, D) this was shown in [EM96]. For Σ-labeled partially ordered sets that are associated to the computations of stably concurrent automata, it has been observed independently in [DK96]. The most compact formula that defines a linear order in pomsets without autoconcurrency can be found in [DM97]. They consider traces only. Nonetheless, following their argumentation verbatim, one can easily see that their formula defines a linear order extension of the partial order of any pomset without autoconcurrency. Knowing this, the following lemma is an immediate reformulation:

Lemma 11.1.5 *There exists a monadic formula* $\mathrm{lin}(x, y, C_1, \ldots, C_m)$ *satisfying: For any finite partial order* $(P, \leq)$ *of width at most* n *and any chains* $C_i \subseteq P$ *for* $1 \leq i \leq m$ *such that* $P = \bigcup_{1\leq i\leq m} C_i$ *and* $C_i \cap C_j = \emptyset$ *for* $1 \leq i < j \leq m$*, the relation*

$$\mathrm{lin}^{(P,\leq,C_1,\ldots,C_m)} = \{(x, y) \in P^2 \mid (P, \leq) \models \mathrm{lin}(x, y, C_1, \ldots, C_m)\}$$

is a linear order extending $\leq$*.*

Theorem 11.1.6 *Let* $(M, \cdot, 1)$ *be a divisibility monoid with finite commutation behavior. There exists a monadic sentence* $\overline{\varphi}$ *such that for any finite partial order* $(P, \leq)$*:*

$$(P, \leq) \models \overline{\varphi} \iff \text{there exists } x \in M \text{ with } \mathbb{J}(\downarrow x, \leq) \cong (P, \leq).$$

Proof. Let $m = |T|$ denote the number of irreducible elements of the monoid $(M, \cdot, 1)$. By Lemma 11.1.2, there exists a formula less that defines the partially ordered set $\mathbb{J}(\downarrow[w], <)$ inside the word $w \in T^\star$. In this formula, replace any subformula of the form $\lambda(z) = t$ by $z \in M_t$ and any subformula of the form $z \leq z'$ by $\mathrm{lin}(z, z', C_1, C_2, \ldots, C_m)$. The result is denoted by less$'$. Now let $\varphi(C_1, C_2, \ldots, C_m)$ denote the following formula

$$\begin{aligned} \exists_{t\in T} M_t \quad (\quad & \bigcup_{t\in T} M_t = \text{ everything} \wedge \\ & M_s \cap M_t = \emptyset \text{ for } s \neq t \wedge \\ & \forall x, y (x < y \leftrightarrow \mathrm{less}'(x, y, C_1, \ldots, C_m)) \\ ). \quad & \end{aligned}$$

Let $x \in M$ and $(C_i)_{1\leq i\leq m}$ be a tuple of mutually disjoint chains whose union equals $\mathbb{J}(\downarrow x, \leq)$. For simplicity, let $(P, \leq) := \mathbb{J}(\downarrow x, \leq)$. Then $\mathrm{lin}^{(P,\leq,C_1,\ldots,C_m)}$ defines a linear order that extends $\leq$. This linear order defines a maximal chain in the lattice $(\downarrow x, \leq)$ which corresponds naturally to a word $w \in T^\star$ with $\mathrm{nat}(w) = x$. Now let M_t be the set of positions in w that are labeled by the irreducible element $t \in T$. Then the sets M_t satisfy the first two conditions of the formula φ. Furthermore, by Lemma 11.1.2, the last statement holds as well. Hence $(P, \leq) \models \varphi(C_1, \ldots, C_m)$. On the contrary, let $(P, \leq)$ be a finite partial order, C_i mutually disjoint chains whose union is P such that $(P, \leq) \models \varphi(C_1, \ldots, C_m)$. Let $P = (x_1, x_2, \ldots, x_k)$ be the enumeration of P that is completely defined by

$(P, \leq) \models \mathrm{lin}(x_i, x_{i+1}, C_1, \ldots, C_m)$. Now consider the word $w = t_1 t_2 \ldots t_k$ with $x_i \in M_{t_i}$ for all i. Due to the construction of less$'$ from less and Lemma 11.1.2, $(P, \leq) \cong (\mathbb{J}(\downarrow(\mathrm{nat}(w)), \leq))$. Hence we found a monoid element $x = \mathrm{nat}(w)$ such that $(P, \leq) \cong (\mathbb{J}(\downarrow m, \leq)$.

Finally, let $\overline{\varphi}$ denote the formula

$$\begin{aligned} \exists_{1 \leq i \leq m} C_i \quad (\quad & \bigcup_{1 \leq i \leq m} C_i = \text{ everything} \wedge \\ & C_i \cap C_j = \emptyset \text{ for } 1 \leq i < j \leq m \wedge \\ & \varphi \\ ). \end{aligned}$$

By Lemma 10.2.2, any partially ordered set $\mathbb{J}(\downarrow x, \leq)$ for $x \in M$ has width at most m. Hence by Dilworth' Theorem, there are mutually disjoint chains C_i that cover P. Now the statement of the theorem follows by the consideration above. □

Since the set of join-irreducible elements of a distributive lattice is definable inside the lattice, we obtain as a direct consequence of the theorem above the following

Corollary 11.1.7 *Let $(M, \cdot, 1)$ be a divisibility monoid with finite commutation behavior. There exists a monadic sentence ψ such that for any finite partial order $(P, \leq)$ it holds:*

$$(P, \leq) \models \psi \iff \exists x \in M : (\downarrow x, \leq) \cong (P, \leq).$$

Let $(M, \cdot, 1)$ be a width-bounded divisibility monoid. To show that in this case the monadic theory $\mathrm{MTh}(\{(\downarrow x, \leq) \mid x \in M\})$ is decidable, we now show that the set of lattices $\{(\downarrow x, \leq) \mid x \in M\}$ is contained in a set of lattices whose monadic theory is decidable. Then, by the corollary above, the decidability of $\mathrm{MTh}(\{(\downarrow x, \leq) \mid x \in M\})$ follows easily:

An undirected graph (T, K) is a *tree* if for any $s, t \in T$, there is a unique path connecting s and t. Now let (V, E) be a finite directed graph and $n \in \mathbb{N}$. Then (V, E) has *tree-width at most n* if there exists a tree (T, K) and a mapping $\psi : T \to 2^V$ such that

1. for any $(x, y) \in E$, there is $t \in T$ with $x, y \in \psi(T)$,
2. for any $s, t, u \in T$ such that t is on the path connecting s and u, we have $\psi(s) \cap \psi(u) \subseteq \psi(t)$,
3. $\bigcup_{s \in T} \psi(i) = V$, and
4. $|\psi(t)| < n$ for any $t \in T$.

Lemma 11.1.8 *Let $n \in \mathbb{N}$ and $(L, \leq)$ a finite distributive lattice of width at most n. Then the graph $(L, -\!\!\!<)$ has tree width at most $2n$.*

Proof. Let m denote the length of L. The tree (T, K) that we construct is (the Hasse diagram of) the linear order on $\{1, 2, \ldots, m\}$. Let $\psi(i)$ be the set of all vertices in $(L, \leq)$ of height $i - 1$ or i.

Now let $x, y \in L$ with $x -\!\!\!< y$. Then, since L is distributive, $h(y) - h(x) = 1$, i.e. $x, y \in \psi(h(y))$. Hence the first property is satisfied. For the second note that $\psi(i) \cap \psi(k) = \emptyset$ whenever there is $i < j < k$. Hence it is trivially satisfied. Similarly, the third requirement holds trivially. Finally, $\psi(i)$ consists of two antichains. Since the size of these antichains is bounded by n, the last requirement $|\psi(i)| \leq 2n$ follows. □

Theorem 11.1.9 *Let $(M, \cdot, 1)$ be a divisibility monoid with finite commutation behavior. Then the monadic theory $\mathrm{MTh}\{(\downarrow m, \leq) \mid m \in M\}$ is decidable iff M is width-bounded.*

Proof. First, let $(M, \cdot, 1)$ be width-bounded by n. Then any lattice $(\downarrow x, \leq)$ has tree-width at most $2n$ by the preceding lemma. Now let μ be a monadic sentence. Then, by Corollary 11.1.7, μ belongs to $\mathrm{MTh}(\{(\downarrow x, \leq) \mid x \in M\})$ iff $\psi \to \mu$ is satisfied by all finite distributive lattices of tree width at most $2n$. But this question is decidable by [Cou90].

If, on the other hand, $(M, \cdot, 1)$ is not width-bounded, by Theorem 10.2.6, any grid $([n]^2, \leq)$ can be embedded into some lattice $(\downarrow x, \leq)$. Since the monadic theory of these grids is undecidable, the monadic theory of all lattices $(\downarrow x, \leq)$ with $x \in M$ is undecidable. □

Let $(M, \cdot, 1)$ be a divisibility monoid and let $\mathfrak{L}$ denote the set of all distributive lattices $(\downarrow x, \leq)$ for $x \in M$. Then, by Theorem 11.1.3, the monadic theory of $\mathbb{J}(\mathfrak{L}) := \{\mathbb{J}(L, \leq) \mid (L, \leq) \in \mathfrak{L}\}$ is decidable. By the theorem above, $\mathrm{MTh}(\mathfrak{L})$ is decidable iff the width of the elements of $\mathfrak{L}$ is uniformly bounded. As an encore which is not directly related to divisibility monoids, we show in the following two sections that this last connection between the bounded width of a class of distributive lattices $\mathfrak{L}$ and the decidability of $\mathrm{MTh}(\mathbb{J}(\mathfrak{L}))$ holds in general and is not a particular feature of divisibility monoids.

11.2 The semilattice of finitely generated ideals

It is the aim of this section to relate the monadic theory of a set of partially ordered sets to the monadic theory of the semilattices of finitely generated ideals that are associated with these partially ordered sets. In particular, we are interested in the relation between the decidabilities of these theories.

Remark 11.2.1 Let $\mathfrak{P}$ be a set of partially ordered sets. Then $\mathrm{MTh}(\mathfrak{P})$ can be reduced in linear time to $\mathrm{MTh}(\mathbb{H}_f(\mathfrak{P}))$.

Proof. Recall that a partially ordered set $(P, \leq)$ is isomorphic to $\mathbb{J}\,\mathbb{H}_f(P, \leq)$. Hence, a sentence is satisfied by $(P, \leq)$ iff its restriction to the join-irreducible elements is valid in $\mathbb{H}_f(P, \leq)$. Since this restriction can be computed in linear time, the statement follows. □

Theorem 11.2.2 *Let $\mathfrak{P}$ be a set of partially ordered sets and $n \in \mathbb{N}$ such that $w(P, \leq) \leq n$ for any $(P, \leq) \in \mathfrak{P}$. Then $\mathrm{Th}(\mathbb{H}_f(\mathfrak{P}))$ can be reduced to $\mathrm{Th}(\mathfrak{P})$ in linear time.*

Proof. The idea of the proof is that any finitely generated ideal in $(P, \leq)$, i.e. any element of $\mathbb{H}_f(P, \leq)$ is generated by at most n elements of P. Therefore, the reduction r is defined by

$$\begin{aligned} r(\exists x \alpha) &= (\exists x_1 \exists x_2 \ldots \exists x_n r(\alpha)), \\ r(x \leq y) &= (\bigwedge_{1 \leq i \leq n} \bigvee_{1 \leq j \leq n} x_i \leq y_j), \\ r(\alpha \vee \beta) &= (r(\alpha) \vee r(\beta)), \text{ and} \\ r(\neg \alpha) &= \neg r(\alpha). \end{aligned}$$

Identifying a tuple $(x_1, x_2, \ldots, x_n)$ in P with its ideal $x_1\!\downarrow \cup x_2\!\downarrow \cup \cdots \cup x_n\!\downarrow$, one easily verifies that

$$\mathbb{H}_f(P, \leq) \models \varphi \iff (P, \leq) \models r(\varphi)$$

for any elementary sentence φ and any $(P, \leq) \in \mathfrak{P}$. Hence, in particular, r reduces $\mathrm{Th}(\mathbb{H}_f(\mathfrak{P}))$ to $\mathrm{Th}(\mathfrak{P})$ in linear time. □

As an immediate consequence of the above, we obtain

Corollary 11.2.3 *Let $\mathfrak{P}$ be a set of partially ordered sets and $n \in \mathbb{N}$ such that $w(P, \leq) \leq n$ for any $(P, \leq) \in \mathfrak{P}$. Then $\mathrm{Th}(\mathbb{H}_f(\mathfrak{P}))$ is decidable iff $\mathrm{Th}(\mathfrak{P})$ is decidable.*

11.2.1 From $\mathrm{MTh}(\mathbb{H}_f(\mathfrak{P}))$ to $\mathrm{MTh}(\mathfrak{P})$

Our next aim is to show a similar result for the monadic theories $\mathrm{MTh}(\mathbb{H}_f(\mathfrak{P}))$ and $\mathrm{MTh}(\mathfrak{P})$. Note that the basic idea of the reduction of $\mathrm{Th}(\mathbb{H}_f(\mathfrak{P}))$ to $\mathrm{Th}(\mathfrak{P})$ is the replacement of an element of $\mathbb{H}_f(P, \leq)$ by an n-tuple in $(P, \leq)$. If we want to extend this to sets of elements in $\mathbb{H}_f(P, \leq)$, it would be natural to consider n-ary relations in $(P, \leq)$. But this is not possible in monadic second order logic. Therefore, we have to perform some extra work to encode certain n-ary relations using sets of elements of P, only. This is the content of the following considerations that lead to Corollary 11.2.9.

Lemma 11.2.4 *Let $(C, \leq)$ be a linearly ordered set and let $k \in \mathbb{N}$. Then C splits into $2k$ mutually disjoint subsets $C(j)$ $(1 \leq j \leq 2k)$ satisfying*
$(\star)$ *For any $1 \leq j \leq 2k$ and for any $x, y \in C(j)$ with $x < y$, the interval $x\!\uparrow \cap y\!\downarrow \subseteq C$ contains at least k elements.*

Proof. Let α be an ordinal and let $C = \{x_\beta \mid \beta < \alpha\}$ be an enumeration of C. By transfinite induction, we construct the subsets $C(j)$ as follows: Let $\beta < \alpha$ and assume that we constructed a partition $(C^\beta(j))_{1 \leq j \leq 2k}$ of $\{x_\gamma \mid \gamma < \beta\}$ satisfying $(\star)$. Consider the sets

$$\begin{aligned} M &= \{x \in C \mid x < x_\beta, |x\!\uparrow \cap x_\beta\!\downarrow| \leq k\} \text{ and} \\ N &= \{x \in C \mid x > x_\beta, |x_\beta\!\uparrow \cap x\!\downarrow| \leq k\}. \end{aligned}$$

Since they are linearly ordered, M and N both contain at most $k-1$ elements. Hence there is $1 \leq j \leq 2k$ with $C^\beta(j) \cap (M \cup N) = \emptyset$. Now define $C^{\beta^+}(j) := C^\beta(j) \cup \{x_\beta\}$ and $C^{\beta+1}(i) := C^\beta(i)$ for $i \neq j$. Then $(C^{\beta+1}(j))_{1 \leq j \leq 2k}$ is a partition of $\{x_\gamma \mid \gamma < \beta + 1\}$ satisfying $(\star)$. For a limit ordinal β, we set $C^\beta(j) := \bigcup_{\gamma<\beta} C^\gamma(j)$ for $1 \leq j \leq 2k$. Now $C(j) := C^\alpha(j)$ finishes the construction. □

Definition 11.2.5 The partial order $(P, \leq)$ has *diabolo width at most m* if, for any $X, Y \subseteq P$ such that $X \times Y \subseteq \|$, we have $|X| \leq m$ or $|Y| \leq m$.

The Figure 11.2 depicts this notion: Let $(P, \leq)$ be a partially ordered set of diabolo width at most m and let $X \subseteq P$ be a set width more than m elements. Then the set $Y := P \setminus (X\!\uparrow \cup X\!\downarrow)$ is incomparable with X. Hence it contains at most m elements.[1]

Note that the width is at most double the diabolo width of a partially ordered set.

[1]The name "diabolo width" was chosen since in this picture the set $X\!\uparrow \cup X\!\downarrow$ looks like a diabolo – a juggling prop that the author hopes to master eventually.

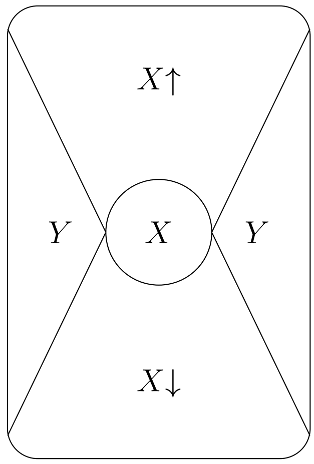


Figure 11.2: Diabolo width

Let $(P, \leq)$ be a partial order and let $C_1, C_2 \subseteq P$ be chains in $(P, \leq)$. We define an equivalence relation $\sim$ on C_2 by $x \sim y$ iff $C_1 \cap x\updownarrow = C_1 \cap y\updownarrow$, i.e. iff x and y are comparable with the same elements of the first chain C_1.

Lemma 11.2.6 *Let $(P, \leq)$ be a partial order of diabolo width at most m. Let $C_1, C_2 \subseteq P$ be chains in $(P, \leq)$ and let $k = (2m+3)^2$. Let $x_i \in C_2$ with $x_i \not\sim x_{i+1}$ and $x_i < x_{i+1}$ for $1 \leq i < k$. Then $C_1 \subseteq x_1\uparrow \cup x_k\downarrow$, i.e. there is no element in C_1 that is incomparable with both x_1 and x_k.*

Proof. For $1 \leq i \leq k$, let $\Downarrow x_i = \{x \in C_1 \mid x < x_i\}$ and $\Uparrow x_i = \{x \in C_1 \mid x > x_i\}$. Then $\Downarrow x_i \cup \Uparrow x_i$ consists of those elements from $C_1 \setminus \{x_i\}$ that are comparable with x_i. Hence $\Downarrow x_i \subseteq \Downarrow x_{i+1}$ and $\Uparrow x_i \supseteq \Uparrow x_{i+1}$. Since $x_i \not\sim x_{i+1}$ and $x_i < x_{i+1}$, we have $\Downarrow x_i \subsetneq \Downarrow x_{i+1}$ or $\Uparrow x_i \supsetneq \Uparrow x_{i+1}$ for $1 \leq i < k$. First we show $\Downarrow x_i \subsetneq \Downarrow x_{i+2m+3}$ for $1 \leq i \leq k - 2m - 3$:

By contradiction, assume $1 \leq i \leq k - 2m - 3$ with $\Downarrow x_i = \Downarrow x_{i+2m+3}$. Let $x \in X := \Uparrow x_i \setminus \Uparrow x_{i+m+1}$ and let $m + 1 < \ell \leq 2m + 3$. If x is comparable with $x_{i+\ell}$, we get $x < x_{i+\ell}$ for otherwise $x_{i+m+1} < x_{i+\ell} \leq x$. Thus $x \in \Downarrow x_{i+\ell} = \Downarrow x_i$, i.e. $x \in \Downarrow x_i \cap \Uparrow x_i$, a contradiction since this set is empty. Thus, any element of X is incomparable with $x_{i+\ell}$ for $m + 1 < \ell \leq 2m + 3$, i.e. X and $Y := \{x_{i+m+2}, x_{i+m+3} \ldots x_{i+2m+3}\}$ are incomparable sets. Since Y contains more than m elements and since the diabolo width of $(P, \leq)$ is m, X contains at most m elements implying that there is $0 \leq j < m$ with $\Uparrow x_{i+j} \setminus \Uparrow x_{i+j+1} = \emptyset$. But this contradicts our assumption that $x_{i+j} \not\sim x_{i+j+1}$. Thus, we proved $\Downarrow x_i \subsetneq \Downarrow x_{i+2m+3}$.

Now let $y_i := x_{(2m+3)i}$ for $1 \le i \le \ell := \frac{k}{2m+3} = 2m+3$, i.e. $y_1 < y_2 < \cdots < y_\ell$ is a subsequence of $x_1 < x_2 < \cdots < x_k$ such that $\Downarrow y_i \subsetneq \Downarrow y_{i+1}$ for $1 \le i < \ell$.

To prove the final goal $C_1 \subseteq \uparrow x_1 \cup \downarrow x_k$, assume by contradiction that x is an element of $C_1 \setminus (x_1 \uparrow \cup x_k \downarrow)$ and let $1 \le i \le m+1$ and $z \in Z := \Downarrow y_{2m+3} \setminus \Downarrow y_{m+2}$. Since z and x belong to the chain C_1, they are comparable. Then $z < x$ for otherwise $x \le z \le y_{2m+3} \le x_k$ would contradict our assumption on x. In case $z \ge y_i$ ($z \le y_i$), we had $x > z \ge y_i \ge x_1$ ($z \le y_i \le y_{m+2}$), contradicting our assumption on x (on z, respectively). Hence z and y_i are incomparable, i.e. the sets $\{y_1, y_2, \ldots, y_{m+1}\}$ and Z are incomparable. Since they both contain more than m elements, the diabolo width of $(P, \le)$ is larger than m, a contradiction.□

Let $(P, \le)$ be a partially ordered set. By a slight abuse of notation, we call an n-tuple $(x_1, x_2, \ldots, x_n) \in P^n$ an *antichain* if the set $\{x_1, x_2, \ldots, x_n\}$ is an antichain. Note that in particular the tuple $(a, a, \ldots, a)$ is an antichain for any $a \in P$. By Antichains, we denote the set of all *tuples* that are antichains.

Lemma 11.2.7 *Let $(P, \le)$ be a partial order of diabolo width at most m. Let $C_1, C_2 \subseteq P$ be chains in $(P, \le)$. For any $M \subseteq C_1 \times C_2 \cap \text{Antichains}$, there exist sets $M_{i,j} \subseteq C_i$ for $i = 1, 2$ and $0 \le j < 4m(2m+3)^2$ such that*

$$M = \left(\bigcup_{j=0}^{4m(2m+3)^2} M_{1,j} \times M_{2,j} \right) \cap \text{Antichains}.$$

Proof. Similarly to the preceding lemma, let $k = (2m+3)^2$. First, we split the chain C_2 into the set C_2^s of those elements that belong to a small $\sim$-equivalence class and its complement, i.e.

$$\begin{aligned} C_2^s &= \{y \in C_2 : |[y]| \le m\}, \text{ and} \\ C_2^l &= C_2 \setminus C_2^s = \{y \in C_2 : |[y]| > m\} \end{aligned}$$

where $[y]$ denotes the $\sim$-equivalence class containing y. Using Lemma 11.2.4, next we split C_1 and C_2^s into $2 \cdot k \cdot m$ disjoint subchains $C_1(j)$ and $C_2^s(j)$ for $1 \le j \le 2km$ such that, for any $x, y \in C_1(j)$ ($\in C_2^s(j)$, respectively) with $x < y$ the interval $\uparrow x \cap \downarrow y$ contains at least km elements from $C_1(j)$ (from $C_2^s(j)$, respectively). This ensures that between any two elements of $C_1(j)$, there are more than m elements of C_1. Similarly, we will use that between two elements of $C_2^s(j)$ there are at least $k \cdot m$ elements of C_2^s. To finish the construction, let

$$\begin{aligned} M^l(j) &= M \cap C_1(j) \times C_2^l, \text{ and} \\ M^s(j) &= M \cap C_1 \times C_2^s(j) \end{aligned}$$

for $1 \leq j \leq 2km$. We establish the lemma showing that M equals the set of antichains that occur in

$$H = \bigcup_{1 \leq j \leq 2km} \left[\pi_1(M^l(j)) \times \pi_2(M^l(j)) \cup \pi_1(M^s(j)) \times \pi_2(M^s(j))\right].$$

Let $(x,y) \in M$. In case $y \in C_2^l$, there is $1 \leq j \leq 2km$ with $x \in C_1(j)$. Hence $(x,y) \in M \cap C_1(j) \times C_2^l = M^l(j)$. Now $(x,y) \in \pi_1(M^l(j)) \times \pi_2(M^l(j)) \subseteq H$ follows immediately. In case $y \in C_2^s$, we find $1 \leq j \leq 2km$ with $y \in C_2^s(j)$. Now $(x,y) \in \pi_1(M^s(j)) \times \pi_2(M^s(j)) \subseteq H$ follows, i.e. we showed $M \subseteq H \cap \text{Antichains}$.

Conversely, we have to show that antichains from $\pi_1(M^l(j)) \times \pi_2(M^l(j))$ or from $\pi_1(M^s(j)) \times \pi_2(M^s(j))$ belong to M for any $1 \leq j \leq 2km$. So let $1 \leq j \leq 2km$ and $(x_1,x_2),(y_1,y_2) \in M^l(j)$ with $x_1 \parallel y_2$. We want to show $x_1 = y_1$ implying $(x_1,y_2) = (y_1,y_2) \in M$. By contradiction assume $x_1 \neq y_1$. Since $x_1, y_1 \in C_1(j)$, they are comparable. We assume $x_1 < y_1$ (the case $y_1 < x_1$ is dual). As remarked earlier, there are more than m elements of C_1 between x_1 and y_1, in particular $|x_1\!\uparrow \cap y_1\!\downarrow| > m$. All elements of this interval are incomparable with y_2 since its endpoints x_1 and y_1 are. Thus we found incomparable sets $x_1\!\uparrow \cap y_1\!\downarrow$ and $[y_2]$ both larger than m. Since this contradicts the assumption on the diabolo width of $(P,\leq)$, we obtain $x_1 = y_1$ and therefore

$$\pi_1(M^l(j)) \times \pi_2(M^l(j)) \cap \text{Antichains} \subseteq M \text{ for any } 1 \leq j \leq 2km.$$

Finally, let $1 \leq j \leq 2km$ and $(x_1,x_2),(y_1,y_2) \in M^s(j)$ with $x_1 \parallel y_2$. To show $x_2 = y_2$, we now assume by contradiction $x_2 < y_2$. Similarly to above, there are at least km elements of C_2^s in the interval $x_2\!\uparrow \cap y_2\!\downarrow$. Since $|[x]| \leq m$ for any $x \in C_2^s$, the chain $C_2^s \cap x_2\!\uparrow \cap y_2\!\downarrow$ contains k mutually not $\sim$-equivalent elements. Hence, by Lemma 11.2.6, $C_1 \setminus (x_2\!\uparrow \cap y_2\!\downarrow) = \emptyset$, contradicting $x_1 \in C_1$ and $x_2 \parallel x_1 \parallel y_2$.□

Theorem 11.2.8 *Let $(P,\leq)$ be a partial order of diabolo width at most m and let $n > 1$. Let $C_i \subseteq P$ be chains for $1 \leq i \leq n$ and let $M \subseteq \prod_{1 \leq i \leq n} C_i \cap \text{Antichains}$. Then M is the intersection of* Antichains *with $(2m+2)^n$ sets of the form*

$$\bigcap_{1 \leq a < b \leq n} \bigcup_{\ell=1}^{4m(2m+3)^2} \left[P^{a-1} \times M_\ell^{a,b} \times P^{b-a-2} \times N_\ell^{a,b} \times P^{n-b-1}\right] \qquad (\star)$$

where $M_\ell^{a,b}, N_\ell^{a,b} \subseteq P$ for all suitable a, b and ℓ.

Proof. By Lemma 11.2.4, we split the chains C_i into $2m+2$ disjoint subchains $C_i(j)$ for $1 \leq j \leq 2m$ such that for any $x, y \in C_i(j)$ with $x < y$ the interval $\uparrow\! x \cap \downarrow\! y$ contains at least $m+1$ elements from P. For $\vec{\jmath} \in \{1,2,\ldots,2m+2\}^n$ let $M_{\vec{\jmath}} = M \cap \prod_{1 \leq i \leq n} C_i(j_i)$. Then M is the union of the sets $M_{\vec{\jmath}}$. Since there are $(2m+2)^n$ sets $M_{\vec{\jmath}}$, it suffices to show that any such set is the intersection of

Antichains with a set of the form ($\star$). Let $1 \leq a < b \leq n$. Applying Lemma 11.2.7 to the set $\pi_{a,b}(M_{\vec{\jmath}})$ and the chains $C_a(j_a)$ and $C_b(j_b)$, we obtain the existence of sets $M_\ell^{a,b} \subseteq C_a(j_a)$ and $N_\ell^{a,b} \subseteq C_b(j_b)$ for $1 \leq \ell \leq 4m(2m+3)^2$ such that

$$\pi_{a,b}(M_{\vec{\jmath}}) = \bigcup_{\ell=1}^{4m(2m+3)^2} M_\ell^{a,b} \times N_\ell^{a,b} \cap \text{Antichains}.$$

Let $H_{\vec{\jmath}}$ denote the set of all antichains in

$$\bigcap_{1 \leq a < b \leq n} \bigcup_{\ell=1}^{4m(2m+3)^2} \left[P^{a-1} \times M_\ell^{a,b} \times P^{b-a-2} \times N_\ell^{a,b} \times P^{n-b-1} \right].$$

Note that $H_{\vec{\jmath}}$ equals the set of all antichains $(x_1, x_2, \ldots, x_n) \in P^n$ such that $(x_a, x_b) \in \pi_{a,b}(M_{\vec{\jmath}})$ for any $1 \leq a < b \leq n$. Thus, in particular, $H_{\vec{\jmath}}$ is a subset of $\prod_{1 \leq i \leq n} C_i(j_i)$ since $\pi_{a,b}(M_{\vec{\jmath}}) \subseteq C_a(j_a) \times C_b(j_b)$. Since $H_{\vec{\jmath}}$ is of the form ($\star$), it remains to show that $M_{\vec{\jmath}} = H_{\vec{\jmath}}$.

The inclusion $M_{\vec{\jmath}} \subseteq H_{\vec{\jmath}}$ is immediate for any element $(x_1, x_2, \ldots, x_n)$ of $M_{\vec{\jmath}}$ is an antichain satisfying $(x_a, x_b) \in \pi_{a,b}(M_{\vec{\jmath}})$ for all suitable a, b.

We show by induction on the size of $I \subseteq \{1, 2, \ldots, n\}$ that $\pi_I(H_{\vec{\jmath}}) \subseteq \pi_I(M_{\vec{\jmath}})$ which, for $I = \{1, 2, \ldots, n\}$ establishes the claim and therefore the theorem. If I contains precisely two elements, the inclusion $\pi_I(H_{\vec{\jmath}}) \subseteq \pi_I(M_{\vec{\jmath}})$ is immediate by what we said above. Now let I contain at least three elements and assume that $\pi_J(H_{\vec{\jmath}}) \subseteq \pi_J(M_{\vec{\jmath}})$ for any proper subset J of I. For notational simplicity, we assume $I = \{1, 2, \ldots, c\}$ for some $3 \leq c \leq n$. Let $(x_1, x_2, \ldots, x_c) \in \pi_I(H_{\vec{\jmath}})$. Then, by the induction hypothesis, there are elements $x_i^i \in C_i(j_i)$ such that

$$(x_1, \ldots, x_{i-1}, x_i^i, x_{i+1}, \ldots, x_c) \in \pi_I(M_{\vec{\jmath}})$$

for any $i \in I$. If for some $1 \leq i \leq c$ we even have $x_i = x_i^i$, we thus get immediately $(x_1, \ldots, x_c) \in \pi_I(M_{\vec{\jmath}})$. Now assume $x_i \neq x_i^i$ for all $1 \leq i \leq c$. Since $x_i, x_i^i \in C_i(j_i)$, they are comparable. Since I contains at least three elements, there are $1 \leq a < b \leq c$ with $x_a < x_a^a$ and $x_b < x_b^b$ or with $x_a > x_a^a$ and $x_b > x_b^b$. By symmetry, it suffices to deal with the first case. Recall that $x_a \not\lessgtr x_b$ since $(x_1, \ldots, x_c)$ (as an element of $\pi_I(H_{\vec{\jmath}})$) is an antichain. Similarly, $x_a^a \not\lessgtr x_b$ and $x_b^b \not\lessgtr x_a$ since $(x_1, \ldots, x_{a-1}, x_a^a, x_{a+1}, \ldots, x_c)$ and $(x_1, \ldots, x_{b-1}, x_b^b, x_{b+1}, \ldots, x_c)$ are antichains as elements of $\pi_I(M_{\vec{\jmath}})$. Thus x_a^a and x_b^b are incomparable for otherwise $x_a^a \leq x_b^b$ implied $x_a < x_b^b$. Since $\{x_a, x_a^a\}$ and $\{x_b, x_b^b\}$ are incomparable sets, so are the intervals $x_a{\uparrow} \cap x_a^a{\downarrow}$ and $x_b{\uparrow} \cap x_b^b{\downarrow}$. Recall that $x_a < x_a^a$ are both elements of $C_a(j_a)$. Thus, the interval $x_a{\uparrow} \cap x_a^a{\downarrow}$ consists of at least $m+1$ elements, and similarly the interval $x_b{\uparrow} \cap x_b^b{\downarrow}$. This contradicts the assumption that $(P, \leq)$ has diabolo width at most m. Therefore, it is impossible that $x_i \neq x_i^i$ for all $1 \leq i \leq c$. This finishes the induction step, i.e. we have indeed $\pi_I(H_{\vec{\jmath}}) \subseteq \pi_I(M_{\vec{\jmath}})$. $\square$

Corollary 11.2.9 *Let $m \in \mathbb{N}$ and $n = 2m$. Then there exists a natural number ℓ and a monadic formula $\varphi(x_1, \ldots, x_n, X_1, X_2, \ldots, X_\ell)$ such that, for any partially ordered set $(P, \leq)$ of diabolo width at most m and any set R of antichains in P, there are sets $M_1, \ldots, M_\ell \subseteq P$ with*

$$R = \{\{x_1, x_2, \ldots, x_n\} \subseteq P \mid (P, \leq) \models \varphi(x_1, \ldots, x_n, M_1, \ldots M_\ell\}.$$

Proof. We explain the idea of the formula and leave the technicalities to the interested reader: We are concerned with partially ordered sets of width at most n, only. Hence any element of R contains at most n elements. By Dilworth' Theorem [Dil50], the partially ordered set $(P, \leq)$ can be covered by n disjoint chains $C_1, \ldots, C_n$. For $I \subseteq \{1, 2, \ldots, n\}$, let R_I denote the set of all antichains in R that meet a chain C_i iff $i \in I$. In particular, the set R_I contains sets of size $|I|$, only. Thus, we can identify it with an $|I|$-ary relation on P such that $R_I \subseteq \prod_{i \in I} C_i$. Now, applying Theorem 11.2.8, we easily construct a formula φ_I with $|I|$ free elementary variables and $(2m+2)^{|I|} n(n-1)4m(2m+3)^2 \cdot 2$ free set variables such that there exist sets M_i with

$$R_I = \{(y_1, \ldots, y_{|I|}) \mid (P, \leq) \models \varphi_I(y_i, M_i)\}.$$

The formula φ is a simple Boolean combination of the formulas φ_I. □

Theorem 11.2.10 *Let $\mathfrak{P}$ be a set of partially ordered sets and $m \in \mathbb{N}$ such that any $(P, \leq)$ in $\mathfrak{P}$ has diabolo width at most m. Then $\mathrm{MTh}(\mathbb{H}_f(\mathfrak{P}))$ can be reduced to $\mathrm{MTh}(\mathfrak{P})$ in linear time.*

Proof. Let φ and ℓ denote the formula and the natural number from Corollary 11.2.9. The reduction r is defined by

$$\begin{aligned}
r(\exists x\alpha) &= (\exists x_1 \exists x_2 \ldots \exists x_n r(\alpha)), \\
r(x \leq y) &= (\bigwedge_{1\leq i\leq n} \bigvee_{1\leq j\leq n} x_i \leq y_j), \\
r(\exists M\alpha) &= (\exists M_1 \exists M_2 \ldots \exists M_\ell r(\alpha)), \\
r(x \in M) &= \varphi(x_1, x_2, \ldots, x_n, M_1, M_2, \ldots, M_\ell), \\
r(\alpha \vee \beta) &= (r(\alpha) \vee r(\beta)), \text{ and} \\
r(\neg\alpha) &= \neg r(\alpha).
\end{aligned}$$

Differently from the proof of Theorem 11.2.2, we spell out the equivalence $\mathbb{H}_f(P, \leq) \models \varphi \iff (P, \leq) \models r(\varphi)$ in some more detail:

Let V be a countable set of individual variables and W that of set variables. We use these variables in monadic formulas that are interpreted over $\mathbb{H}_f(P, \leq)$.

By $V' := V \times \{1, 2, \ldots, n\}$ and $W' := W \times \{1, 2, \ldots, \ell\}$ we denote the individual and set variables when speaking on the partial orders in $\mathfrak{P}$. For simplicity, we abbreviate (x, i) by x_i and similarly (A, j) by A_j for $x \in V$ and $A \in W$.

Let $(P, \leq)$ be a partially ordered set in $\mathfrak{P}$. With any $a \in \mathbb{H}_f(P, \leq)$, we associate an n-tuple $f'(a)$ in P with $\{f'(a)_1, f'(a)_2, \ldots, f'(a)_n\} = \max(a)$. Such an n-tuple exists since a is a finitely generated ideal in $(P, \leq)$ implying that it has at most n maximal elements. Furthermore, $a = \bigcup_{1 \leq i \leq n} f'(a)_i\!\downarrow$. Note that the coordinates of the tuple $f'(a)$ are incomparable if not equal.

Similarly, we find a function g' that maps subsets of $\mathbb{H}_f(P, \leq)$ to ℓ-tuples of subsets of P as follows: Let $M \subseteq \mathbb{H}_f(P, \leq)$ be a set of finitely generated ideals in $(P, \leq)$. By R, we denote the set of all $(\leq n)$-subsets $\max(a)$ of P for some $a \in M$, i.e. $R = \{\max(a) \mid a \in M\}$. Then R is a set of antichains in the partially ordered set $(P, \leq)$ of diabolo width at most m. Hence, by Corollary 11.2.9 there exist sets $M_1, M_2, \ldots, M_\ell \subseteq P$ with

$$R = \{\{x_1, x_2, \ldots, x_n\} \subseteq P \mid (P, \leq) \models \varphi(x_1, \ldots, x_n, M_1, \ldots M_\ell\}.$$

For $1 \leq j \leq \ell$, let $g'(M)_j := M_j$. Then we obtain for any $M \subseteq \mathbb{H}_f(P, \leq)$ and any $a \in \mathbb{H}_f(P, \leq)$:

$$a \in M \text{ iff } (P, \leq) \models \varphi(f'(a), g'(M)).$$

Now let (f, g) be an interpretation of the elementary variables V and the set variables W in $\mathbb{H}_f(P, \leq)$, i.e. $f : V \to \mathbb{H}_f(P, \leq)$ and $g : W \to 2^{\mathbb{H}_f(P, \leq)}$. By $f^\star(x_i) := (f' \circ f(x))_i$ and $g^\star(M_j) := (g' \circ g(M))_j$ for $x_i \in V'$ and $M_j \in W'$, we define an interpretation $(f^\star, g^\star)$ of V' and W' in $(P, \leq)$ from (f, g).

To finish the proof, one shows by induction on the monadic formula φ that $\mathbb{H}_f(P, \leq) \models_{(f,g)} \varphi$ iff $(P, \leq) \models_{(f^\star, g^\star)} r(\varphi)$. This is an easy exercise which is left to the reader. ☐

11.2.2 Decidable monadic theory implies bounded diabolo width

The theorem above implies in particular that $\mathrm{MTh}(\mathbb{H}_f(\mathfrak{P}))$ is decidable whenever $\mathrm{MTh}(\mathfrak{P})$ is decidable and the diabolo width of the elements of $\mathfrak{P}$ is bounded above. In this section, we will show that also the other implication holds. In addition, we consider the question when the monadic chain and the monadic antichain theories are decidable.

It will be convenient to use the notation $\underline{m}$ for the set $\{0, 1, \ldots, m\}$ of non-negative integers properly smaller than m. Let $\mathcal{G}$ denote the set of all finite grids, seen as distributive lattices, i.e. the set of all distributive lattices $(\underline{m}, \leq) \times (\underline{n}, \leq)$ for $m, n > 1$. Given a Turing machine M, one can easily formulate a monadic sentence which is satisfied by the partially ordered set $L = (\underline{m} \times \underline{n}, \leq)$ iff the machine stops after m steps using n cells of the tape. Hence there is a monadic sentence μ such that $\mu \in \mathrm{MTh}(\mathcal{G})$ iff M does not stop, i.e. the monadic theory of $\mathcal{G}$ is undecidable. For this encoding, one has to quantify over arbitrary subsets of the grid. In the sequel, we will show that not only the monadic theory of the grids, but also their monadic antichain and their monadic chain theory are undecidable.

The antichain theory

Lemma 11.2.11 *The monadic theory of $\mathcal{G}$ can be reduced to the monadic antichain theory of $\mathcal{G}$ in linear time.*

Proof. The *grid graph of dimension* (m, n) is the structure $(\underline{m} \times \underline{n}, E_{m,n})$ with $((i, j), (i', j')) \in E_{m,n}$ iff $i = i'$ and $j + 1 = j'$ or $i + 1 = i$ and $j = j'$ for any $(i, j), (i', j') \in \underline{m} \times \underline{n}$. Let $\mathcal{GG}$ denote the set of all structures isomorphic to some grid graph. (Note that for technical convenience, for grid graphs we allow $m = n = 1$ but not for the distributive lattices). Note that the grid graph of dimension (m, n) is the Hasse diagram of the grid $(\underline{m} \times \underline{n}, \leq)$. Hence it suffices to reduce the monadic theory of the set of all grid graphs to the monadic antichain theory of $\mathcal{G}$.

For a monadic sentence φ over the binary relation symbol E, we construct a monadic formula over the vocabulary $\leq$ as follows: First, we restrict the quantification in φ to the new set variable X. Afterwards, any subformula of the form $(x, y) \in E$ is replaced by

$$\sup(x, y) \in E_1 \cup E_2 \wedge \inf(x, e) \leq \inf(y, e).$$

Let φ' denote the result of this procedure. Then φ' is a monadic formula with free variables contained in $\{X, E_1, E_2, e\}$.

Now we describe the reduction of $\mathrm{MTh}(\mathcal{GG})$ to $\mathrm{MATh}(\mathcal{G})$: It is easily seen that there is a monadic sentence γ such that a graph (X, E) satisfies γ iff it belongs to $\mathcal{GG}$. Let φ be a monadic sentence over the binary relation symbol E. Then define

$$\overline{\varphi} := \forall X \forall E_1 \forall E_2 \forall e (e \in \mathbb{J} \rightarrow (\gamma' \rightarrow \varphi')).$$

We show that φ belongs to $\mathrm{MTh}(\mathcal{GG})$ iff $\overline{\varphi}$ belongs to $\mathrm{MATh}(\mathcal{G})$:

First let $\varphi \in \mathrm{MTh}(\mathcal{GG})$. Furthermore, let $m, n > 1$ and $L := (\underline{m} \times \underline{n}, \leq)$. Let $X, E_1, E_2 \subseteq L$ be antichains and $e \in \mathbb{J}(L)$ with $(L, \leq) \models_A \gamma'(X, E_1, E_2, e)$. We have to show that $(L, \leq) \models \varphi'(X, E_1, E_2, e)$. First, define a binary relation $E \subseteq X^2$ by $(x, y) \in E$ iff $\sup(x, y) \in E_1 \cup E_2$ and $\inf(x, e) \leq \inf(y, e)$. By the construction of γ' and the fact that $(L, \leq) \models_A \gamma'(X, E_1, E_2, e)$, the graph (X, E) satisfies γ, i.e. it is isomorphic to a grid graph. Hence $(X, E) \models \varphi$ implying $(L, \leq) \models_A \varphi'(X, E_1, E_2, e)$ as required.

Conversely, let $\overline{\varphi} \in \mathrm{MATh}(\mathcal{G})$ and let $m, n \in \mathbb{N}$. To show $(\underline{m} \times \underline{n}, E_{n,m}) \models \varphi$, we consider the grid $L = (\overline{(mn+1)}^2, \leq)$. First, we define the following sets:
$X := \{(i, mn - i) \mid 0 \leq i < mn\}$,
$E_1 := \{(i, mn - i - 1) \mid 0 \leq i < mn, i \mod n \neq n - 1\}$, and
$E_2 := \{(i, (m-1)n - i) \mid 0 \leq i < (m-1)n\}$. Note that these sets are antichains for increasing i increases the first and decreases the second component of any of their elements. Finally, let $e := (mn, 1) \in \mathbb{J}(L)$. Since $L \models \overline{\varphi}$, we obtain $L \models (\gamma' \to \varphi')(X, E_1, E_2, e)$. Next, we define a binary relation E on X by

$$(x, y) \in E :\iff \sup(x, y) \in E_1 \cup E_2 \text{ and } \inf(x, e) \leq \inf(y, e).$$

We show that (X, E) and $(\underline{m} \times \underline{n}, E_{m,n})$ are isomorphic: Define the bijection $f : \underline{m} \times \underline{n} \to X$ by $f(a_1, a_2) := (a_1 n + a_2, mn - a_1 n - a_2)$. The following sequence of equivalences establishes that f is a graph isomorphism:

$$((a_1, a_2), (b_1, b_2)) \in E_{m,n}$$

$$\iff a_1 + 1 = b_1 \text{ and } a_2 = b_2, \text{ or } \\ a_1 = b_1 \text{ and } a_2 + 1 = b_2$$

$$\iff a_1 \leq b_1 \text{ and either } \\ (a_1 n + a_2, mn - b_1 n - b_2) \in E_1, \text{ or } \\ (a_1 n + a_2, mn - b_1 n - b_2) \in E_2$$

$$\iff \inf(f(a_1, a_2), e) = a_1 \leq b_1 = \inf(f(b_1, b_2), e) \text{ and } \\ \sup(f(a_1, a_2), f(b_1, b_2)) \in E_1 \cup E_2$$

$$\iff (f(a_1, a_2), f(b_1, b_2)) \in E.$$

Hence $(X, E) \cong (\underline{m} \times \underline{n}, E_{m,n})$ implying $(X, E) \models \gamma$. By the construction of E and of γ', this implies $L \models_A \gamma'(X, E_1, E_2, e)$ and therefore $L \models_A \varphi'(X, E_1, E_2, e)$. By the same argument, $(X, E) \models \varphi$, i.e. $\varphi \in \mathrm{MTh}(\mathcal{GG})$. □

Since the monadic theory $\mathrm{MTh}(\mathcal{G})$ is undecidable, so is the monadic antichain theory $\mathrm{MATh}(\mathcal{G})$ by the lemma above. Note that $\mathcal{G} = \mathbb{H}_f(\mathfrak{P})$ where $\mathfrak{P}$ contains all finite partial orders that consist of two incomparable chains. These partial orders have unbounded diabolo width. Next we want to generalize this undecidability to any class $\mathbb{H}_f(\mathfrak{P})$ where the elements of $\mathfrak{P}$ have unbounded diabolo width.

Before doing so, we need some prerequisites: Let $(P, \leq)$ be a partially ordered set, $A \subseteq P$ an antichain and $a \in A$. On A, we define a binary relation R_a by $(x, y) \in R_a$ iff $x \vee a \leq y \vee a$. There is a monadic formula φ with free variables A and a such that $(P, \leq) \models_A \varphi(A, a)$ iff

1. (A, R_a) is a finite linear order, and

2. for any $x, y, x', y' \in A$ with $(x, y), (x', y') \in R_a$ we have

$$x \vee y \leq x' \vee y' \iff (x, x'), (y', y) \in R_a.$$

Assume A and a have these properties. Now let $b \in A$ and define

$$G(b, A) := \{x \vee y \mid x, y \in A, b \leq x \vee y\}.$$

We show that $(G(b, A), \leq)$ is a grid: For $(x, y) \in R_a$ by the second property $b \leq x \vee y$ iff $(x, b), (b, y) \in R_a$ since $b = b \vee b$ and $(b, b) \in R_a$. Let $a_1, a_2, \ldots, a_k$ be the enumeration of A according to the linear order R_a. Then $(a_i, a_j) \in R_a$ for $1 \leq i \leq j \leq k$. Thus, we showed that the mapping

$$\begin{aligned} f \quad : \quad & (\{a_1, a_2, \ldots, a_m\}, R_a) \times (\{a_m, a_{m+1}, \ldots, a_k\}, R_a^{-1}) \to (G(b, A), \leq) \\ & (a_i, a_j) \mapsto a_i \vee a_j \end{aligned}$$

is an order isomorphism where $b = a_m$. Hence the poset $(G(b, A), \leq)$ is isomorphic to the grid $(\underline{m} \times \underline{(k - m + 1)}, \leq)$, i.e. $G(b, A)$ is a grid.

On the other hand, let $\mathfrak{P}$ be a class of partially ordered sets such that the diabolo width of its elements is not bounded above. We show that for any $n \in \mathbb{N}$ there exists a poset $(P, \leq) \in \mathfrak{P}$, an antichain $A \subseteq \mathbb{H}_f(P, \leq)$, $a, b \in A$ such that $\mathbb{H}_f(P, \leq) \models_A \varphi(A, a)$ and $G(b, A) \cong (\underline{n} \times \underline{n}, \leq)$: By Lemma 10.2.5 $\mathfrak{P}$ contains a poset $(P, \leq)$ allowing a semilattice embedding $f : (\underline{(2n)}^2, \leq) \hookrightarrow \mathbb{H}_f(P, \leq)$. Let $A := \{f(i, 2n - i) \mid 0 \leq i < 2n\}$, $a := f(0, 2n)$ and $b := f(n, n)$. Then $a \vee f(i, 2n - i) \leq a \vee f(j, 2n - j)$ iff $i \leq j$ for $i, j \in \underline{2n}$ since f is a semilattice embedding. Hence (A, R_a) is a finite linearly ordered set as required by the first property stated by φ. Now let $i, j, i', j' \in \underline{2n}$ with $i \leq j$ and $i' \leq j'$. Then $f(j, 2n-i) = f(i, 2n-i) \vee f(j, 2n-j) \leq f(i', 2n-i') \vee f(j', 2n-j') = f(j', 2n-i')$ iff $i \leq i'$ and $j \geq j'$, i.e. the second property holds as well. Hence $\mathbb{H}_f(P, \leq)$ satisfies $\varphi(A, a)$. Since, for $i, j \in \underline{2n}$ it holds $f(i, 2n - i) \vee f(j, 2n - j) \geq b = f(n, n)$ iff $n \leq i$ and $n \geq j$, the mapping f is an isomorphism from $(\underline{(2n)}^2, \leq)$ to $G(b, A)$ as claimed above.

Lemma 11.2.12 *Let $\mathfrak{P}$ be a set of partially ordered sets such that the diabolo width of its members is not bounded above. Then the monadic antichain theory* $\mathrm{MATh}(\mathcal{G})$ *can be reduced to the monadic antichain theory* $\mathrm{MATh}(\mathbb{H}_f(\mathfrak{P}))$.

Proof. A monadic sentence ψ belongs to $\mathrm{MATh}(\mathcal{G})$ iff the monadic sentence $\forall A \forall a, b((\varphi(A, a) \wedge b \in A) \to \psi')$ belongs to $\mathrm{MTh}(\mathbb{H}_f(\mathfrak{P}))$ where ψ' is the restriction of ψ to $G(b, A)$. □

Thus, with $\mathfrak{P}$ as above, the undecidable monadic theory $\mathrm{MTh}(\mathcal{G})$ can be reduced to the monadic antichain theory $\mathrm{MATh}(\mathbb{H}_f(\mathfrak{P}))$. Hence this antichain theory is undecidable.

The chain theory

The following two lemmas imply a similar result for the monadic chain theory $\mathrm{MCTh}(\mathbb{H}_f(\mathfrak{P}))$. But differently from Lemma 11.2.11, the following lemma does not reduce the full monadic theory of $\mathcal{G}$ but the monadic antichain theory to its chain theory. The main ingredient of the proof is the following: Let $n, m > 1$ and $L := (\underline{m} \times \underline{n}, \leq)$. Then $e := (1, m)$ and $\overline{e} := (n, 1)$ are maximal join irreducible and incomparable elements of L. We define a partial order $\sqsubseteq$ on L by $x \sqsubseteq y$ iff $\inf(x, e) \leq \inf(y, e)$ and $\inf(x, \overline{e}) \geq \inf(y, \overline{e})$. For $x = (x_1, x_2)$ and $y = (y_1, y_2)$, it holds $\inf(x, e) = x_2$ and $\inf(x, \overline{e}) = x_1$. Hence $x \sqsubseteq y$ iff $x_2 \leq y_2$ and $x_1 \geq y_1$. In other words, $(\underline{m} \times \underline{n}, \sqsubseteq)$ equals $(\underline{m}, \geq) \times (\underline{n}, \leq)$ and is therefore isomorphic to L. Now let x and y be incomparable with respect to $\sqsubseteq$. Then $x_1 < y_1$ and $x_2 < y_2$ or vice versa. In particular $x \leq y$ or $y \leq x$. Hence antichains in $(\underline{m} \times \underline{n}, \sqsubseteq)$ are chains in $(\underline{m} \times \underline{n}, \leq)$ (the converse implication does not hold).

Lemma 11.2.13 *The monadic antichain theory* $\mathrm{MATh}(\mathcal{G})$ *can be reduced in linear time to the monadic chain theory* $\mathrm{MCTh}(\mathcal{G})$.

Proof. Let φ be a monadic formula not containing the variables e and $\overline{e}$. In φ, replace any atomic formula $x \leq y$ by

$$\inf(x, e) \leq \inf(y, e) \wedge \inf(x, \overline{e}) \geq \inf(y, \overline{e})$$

and replace any subformula of the form $\exists X \psi$ by $\exists X(\mathrm{antichain}_{\sqsubseteq}(X) \wedge \psi)$ where $\mathrm{antichain}_{\sqsubseteq}(X)$ denotes the formula

$$\forall x, y((x, y \in X \wedge \inf(x, e) \leq \inf(y, e)) \to \inf(x, \overline{e}) \leq \inf(y, \overline{e})).$$

The subformula $\mathrm{antichain}_{\sqsubseteq}(X)$ is satisfied by a set X iff its elements are mutually incomparable with respect to $\sqsubseteq$. Denote the result of these replacements by φ'.

Let $m, n \geq 1$, $e = (1, m)$ and $\overline{e} = (n, 1)$. As a prerequisite, we show that $(\underline{m} \times \underline{n}, \sqsubseteq) \models_A \varphi$ iff $(\underline{m} \times \underline{n}, \leq) \models_C \varphi'$ by induction: Clearly, $(\underline{m} \times \underline{n}, \sqsubseteq) \models_A x \leq y$ iff $x \sqsubseteq y$ iff $(\underline{m} \times \underline{n}, \leq) \models_C \inf(x, e) \leq \inf(y, e) \wedge \inf(x, \overline{e}) \geq \inf(y, \overline{e})$ which equals $(x \leq y)'$. Now let φ_i $(i = 1, 2)$ be monadic formulas such that for any antichains X_i w.r.t. $\sqsubseteq$ and elements x_j:

$$\begin{gathered}(\underline{m} \times \underline{n}, \sqsubseteq) \models_A \varphi_i(X_1, \ldots, X_k, x_1, \ldots, x_\ell) \\ \Longleftrightarrow \\ (\underline{m} \times \underline{n}, \leq) \models_C \varphi_i'(X_1, \ldots, X_k, x_1, \ldots, x_\ell).\end{gathered}$$

It is straightforward to check that this equivalence holds for $\neg\varphi_1$, $\varphi_1 \wedge \varphi_2$ and for $\exists x_\ell \varphi_1$, too. The only nontrivial case in the induction is the formula $\varphi = \exists X_k \varphi_1$: So let $X_i \subseteq \underline{m} \times \underline{n}$ be antichains w.r.t. $\sqsubseteq$ for $1 \leq i < k$ and let $x_j \in \underline{m} \times \underline{n}$. Then $(\underline{m} \times \underline{n}, \sqsubseteq) \models_A \varphi(X_1, \ldots, X_{k-1}, x_1, \ldots, x_\ell)$ iff there exists an antichain X_k

w.r.t. $\sqsubseteq$ such that $(\underline{m} \times \underline{n}, \sqsubseteq) \models_A \varphi_1(X_1, \ldots, X_k, x_1, \ldots, x_\ell)$. By the induction hypothesis, this is equivalent to $(\underline{m}\times\underline{n}, \leq) \models_C \varphi_1'(X_1, \ldots, X_k, x_1, \ldots, x_\ell)$. By the remarks preceding this lemma, X_k is a chain w.r.t. $\leq$. Hence the last statement is equivalent to $(\underline{m}\times\underline{n}, \leq) \models_A (\exists X_k(\mathrm{antichain}_\sqsubseteq(X_k)\wedge\varphi_1'))(X_1, \ldots, X_{k-1}, x_1, \ldots, x_\ell)$ which equals φ'.

Now it is straightforward to show that for a monadic sentence φ it holds $(\underline{m} \times \underline{n}, \leq) \models_A \varphi$ iff $(\underline{m} \times \underline{n}, \leq) \models_C \exists e, \overline{e}(e, \overline{e} \in \max(\mathbb{J}) \wedge e \neq \overline{e} \wedge \varphi')$ which is the desired reduction. □

Lemma 11.2.14 *Let $\mathfrak{P}$ be a set of partially ordered sets such that the diabolo width of its members is not bounded above. Then the monadic chain theory* MCTh($\mathcal{G}$) *can be reduced to the monadic chain theory* MCTh($\mathbb{H}_f(\mathfrak{P})$).

Proof. By Lemma 10.2.5, there is $(P, \leq) \in \mathfrak{P}$ such that $\mathbb{H}_f(P, \leq)$ contains a subposet that is isomorphic to the square grid of dimension (n, n). Now we describe the reduction: Let φ be a monadic sentence. Then $\overline{\varphi}$ is the sentence
$\forall C_1, C_2 \forall x_1, y_1, x_2, y_2 :$
$([C_1, C_2$ are finite and incomparable chains $\wedge$
$(x_1, y_1 \in C_1 \wedge x_2, y_2 \in C_2 \rightarrow (x_1 \vee x_2 \leq y_1 \vee y_2 \leftrightarrow (x_1 \leq y_1 \wedge x_2 \leq y_2)))]$
$\rightarrow \varphi')$
where φ' is the restriction of φ to the set $\{x \vee y \mid x \in C_1, y \in C_2\}$.

Suppose $\overline{\varphi} \in$ MCTh($\mathbb{H}_f(\mathfrak{P})$) and let $m, n \in \mathbb{N}$ and let $L = (\underline{m} \times \underline{n}, \leq)$. We show $L \models_C \varphi$: By the consideration above, there exists $(P, \leq) \in \mathfrak{P}$ such that $\{I(i, j) \mid 1 \leq i \leq m, 1 \leq j \leq n\}$ is isomorphic to L. We consider the two finite chains $C_1 := \{I(i, 0) \mid 1 \leq i \leq m\}$ and $C_2 := \{I(0, j) \mid 1 \leq j \leq n\}$. Then one can easily check that the properties stated by $\overline{\varphi}$ hold. Hence $L \models_C \varphi$.

Conversely, let $\varphi \in$ MCTh($\mathcal{G}$). Let $(P, \leq) \in \mathfrak{P}$, $C_1, C_2 \subseteq \mathbb{H}_f(P, \leq)$ be finite incomparable chains such that for any $x_i, y_i \in C_i$ it holds $x_1 \vee x_2 \leq y_1 \vee y_2$ iff $x_1 \leq y_1$ and $x_2 \leq y_2$. Then the set of suprema $\{x \vee y \mid x \in C_1, y \in C_2\}$ is isomorphic to the grid $(\underline{|C_1|} \times \underline{|C_2|}, \leq)$. Since $(\underline{|C_1|} \times \underline{|C_2|}, \leq) \models_C \varphi$, we obtain $\{x \vee y \mid x \in C_1, y \in C_2\} \models_C \varphi$ and therefore $\mathbb{H}_f(P, \leq) \models_C \overline{\varphi}$. □

Now we can prove a result on the monadic theories analogously to Corollary 11.2.3:

Theorem 11.2.15 *Let $\mathfrak{P}$ be a set of partially ordered sets.*

1. *The following are equivalent:*
   - *(i) The monadic theory* $\mathrm{MTh}(\mathbb{H}_f(\mathfrak{P}))$ *is decidable.*
   - *(ii) The monadic chain theory* $\mathrm{MCTh}(\mathbb{H}_f(\mathfrak{P}))$ *is decidable.*
   - *(iii) the monadic theory* $\mathrm{MTh}(\mathfrak{P})$ *is decidable and the diabolo width of the elements of* $\mathfrak{P}$ *is bounded above.*
   - *(iv) the monadic chain theory* $\mathrm{MCTh}(\mathfrak{P})$ *is decidable and the diabolo width of the elements of* $\mathfrak{P}$ *is bounded above.*
2. *The monadic antichain theory* $\mathrm{MATh}(\mathbb{H}_f(\mathfrak{P}))$ *is decidable if and only if the elementary theory* $\mathrm{Th}(\mathfrak{P})$ *is decidable and the diabolo width of the elements of* $\mathfrak{P}$ *is bounded above.*

Proof. The implication (i)⇒(ii) is trivial. Now assume $\mathrm{MCTh}(\mathbb{H}_f(\mathfrak{P}))$ to be decidable. Next we show the implication (ii)⇒(iv): The monadic chain theory $\mathrm{MCTh}(\mathfrak{P})$ is decidable by Remark 11.2.1. By contradiction, assume that the diabolo width of the elements of $\mathfrak{P}$ is unbounded. By Lemmas 11.2.11, 11.2.13, and 11.2.14, the monadic theory of the grids $\mathrm{MTh}(\mathcal{G})$ can be reduced to the monadic chain theory of $\mathbb{H}_f(\mathfrak{P})$, contradicting the decidability of this latter theory.

For the implication $(iv) \Rightarrow (iii)$ note that the width of the elements of $\mathfrak{P}$ is bounded by n, say. Hence any subset of P with $(P, \leq) \in \mathfrak{P}$ is the union n chains. Therefore, the monadic theory of $\mathfrak{P}$ can be reduced to the monadic chain theory of $\mathfrak{P}$. Hence (iii) holds. The last implication $(iii) \Rightarrow (i)$ follows from Theorem 11.2.10.

It remains to show the second statement: The decidability of $\mathrm{MATh}(\mathbb{H}_f(\mathfrak{P}))$ trivially implies that of $\mathrm{Th}(\mathbb{H}_f(\mathfrak{P}))$ and therefore that of $\mathrm{Th}(\mathfrak{P})$. The diabolo with of the elements of $\mathfrak{P}$ is bounded above by Lemmas 11.2.11 and 11.2.12. Thus, we showed one implication.

Conversely, let the diabolo width of the elements of $\mathfrak{P}$ be bounded by n and let $\mathrm{Th}(\mathfrak{P})$ be decidable. Since then the width of the elements of $\mathfrak{P}$ is uniformly bounded by $2n$, the elementary theory $\mathrm{Th}(\mathbb{H}_f(\mathfrak{P}))$ is decidable by Corollary 11.2.3. In addition, the width of the elements of $\mathbb{H}_f(\mathfrak{P})$ is bounded above by some $m \in \mathbb{N}$. Hence any antichain contains at most m elements implying that $\mathrm{MATh}(\mathbb{H}_f(\mathfrak{P}))$ can be reduced to $\mathrm{Th}(\mathbb{H}_f(\mathfrak{P}))$. Hence the monadic antichain theory $\mathrm{MATh}(\mathbb{H}_f(\mathfrak{P}))$ is decidable. □

11.3 Finite distributive lattices

Since, for any finite distributive lattice $(L, \leq)$ it holds $(L, \leq) \cong \mathbb{H}_f \, \mathbb{J}(L, \leq)$, we can now characterize the sets of finite distributive lattices having a decidable monadic (chain, antichain) theory:

Corollary 11.3.1 *Let $\mathfrak{L}$ be a set of finite distributive lattices.*

1. *The following are equivalent:*

   (i) The monadic theory MTh$(\mathfrak{L})$ *is decidable.*

   (ii) The monadic chain theory MCTh$(\mathfrak{L})$ *is decidable.*

   (iii) the monadic theory MTh$(\mathbb{J}(\mathfrak{L}))$ *is decidable and the width of the elements of $\mathfrak{L}$ is bounded above.*

   (iv) the monadic chain theory MCTh$(\mathbb{J}(\mathfrak{L}))$ *is decidable and the width of the elements of $\mathfrak{L}$ is bounded above.*

2. *The monadic antichain theory* MATh$(\mathfrak{L})$ *is decidable if and only if the elementary theory* Th$(\mathbb{J}(\mathfrak{L}))$ *is decidable and the width of the elements of $\mathfrak{L}$ is bounded above.*

Proof. Since $\mathbb{H}_f(\mathbb{J}(\mathfrak{L})) = \mathfrak{L}$, it remains to show that the width of the elements of $\mathfrak{L}$ is bounded if and only if the diabolo width of the elements of $\mathbb{J}(\mathfrak{L})$ is bounded. In the proof of Lemma 11.2.14 we saw that a bounded width of the elements of $\mathfrak{L}$ implies a bound of the diabolo width of the elements of $\mathbb{J}(\mathfrak{L})$.

To show the other implication assume $dw(\mathbb{J}(L, \leq)) < n - 2$ for any lattice $(L, \leq) \in \mathfrak{L}$. By contradiction, suppose that the width of the elements of $\mathfrak{L}$ is unbounded. Then there exists $(L, \leq)$ in $\mathfrak{L}$ such that $w(L, \leq) \geq R_{n+1}(6^n)$. By Theorem 10.2.6, there exists a lattice embedding $\eta : [n-1] \times [n-1] \to L$. Let $A := \mathbb{J}(L) \cap {\downarrow}\eta(1, n-1)$ and $B := \mathbb{J}(L) \cap {\downarrow}\eta(n-1, 1)$. Since the elements $\eta(1, i)$ and $\eta(j, 1)$ are pairwise incomparable for $i, j > 1$, $A \setminus B$ and $B \setminus A$ both contain at least $n - 2$ elements. Furthermore, these two sets are incomparable. Hence the diabolo width of $\mathbb{J}(L, \leq)$ is at least $n - 2$, a contradiction. $\square$

Now let $\mathfrak{L}_1$ and $\mathfrak{L}_2$ be sets of finite distributive lattices. Suppose that the elementary theories of $\mathfrak{L}_1$ and $\mathfrak{L}_2$ are decidable. Then, as an easy consequence of the Feferman-Vaught Theorem, the set of direct products of lattices from $\mathfrak{L}_1$ and lattices from $\mathfrak{L}_2$ has a decidable elementary theory. Next, we want to characterize when this set has a decidable monadic (chain, antichain) theory:

Corollary 11.3.2 *Let* $\mathfrak{L}_1$ *and* $\mathfrak{L}_2$ *be sets of finite distributive lattices and define* $\mathfrak{L} := \{(L_1, \leq) \times (L_2, \leq) \mid (L_i, \leq) \in \mathfrak{L}_i\}$. *If* $\mathrm{MTh}(\mathfrak{L}_i)$ *(*$\mathrm{MATh}(\mathfrak{L}_i)$, $\mathrm{MCTh}(\mathfrak{L}_i)$*, resp.) is decidable, then* $\mathrm{MTh}(\mathfrak{L})$ *(*$\mathrm{MATh}(\mathfrak{L})$, $\mathrm{MCTh}(\mathfrak{L})$*, resp.) is decidable iff* $\mathfrak{L}_1$ *or* $\mathfrak{L}_2$ *is finite.*

Proof. We give the proof for the monadic theories, only. The other cases can be handled similarly. If both $\mathfrak{L}_1$ and $\mathfrak{L}_2$ are infinite, we find for any $n \in \mathbb{N}$ lattices $(L_1, \leq) \in \mathfrak{L}_1$ and $(L_2, \leq) \in \mathfrak{L}_2$ of length at least n. Then the width of the direct product $(L_1, \leq) \times (L_2, \leq)$ is at least n, i.e. the width of the lattices in $\mathfrak{L}$ is not bounded. Hence $\mathrm{MTh}(\mathfrak{L})$ is undecidable.

Conversely let $\mathfrak{L}_1$ be finite. Then there is $n \in \mathbb{N}$ with $|L_1| \leq n$ for any lattice $(L_1, \leq) \in \mathfrak{L}_1$. Since $\mathrm{MTh}(\mathfrak{L}_2)$ is decidable, we can assume $w(\mathfrak{L}_2) \leq n$. Note that the width $w(L_1 \times L_2, \leq)$ is at most $|L_1| \cdot w(L_2, \leq)$ for any finite distributive lattices $(L_1, \leq)$ and $(L_2, \leq)$. Hence $w(\mathfrak{L}) \leq n^2$. It remains to show that $\mathbb{J}(\mathfrak{L})$ has a decidable monadic theory: For finite distributive lattices $(L_1, \leq)$ and $(L_2, \leq)$, one has $\mathbb{J}(L_1 \times L_2, \leq) = \mathbb{J}(L_1, \leq) \dot\cup \mathbb{J}(L_2, \leq)$. Thus, we have to show that the monadic theory of $\{(P_1, \leq) \dot\cup (P_2, \leq) \mid (P_i, \leq) \in \mathbb{J}(\mathfrak{L}_i)\}$ is decidable. This follows from the composition theorem from Shelah [She75] (cf. [Tho97a] for the proof of this result) since $\mathrm{MTh}(\mathbb{J}(\mathfrak{L}_i))$ is decidable. □

Note that the in the corollary above we assumed from the very beginning that $\mathrm{MTh}(\mathfrak{L}_i)$ is decidable for $i = 1, 2$. Actually, the finiteness of $\mathfrak{L}_1$ or $\mathfrak{L}_2$ follows without this assumption from the decidability of $\mathrm{MTh}(\mathfrak{L})$. We finish this section with an example of classes $\mathfrak{L}_1$, $\mathfrak{L}_2$ and $\mathfrak{L}$ as in the corollary above such that $\mathfrak{L}_1$ is finite, $\mathfrak{L}$ has a decidable monadic theory but the monadic theory of $\mathfrak{L}_2$ is undecidable:

Example 11.3.3 For simplicity, let $\underline{2}$ denote the Boolean lattice $(\{1, 2\}, \leq)$. Let $\mathfrak{L}_1$ consist of the lattices $\underline{2}^i$ for $0 \leq i \leq 2$ (i.e. $\mathfrak{L}_1$ contains the one-point-lattice, the Boolean lattice and the diamond). Let Lin denote the set of finite linear orders and let $\mathfrak{P} \subseteq \mathrm{Lin}$ be an undecidable set of linear orders. We define a set of finite distributive lattices $\mathfrak{L}_2 \subseteq \{\underline{2}^i \times (L, \leq) \mid 0 \leq i \leq 2, (L, \leq) \in \mathrm{Lin}\}$ by $\underline{2}^i \times (L, \leq) \in \mathfrak{L}_2$ iff

1. $i \in \{0, 2\}$ and $(L, \leq) \in \mathrm{Lin}$, or
2. $i = 1$ and $(L, \leq) \in \mathfrak{P}$.

Then $\mathfrak{L}_2$ is a set of finite distributive lattices. It is undecidable since the subset of $\mathfrak{L}_2$ of lattices of width 2 corresponds to $\mathfrak{P}$ which was chosen to be undecidable. Hence, in particular, $\mathfrak{L}_2$ has an undecidable monadic theory. It is straightforward to show that the set of direct products of lattices from $\mathfrak{L}_1$ and $\mathfrak{L}_2$ equals the set $\{\underline{2}^i \times (L, \leq) \mid 1 \leq i \leq 5, (L, \leq) \in \mathrm{Lin}\} = \mathfrak{L}$. Since Lin has a decidable monadic theory, we can apply the corollary above and obtain that $\mathrm{MTh}(\mathfrak{L})$ is decidable.

Remark The important property that we used in this section is the isomorphism of $(L, \leq)$ and $\mathbb{H}_f\,\mathbb{J}(L, \leq)$ whenever $(L, \leq)$ is a finite distributive lattice. The reader may check that Corollary 11.3.1 holds verbatim if we require $\mathfrak{L}$ to consist of distributive lattices satisfying this isomorphism. In Corollary 11.3.2, we obtained that $\mathrm{MTh}(\mathfrak{L})$ is decidable if and only if $\mathfrak{L}_1$ or $\mathfrak{L}_2$ is a finite set of *finite* lattices.

Main theorems

Theorem 3.3.4 There exists an algorithm that solves the following decision problem:
input: 1. an alphabet Σ,
2. a basis algorithm of an effective and monotone Σ-ACM $\mathcal{A}'$,
3. the set of final states F' of $\mathcal{A}'$,
4. a finite basis of $(Q_c, \sqsubseteq_c)$, and an algorithm to decide $\sqsubseteq_c$ for $c \in \Sigma$.

output: Is $L(\mathcal{A}')$ empty?

Corollary 3.3.5 Let $\mathcal{A}$ be a monotone and effective Σ-ACM. Then the set $L(\mathcal{A})$ is recursive.

Theorem 4.1.7 Let Σ be an alphabet with at least two letters. Then there is no algorithm that on input of a Σ-ACA $\mathcal{A}$ decides whether it accepts all Σ-dags, i.e. whether $L(\mathcal{A}) = \mathbb{D}$.

Corollary 4.1.8 Let Σ be an alphabet with at least two letters. Then the equivalence of Σ-ACAs, i.e. the question whether $L(\mathcal{A}_1) = L(\mathcal{A}_2)$, is undecidable.

Theorem 4.1.10 Let Σ be an alphabet with at least two letters. Then there is no algorithm that on input of a Σ-ACA $\mathcal{A}$ decides any of the following questions:

1. Is $\mathbb{D} \setminus L(\mathcal{A})$ recognizable?
2. Is $\mathcal{A}$ equivalent with some deterministic Σ-ACA?

Theorem 5.1.1 Let $\mathcal{A}$ be a possibly nondeterministic Σ-ACA. There exists a monadic sentence φ over Σ such that $L(\mathcal{A}) = \{t \in \mathbb{D} \mid t \models \varphi\}$.

Theorem 5.2.9 Let φ be a monadic sentence and let $k \in \mathbb{N}$. Then there exists a Σ-ACA $\mathcal{A}$ such that $L(\mathcal{A}) = \{t \in \mathbb{D}_k \mid t \models \varphi\}$.

Theorem 6.1.7 Let $\mathcal{B}$ be a branching automaton. Then there exists a Σ-ACA $\mathcal{A}$ such that $\mathrm{Ha}(L(\mathcal{B}) \cap \mathrm{SP}_{wa}(\Sigma)) = L(\mathcal{A})$.

Theorem 6.1.9 Let $\mathcal{A}$ be a Σ-ACA. Then there exists a branching automaton $\overline{\mathcal{B}}$ such that $\mathrm{Ha}(L(\overline{\mathcal{B}})) = L(\mathcal{A}) \cap \mathrm{Ha}(\mathrm{SP}(\Sigma))$.

Theorem 6.1.10 Let $L \subseteq \mathrm{SP}(\Sigma)$ be a width-bounded sp-language. Then L can be accepted by a branching automaton iff it is monadically axiomatizable.

Theorem 6.2.1 Let $\mathcal{B}$ be a P-asynchronous automaton over Σ. Then there exists a Σ-ACA $\mathcal{A}$ with $\mathrm{Ha}(L(\mathcal{B})) = L(\mathcal{A})$.

Theorem 8.2.10 Let T be a finite set and E a set of equations of the form $ab = cd$ with $a, b, c, d \in T$. Let $\sim$ be the least congruence on $T^\star$ containing E. Then $M := T^\star/\sim$ is a divisibility monoid if and only if (i)-(iii) hold for any $a, b, c, b', c' \in T$:
(i) $(\downarrow(a \cdot b \cdot c), \leq)$ is a distributive lattice,
(ii) $a \cdot b \cdot c = a \cdot b' \cdot c'$ or $b \cdot c \cdot a = b' \cdot c' \cdot a$ implies $b \cdot c = b' \cdot c'$, and
(iii) $a \cdot b = a' \cdot b'$, $a \cdot c = a' \cdot c'$ and $a \neq a'$ imply $b = c$.
Furthermore, each divisibility monoid arises this way.

Theorem 9.1.8 Let $(M, \cdot, 1)$ be a divisibility monoid with finite commutation behavior. Let $X \subseteq T^\star$ be recognizable and $n := \mathrm{rk}(X)$ be finite. Then $\mathrm{nat}(X)$ is recognizable in M.

Theorem 9.2.8 Let $(M, \cdot, 1)$ be a divisibility monoid with finite commutation behavior. Let $L \subseteq M$ be c-rational. Then L is recognizable.

Theorem 9.3.8 Let $(M, \cdot, 1, \rho)$ be a labeled divisibility monoid with finite commutation behavior. Let $L \subseteq M$. Then the following are equivalent:
1. L is recognizable
2. L is c-rational
3. L is mc-rational.

Theorem 10.2.8 Let $(M, \cdot, 1)$ be a divisibility monoid with finite commutation behavior. Then the following are equivalent
1. M is width-bounded,
2. M is rational, and
3. any set $L \subseteq M$ is rational iff it is recognizable.

Theorem 11.1.3 Let $(M, \cdot, 1)$ be a divisibility monoid with finite commutation behavior. Then the monadic theory $\mathrm{MTh}(\{(\mathbb{J}(\downarrow m), <) \mid m \in M\})$ is decidable.

Theorem 11.1.4 Let (Σ, D) be a finite dependence alphabet. Then the monadic theory of $(\mathbb{J}(\mathbb{M}(\Sigma, D)), \leq)$ is decidable iff D is transitive.

Theorem 11.1.9 Let $(M, \cdot, 1)$ be a divisibility monoid with finite commutation behavior. Then the monadic theory $\mathrm{MTh}\{(\downarrow m, \leq) \mid m \in M\}$ is decidable iff M is width-bounded.

Theorem 11.2.2 Let $\mathfrak{P}$ be a set of partially ordered sets of uniformly bounded width. Then $\mathrm{Th}(\mathbb{H}_f(\mathfrak{P}))$ can be reduced to $\mathrm{Th}(\mathfrak{P})$ in linear time.

Theorem 11.2.10 Let $\mathfrak{P}$ be a set of partially ordered sets whose diabolo width is uniformly bounded. Then $\mathrm{MTh}(\mathbb{H}_f(\mathfrak{P}))$ can be reduced to $\mathrm{MTh}(\mathfrak{P})$ in linear time.

Corollary 11.3.1 Let $\mathfrak{L}$ be a set of finite distributive lattices.

1. The following are equivalent:

   (i) The monadic theory $\mathrm{MTh}(\mathfrak{L})$ is decidable.

   (ii) The monadic chain theory $\mathrm{MCTh}(\mathfrak{L})$ is decidable.

   (iii) the monadic theory $\mathrm{MTh}(\mathbb{J}(\mathfrak{L}))$ is decidable and the width of the elements of $\mathfrak{L}$ is bounded above.

   (iv) the monadic chain theory $\mathrm{MCTh}(\mathbb{J}(\mathfrak{L}))$ is decidable and the width of the elements of $\mathfrak{L}$ is bounded above.

2. The monadic antichain theory $\mathrm{MATh}(\mathfrak{L})$ is decidable if and only if the elementary theory $\mathrm{Th}(\mathbb{J}(\mathfrak{L}))$ is decidable and the width of the elements of $\mathfrak{L}$ is bounded above.

Open problems

We list some questions that are left open in the present work. For more details see the page indicated.

- Is the emptiness of $L(\mathcal{A})$ for nonmonotone but effective asynchronous cellular machines decidable (page 43)? Furthermore, we did not consider the complexity of the emptiness problem for asynchronous cellular machines or automata.

- Is it decidable whether an asynchronous cellular machine accepts some Hasse-diagram (page 43)?

- For which sets of Σ-dags L is the set of Σ-ACAs $\mathcal{A}$ with $L(\mathcal{A}) = L$ recursive (page 59)?

- Let $k \in \mathbb{N}$. Is the set of Σ-ACAs $\mathcal{A}$ satisfying $L(\mathcal{A}) \cap \mathbb{D}_k = L(\mathcal{A}_d) \cap \mathbb{D}_k$ for some deterministic Σ-ACA $\mathcal{A}_d$ recursive (page 71)?

- Is there an extension of the monadic second order logic that allows one to axiomatize precisely the rational sp-languages (page 75)?

- Does there exist a divisibility monoid with infinite commutation behavior (page 95)? If this is the case, is the property to have finite commutation behavior decidable on input of a presentation as in Theorem 8.2.10?

- Is it possible to find finitely many sets C_q in a divisibility monoid such that rational sets where the iteration is applied to subsets of C_q only are recognizable (page 122)?

- We showed that any rational divisibility monoid with finite commutation behavior is width-bounded. Is this implication valid without the assumption "finite commutation behavior" (if there exists a divisibility monoid with infinite commutation behavior at all, page 139)?

- Is the property to be width-bounded (i.e. to satisfy Kleene's Theorem) decidable on input of a presentation as in Theorem 8.2.10?

Bibliography

[AK79] A.V. Anisimov and D.E. Knuth. Inhomogeneous sorting. *International Journal of Computer and Information Sciences*, 8:255–260, 1979.

[Arn91] A. Arnold. An extension of the notions of traces and of asynchronous automata. *Informatique Théorique et Applications*, 25:355–393, 1991.

[BCS93] P. Boldi, F. Cardone, and N. Sabadini. Concurrent automata, prime event structures and universal domains. In M. Droste and Y. Gurevich, editors, *Semantics of Programming Languages and Model Theory*, pages 89–108. Gordon and Breach Science Publ., OPA Amsterdam, 1993.

[BDK95] F. Bracho, M. Droste, and D. Kuske. Dependence orders for computations of concurrent automata. In *STACS'95*, Lecture Notes in Comp. Science vol. 900, pages 467–478. Springer, 1995.

[BDK97] F. Bracho, M. Droste, and D. Kuske. Representation of computations in concurrent automata by dependence orders. *Theoretical Comp. Science*, 174:67–96, 1997.

[BE97] O. Burkart and J. Esparza. More infinite results. *Bulletin of the EATCS*, 62:138–159, 1997. Columns: Concurrency.

[Ber66] R. Berger. *The undecidability of the domino problem*. Mem. Am. Math. Soc. vol. 66. AMS, 1966.

[Ber78] G. Berry. Stable models of typed λ-calculi. In *5th ICALP*, Lecture Notes in Comp. Science vol. 62, pages 72–89. Springer, 1978.

[Ber79] J. Berstel. *Transductions and context-free languages*. Teubner Studienbücher, Stuttgart, 1979.

[Bir73] G. Birkhoff. *Lattice Theory*. Colloquium Publications vol. 25. American Mathematical Society, Providence, 1973.

[Bod98] Hans L. Bodländer. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Comp. Science*, 209:1–45, 1998.

[BR94] I. Biermann and B. Rozoy. Context traces and transition systems. In *ISCIS IX, Antalya, Türkei*, 1994.

[BR95] I. Biermann and B. Rozoy. Graphs for generalized traces. In *STRICT95*, pages 101–115. Berlin, 1995.

[Büc60] J.R. Büchi. On a decision method in restricted second order arithmetics. In E. Nagel et al., editor, *Proc. Intern. Congress on Logic, Methodology and Philosophy of Science*, pages 1–11. Stanford University Press, Stanford, 1960.

[Cam94] P.J. Cameron. *Combinatorics*. Cambridge Univ. Press, 1994.

[CF69] P. Cartier and D. Foata. *Problemes combinatoires de commutation et rearrangements*. Lecture Notes in Mathematics vol. 85. Springer, Berlin - Heidelberg - New York, 1969.

[CLR95] M. Clerbout, M. Latteux, and Y. Roos. Semi-Commutations. In *[DR95]*, pages 487–552. 1995.

[CM88] P. Cori and Y. Métivier. Approximation of a trace, asynchronous automata and the ordering of events in distributed systems. In *ICALP'88*, Lecture Notes in Comp. Science vol. 317, pages 147–161. Springer, 1988.

[CMZ93] R. Cori, Y. Métivier, and W. Zielonka. Asynchronous mappings and asynchronous cellular automata. *Information and Computation*, 106:159–202, 1993.

[Cou90] B. Courcelle. The monadic second-order logic of graphs. I: Recognizable sets of finite graphs. *Information and Computation*, 85:12–75, 1990.

[CP85] P. Cori and D. Perrin. Automates et commutations partielles. *R.A.I.R.O. - Informatique Théorique et Applications*, 19:21–32, 1985.

[DG96] M. Droste and P. Gastin. Asynchronous cellular automata for pomsets without autoconcurrency. In *CONCUR'96*, Lecture Notes in Comp. Science vol. 1119, pages 627–638. Springer, 1996.

[DG98] V. Diekert and P. Gastin. Approximating traces. *Acta Inf.*, 35:567–593, 1998.

[DGK00] M. Droste, P. Gastin, and D. Kuske. Asynchronous cellular automata for pomsets. *Theoretical Comp. Science*, 247:1–38, 2000. (Fundamental study).

[Die90] V. Diekert. *Combinatorics on Traces*. Lecture Notes in Comp. Science vol. 454. Springer, 1990.

[Die91] V. Diekert. On the concatenation of infinite traces. In C. Choffrut et al., editor, *8th STACS*, Lecture Notes in Comp. Science vol. 480, pages 105–117. Springer, 1991.

[Die93] V. Diekert. On the concatenation of infinite traces. *Theoretical Comp. Science*, 113:35–54, 1993.

[Die96] R. Diestel. *Graphentheorie*. Springer, 1996.

[Dil50] R.P. Dilworth. A decomposition theorem for partially ordered sets. *Annals of Mathematics*, 51:161–166, 1950.

[DK96] M. Droste and D. Kuske. Logical definability of recognizable and aperiodic languages in concurrency monoids. In *Computer Science Logic*, Lecture Notes in Comp. Science vol. 1092, pages 467–478. Springer, 1996.

[DK98] M. Droste and D. Kuske. Recognizable and logically definable languages of infinite computations in concurrent automata. *International Journal of Foundations of Computer Science*, 9:295–313, 1998.

[DK99] M. Droste and D. Kuske. On recognizable languages in divisibility monoids. In G. Ciobanu and Gh. Paun, editors, *FCT99*, Lecture Notes in Comp. Science vol. 1684, pages 246–257. Springer, 1999.

[DK01] M. Droste and D. Kuske. Recognizable languages in divisibility monoids. *Mathematical Structures in Computer Science*, 11:741–770, 2001.

[DM95] V. Diekert and A. Muscholl. Construction of asynchronous automata. In *[DR95]*, pages 249–267. 1995.

[DM97] V. Diekert and Y. Métivier. Partial commutation and traces. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages Volume 3*, pages 457–533. Springer, 1997.

[DR95] V. Diekert and G. Rozenberg. *The Book of Traces*. World Scientific Publ. Co., 1995.

[Dro90] M. Droste. Concurrency, automata and domains. In *17th ICALP*, Lecture Notes in Comp. Science vol. 443, pages 195–208. Springer, 1990.

[Dro92] M. Droste. Concurrent automata and domains. *Intern. J. of Found. of Comp. Science*, 3:389–418, 1992.

[Dro94] M. Droste. A Kleene theorem for recognizable languages over concurrency monoids. In *21th ICALP*, Lecture Notes in Comp. Science vol. 820, pages 388–398. Springer, 1994.

[Dro95] M. Droste. Recognizable languages in concurrency monoids. *Theoretical Comp. Science*, 150:77–109, 1995.

[Dro96] M. Droste. Aperiodic languages in concurrency monoids. *Information and Computation*, 126:105–113, 1996.

[DS93] M. Droste and R.M. Shortt. Petri nets and automata with concurrency relations - an adjunction. In M. Droste and Y. Gurevich, editors, *Semantics of Programming Languages and Model Theory*, pages 69–87. Gordon and Breach Science Publ., OPA, Amsterdam, 1993.

[Dub86] C. Duboc. Commutations dans les monoïdes libres: un cadre théorique pour l'étude du parallelisme. Thèse, Faculté des Sciences de l'Université de Rouen, 1986.

[Ebi94] W. Ebinger. *Charakterisierung von Sprachklassen unendlicher Spuren durch Logiken.* PhD thesis, Universität Stuttgart, 1994.

[Elg61] C.C. Elgot. Decision problems of finite automata design and related arithmetics. *Trans. Am. Math. Soc.*, 98:21–51, 1961.

[EM65] C.C. Elgot and G. Mezei. On relations defined by generalized finite automata. *IBM J. Res. Develop.*, 9:47–65, 1965.

[EM93] W. Ebinger and A. Muscholl. Logical definability on infinite traces. In *20th ICALP*, Lecture Notes in Comp. Science vol. 700, pages 335–346. Springer, 1993.

[EM96] W. Ebinger and A. Muscholl. Logical definability on infinite traces. *Theoretical Comp. Science*, 154:67–84, 1996.

[FGL90] E. Fried, G. Grätzer, and H. Lakser. Projective geometries as cover-preserving sublattices. *Algebra Universalis*, 27:270–278, 1990.

[FS98] A. Finkel and Ph. Schnoebelen. Fundamental structures in well-structured infinite transition systems. In *LATIN'98*, Lecture Notes in Comp. Science vol. 1380, pages 102–118. Springer, 1998. Extended Abstract of [FS01].

[FS01] A. Finkel and Ph. Schnoebelen. Well-structured transition systems everywhere! *Theoretical Comp. Science*, 256:63–92, 2001.

[GP92] P. Gastin and A. Petit. Poset properties of complex traces. In *17th MFCS*, Lecture Notes in Computer Science vol. 629, pages 255–263. Springer, 1992.

[GR93] P. Gastin and B. Rozoy. The poset of infinitary traces. *Theoretical Comp. Science*, 125:167–204, 1993.

[Has91] K. Hashigushi. Recognizable closures and submonoids of free partially commutative monoids. *Theoretical Comp. Science*, 86:233–241, 1991.

[Hig52] G. Higman. Ordering by divisibility in abstract algebras. *Proc. London Math. Soc.*, 2:326–336, 1952.

[HKT92] P. W. Hoogers, H. C. M. Kleijn, and P. S. Thiagarajan. A trace semantics for Petri nets. In W. Kuich, editor, *ICALP'92*, Lecture Notes in Comp. Science vol. 623, pages 595–604. Springer, 1992.

[Hoo94] P.W. Hoogers. *Behavioural aspects of Petri nets.* PhD thesis, Rijksuniversiteit te Leiden, 1994.

[KK00] R. Kummetz and D. Kuske. The topology of Mazurkiewicz traces. Technical Report MATH-AL-11-2000, TU Dresden, 2000. Accepted for Theoretical Computer Science.

[Kle56] S.C. Kleene. Representation of events in nerve nets and finite automata. In C.E. Shannon and J. McCarthy, editors, *Automata Studies*, Annals of Mathematics Studies vol. 34, pages 3–40. Princeton University Press, 1956.

[KM00] D. Kuske and R. Morin. Pomsets for local trace languages: Recognizability, logic and Petri nets. *Journal for Languages, Automata, and Combinatorics*, 2000. To appear.

[KP92] S. Katz and D. Peled. Defining conditional independence using collapses. *Theoretical Comp. Science*, 101:337–359, 1992.

[KS98] D. Kuske and R.M. Shortt. Topology for computations of concurrent automata. *International Journal of Algebra and Computation*, 8:327–362, 1998.

[Kus94a] D. Kuske. *Modelle nebenläufiger Prozesse – Monoide, Residuensysteme und Automaten.* PhD thesis, Universität GHS Essen, 1994.

[Kus94b] D. Kuske. Nondeterministic automata with concurrency relations and domains. In S. Tison, editor, *Colloquium on Trees in Algebra and Programming*, Lecture Notes in Comp. Science vol. 787, pages 202–217. Springer, 1994.

[Kus98] D. Kuske. Asynchronous cellular automata and asynchronous automata for pomsets. In *CONCUR'98*, Lecture Notes in Comp. Science vol. 1466, pages 517–532. Springer, 1998.

[Kus99] D. Kuske. Symmetries of the partial order of traces. *Order*, 16:133–148, 1999.

[Kwi90] M. Kwiatkowska. A metric for traces. *Information Processing Letters*, 35:129–135, 1990.

[LW98a] K. Lodaya and P. Weil. A Kleene iteration for parallelism. In V. Arvind and R. Ramanujam, editors, *FST and TCS 98*, Lecture Nodes in Computer Science vol. 1530, pages 355–366. Springer, 1998.

[LW98b] K. Lodaya and P. Weil. Series-parallel posets: algebra, automata and languages. In M. Morvan, Ch. Meinel, and D. Krob, editors, *STACS98*, Lecture Nodes in Computer Science vol. 1373, pages 555–565. Springer, 1998.

[LW00] K. Lodaya and P. Weil. Series-parallel languages and the bounded-width property. *Theoretical Comp. Science*, 237:347–380, 2000.

[Maz77] A. Mazurkiewicz. Concurrent program schemes and their interpretation. Technical report, DAIMI Report PB-78, Aarhus University, 1977.

[Maz87] A. Mazurkiewicz. Traces theory. In W. Brauer et al., editor, *Petri Nets, Applications and Relationship to other Models of Concurrency*, Lecture Notes in Comp. Science vol. 255, pages 279–324. Springer, 1987.

[Maz95] A. Mazurkiewicz. Introduction to trace theory. In V. Diekert and G. Rozenberg, editors, *The Book of Traces*, chapter 1, pages 3–42. World Scientific Publ. Co., 1995.

[Mét86] Y. Métivier. On recognizable subsets of free partially commutative monoids. In *ICALP'86*, Lecture Notes in Comp. Science, pages 254–264. Springer, 1986.

[Mol96] Faron Moller. Infinite results. In U. Montanari and V. Sassone, editors, *CONCUR'96*, Lecture Notes in Comp. Science vol. 1119, pages 195–216. Springer, 1996.

[Muk92] M. Mukund. Petri nets and step transition systems. *International Journal of Foundations of Computer Science*, 3:443–478, 1992.

[NRT90] M. Nielsen, G. Rosenberg, and P.S. Thiagarajan. Behavioural notions for elementary net systems. *Distributed Computing*, 4:45–59, 1990.

[NW63] C.St.J.A. Nash-Williams. On well-quasi-ordering finite trees. *Proc. Camb. Philos. Soc.*, 59:833–835, 1963.

[Och85] E. Ochmański. Regular behaviour of concurrent systems. *Bull. Europ. Assoc. for Theor. Comp. Science*, 27:56–67, 1985.

[PS90] M. Peletier and J. Sakarovitch. Easy multiplications. II. Extensions of rational semigroups. *Information and Computation*, 88:18–59, 1990.

[Rab69] M.O. Rabin. Decidability of second-order theories and automata on infinite trees. *Trans. Amer. Math. Soc.*, 141:1–35, 1969.

[Ram30] F.P. Ramsey. On a problem of formal logic. *Proc. London Math. Soc.*, 30:264–286, 1930.

[Sak87] J. Sakarovitch. Easy multiplications. I. The realm of Kleene's Theorem. *Information and Computation*, 74:173–197, 1987.

[Sch61] M.P. Schützenberger. On the definition of a family of automata. *Inf. Control*, 4:245–270, 1961.

[Sch98] V. Schmitt. Stable trace automata vs full trace automata. *Theoretical Comp. Science*, 200:45–100, 1998.

[She75] S. Shelah. The monadic theory of order. *Annals of Mathematics*, 102:379–419, 1975.

[Sta89] E.W. Stark. Connections between a concrete and an abstract model of concurrent systems. In *5th Conf. on the Mathematical Foundations of Programming Semantics*, Lecture Notes in Comp. Science vol. 389, pages 53–79. Springer, 1989.

[Ste91] M. Stern. *Semimodular lattices*. Teubner Texte vol. 125. Teubner, 1991.

[Teo93] D. Teodosiu. Bereichstheoretische Eigenschaften komplexer Spuren. Master's thesis, Universität Stuttgart, 1993.

[Tho90a] W. Thomas. Automata on infinite objects. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, pages 133–191. Elsevier Science Publ. B.V., 1990.

[Tho90b] W. Thomas. On logical definability of trace languages. In V. Diekert, editor, *Proceedings of a workshop of the ESPRIT BRA No 3166: Algebraic and Syntactic Methods in Computer Science (ASMICS) 1989*, Report TUM-I9002, Technical University of Munich, pages 172–182, 1990.

[Tho97a] W. Thomas. Ehrenfeucht games, the composition method, and the monadic theory of ordinal words. In J. Mycielski et al., editor, *Structures in Logic and Computer Science, A Selection of Essays in Honor* *of A. Ehrenfeucht*, Lecture Notes in Comp. Science vol. 1261, pages 118–143. Springer, 1997.

[Tho97b] W. Thomas. Languages, automata, and logic. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, pages 389–455. Springer Verlag, 1997.

[Win87] G. Winskel. Event structures. In W. Brauer, W. Reisig, and G. Rozenberg, editors, *Petri nets: Applications and Relationships to Other Models of Concurrency*, Lecture Notes in Comp. Science vol. 255, pages 325–392. Springer, 1987.

[Zie87] W. Zielonka. Notes on finite asynchronous automata. *R.A.I.R.O. - Informatique Théorique et Applications*, 21:99–135, 1987.

[Zie95] W. Zielonka. Asynchronous automata. In *[DR95]*, pages 205–248. 1995.

Index