# 3
# Transition systems

## 3.1 Transition-system spaces

This chapter introduces the semantic domain that is used throughout this book. The goal is to model reactive systems; the most important feature of such systems is the interaction between a system and its environment. To describe such systems, the well-known domain of *transition systems*, *process graphs*, or *automata* is chosen. In fact, it is the domain of non-deterministic (finite) automata known from formal language theory. An automaton models a system in terms of its states and the transitions that lead from one state to another state; transitions are labeled with the actions causing the state change. An automaton is said to describe the *operational* behavior of a system. An important observation is that, since the subject of study is interacting systems, not just the language generated by an automaton is important, but also the states traversed during a run or execution of the automaton. The term 'transition system' is the term most often used in reactive-systems modeling. Thus, also this book uses that term.

The semantic domain serves as the basis for the remainder of the book. The meaning of the various equational theories for reasoning about reactive systems developed in the remaining chapters is defined in terms of the semantic domain, in the way explained in the previous chapter. Technically, it turns out to be useful to embed all transition systems that are of interest in one large set of states and transitions, from which the individual transition systems can be extracted. Such a large set of states and transitions is called a *transition-system space*.

**Definition 3.1.1 (Transition-system space)** A *transition-system space* over a set of labels $L$ is a set $S$ of *states*, equipped with one ternary relation $\rightarrow$ and one subset $\downarrow$:

https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press

(i) $\rightarrow \subseteq S \times L \times S$ is the set of *transitions*;

(ii) $\downarrow \subseteq S$ is the set of *terminating* or *final* states.

The notation $s \overset{a}{\rightarrow} t$ is used for $(s, a, t) \in \rightarrow$. It is said that $s$ has an $a$-step to $t$. Notation $s\downarrow$ is used for $s \in \downarrow$, and it is often said that $s$ has a (successful) termination option; $s \notin \downarrow$ is denoted $s\not\downarrow$. The fact that for all states $t \in S$, it holds that $(s, a, t) \notin \rightarrow$, is abbreviated as $s \overset{a}{\not\rightarrow}$. It is also said that $s$ does not have an $a$-step.

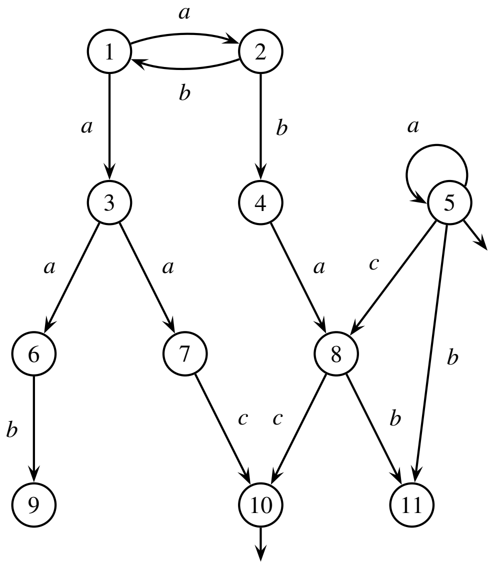


Fig. 3.1. An example of a transition-system space.

**Example 3.1.2 (Transition-system space)** Consider Figure 3.1. It visualizes a transition-system space with eleven states, depicted by circles. Terminating states are marked with a small outgoing arrow. Transitions are visualized by labeled arrows connecting two states.

In the remainder, assume that $(S, L, \rightarrow, \downarrow)$ is a transition-system space. Each state $s \in S$ can be identified with a transition system that consists of all states and transitions reachable from $s$. The notion of reachability is defined by generalizing the transition relation to sequences of labels, called words. The set of all words consisting of symbols from some set $A$ is denoted $A^*$.

**Definition 3.1.3 (Reachability)** The reachability relation $\rightarrow^* \subseteq S \times L^* \times S$ is defined inductively by the following clauses:

https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press

(i) $s \xrightarrow{\epsilon}{}^* s$ for each $s \in S$, where $\epsilon \in L^*$ denotes the empty word;

(ii) for all $s, t, u \in S, \sigma \in L^*$, and $a \in L$, if $s \xrightarrow{\sigma}{}^* t$ and $t \xrightarrow{a} u$, then $s \xrightarrow{\sigma a}{}^* u$, where $\sigma a$ is the word obtained by appending $a$ at the end of $\sigma$.

A state $t \in S$ is said to be *reachable* from state $s \in S$ if and only if there is a word $\sigma \in L^*$ such that $s \xrightarrow{\sigma}{}^* t$.

**Example 3.1.4 (Reachability)** Consider the transition-system space of Figure 3.1. From states 1 and 2, all states except state 5 are reachable; from state 3, states 3, 6, 7, 9, and 10 are reachable, et cetera.

**Definition 3.1.5 (Transition system)** For each state $s \in S$, the *transition system* induced by $s$ consists of all states reachable from $s$, and it has the transitions and final states induced by the transition-system space. State $s$ is called the *initial state* or *root* of the transition system associated with $s$. Note that often the terminology 'transition system $s$' is used to refer to the transition system *induced* by $s$.

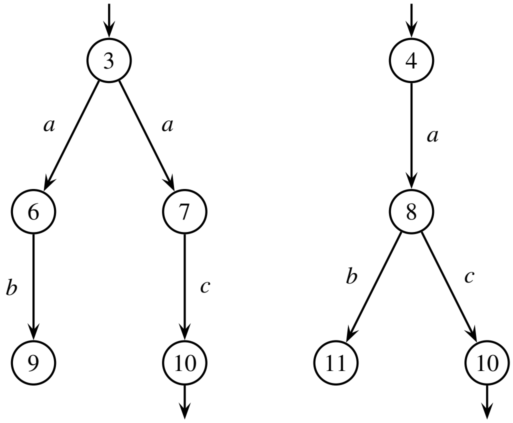


Fig. 3.2. Examples of simple transition systems.

**Example 3.1.6 (Transition systems)** Consider again Figure 3.1. The transition system with root state 3 consists of states 3, 6, 7, 9, and 10. Taking state 4 as the root yields a transition system consisting of states 4, 8, 10, and 11. Figure 3.2 visualizes these two transition systems; the root states are marked with small incoming arrows. The states are rearranged slightly to emphasize the similarity between the two transition systems.

https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press

It follows from the above definitions that the notion of a transition system coincides with the classical notion of an automaton. The classical notion of language equivalence on automata says that two automata are equivalent if and only if they have the same *runs* from the initial state to a final state.

**Definition 3.1.7 (Run, language equivalence)** A word $\sigma \in L^*$ is a *complete execution* or *run* of a transition system $s \in S$ if and only if there is some $t \in S$ with $s \xrightarrow{\sigma}{}^* t$ and $t\downarrow$.

Two transition systems are language equivalent if and only if they have the same set of runs.

**Example 3.1.8 (Language equivalence)** Consider once more the two transition systems depicted in Figure 3.2. Both transition systems have only one run, namely $ac$. The two systems are thus language equivalent. Note that $ab$ is not a run of any of the two transition systems because states 9 and 11 are non-terminating.

Language equivalence turns out to be insufficient for our purposes. Since interaction of automata is considered, there is a need to consider all states of an automaton, not just the complete runs. This can be illustrated with Frank Stockton's story 'The Lady or the Tiger?' (see (Smullyan, 1982)).

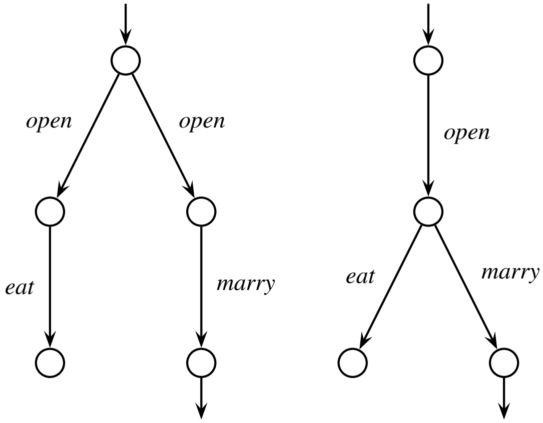


Fig. 3.3. The lady or the tiger?

**Example 3.1.9 (The lady or the tiger?)** A prisoner is confronted with two closed doors. Behind one of the doors is a dangerous tiger, and behind the other there is a beautiful lady. If the prisoner opens the door hiding the tiger

https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press

he will be eaten. On the other hand, if he opens the door hiding the beautiful lady, the lady and he will get married and he will be free. Unfortunately, the prisoner does not know what hides behind what door.

This situation can be described as a transition system using the following actions:

(i) *open* representing the act of opening a door;
(ii) *eat* representing the act of (the tiger) eating the young man;
(iii) *marry* representing the act of the beautiful lady and the prisoner getting married.

The situation described above can be modeled by the left transition system depicted in Figure 3.3. This transition system models the fact that after a door has been opened the prisoner is confronted with either the tiger or the beautiful lady. He does not have the possibility to select his favorite, which conforms to the description above. The observant reader might notice the similarity with the left transition system of Figure 3.2. Note the subtle fact that transition *marry* ends in a terminating state, whereas transition *eat* does not. This can be interpreted to mean that the *marry* transition results in a *successful* termination of the process, and that the *eat* transition results in *unsuccessful* termination. A non-terminating state in which no actions are possible, such as the state resulting from the *eat* transition, is often called a *deadlock* state (see also Definition 3.1.14).

One might wonder whether the same process cannot be described by the slightly simpler transition system on the right of Figure 3.3, which strongly resembles the right transition system in Figure 3.2. However, there are good reasons why the situation modeled by the right transition system of Figure 3.3 is different from the situation described by the other transition system in that figure. In the right transition system, the choice between the tiger and the lady is only made after opening a door. It either describes a situation with only one door leading to both the tiger and the lady or a situation with two doors both leading to both the tiger and the lady; in either case, the prisoner still has a choice after opening a door. Clearly, these situations differ considerably from the situation described above.

The above example illustrates that there are good reasons to distinguish the two transition systems of Figure 3.2. Example 3.1.8 shows that these two transition systems are language equivalent, thus confirming the earlier remark that language equivalence does not suit our purposes. It does not in all cases distinguish transition systems that clearly model different situations. Two aspects are not sufficiently taken into account. First, two transition systems that are language equivalent may have completely different sets of ‘runs’ that end in

https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press

a non-terminating state. As explained in the above example, terminating and non-terminating states may be used to model successful and unsuccessful termination of a process, respectively. In such cases, it is desirable to distinguish transition systems that have the same sets of runs ending in terminating states but that differ in executions leading to non-terminating states. Second, language equivalence does not distinguish transition systems that have the same runs but that are different in the moments that *choices* are made. To illustrate this, consider again the transition system on the right-hand side of Figure 3.2. It has a state in which both $b$ and $c$ are possible as the following action (state 8), whereas the system on the left only has states where exactly one of the two is possible (states 6 and 7). Thus, if for some reason action $c$ is impossible, is blocked, then the transition system on the left can become stuck after the execution of an $a$ action, whereas the one on the right cannot. The choice whether or not to execute $b$ or $c$ in the left transition system is made (implicitly) upon execution of the $a$ action in the initial state, whereas the same choice is made only after execution of the initial $a$ action in the right transition system. It is said that the transition systems have different *branching structure*.

A choice between alternatives that are initially identical, as in state 3 of the leftmost transition system of Figure 3.2, is called a *non-deterministic* choice. Such choices are the subject of many investigations in the theory of concurrency. A detailed treatment of non-determinism is beyond the scope of this book.

It is often desirable to be able to distinguish between transition systems with the same runs that have different termination behavior or that have different branching structure. In order to do this, a notion of equivalence is defined that is finer than language equivalence, in the sense that it distinguishes transition systems accepting the same language but with different termination behavior or branching structure. This notion is called bisimilarity or bisimulation equivalence.

**Definition 3.1.10 (Bisimilarity)** A binary relation $R$ on the set of states $S$ of a transition-system space is a *bisimulation* relation if and only if the following so-called *transfer conditions* hold:

(i) for all states $s, t, s' \in S$, whenever $(s, t) \in R$ and $s \xrightarrow{a} s'$ for some $a \in L$, then there is a state $t'$ such that $t \xrightarrow{a} t'$ and $(s', t') \in R$;

(ii) vice versa, for all states $s, t, t' \in S$, whenever $(s, t) \in R$ and $t \xrightarrow{a} t'$ for some $a \in L$, then there is a state $s'$ such that $s \xrightarrow{a} s'$ and $(s', t') \in R$;

(iii) whenever $(s, t) \in R$ and $s\downarrow$ then $t\downarrow$;

(iv) whenever $(s, t) \in R$ and $t\downarrow$ then $s\downarrow$.

https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press

The transfer conditions are illustrated in Figures 3.4 and 3.5. These figures show the transition systems starting from $s$ and $t$ as consisting of disjoint sets of states, but this is just done for clarity; it is possible that states are shared.

Two transition systems $s, t \in S$ are *bisimulation equivalent* or *bisimilar*, notation $s \leftrightarroweq t$, if and only if there is a bisimulation relation $R$ on $S$ with $(s, t) \in R$.

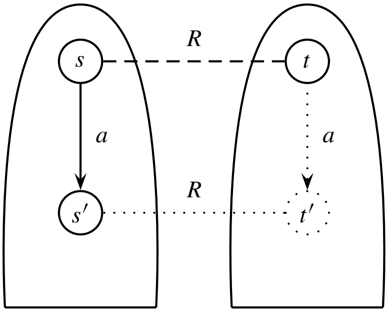


Fig. 3.4. Transfer condition (i) of a bisimulation.

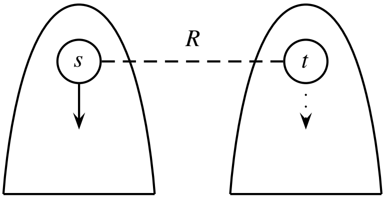


Fig. 3.5. Transfer condition (iii) of a bisimulation.

One can think of the notion of bisimilarity in terms of a two-person game. Suppose two players each have their own behavior, captured in the form of a transition system. The game is played as follows. First, one of the players makes a transition or move from the initial state of his or her transition system. The role of the other player is to match this move precisely, also starting from the initial state. Next, again one of the players makes a move. This does not have to be the same player as the one that made the first move. The other must try to match this move, and so on. If both players can play in such a way that at each point in the game any move by one of the players can be matched by a move of the other player, then the transition systems are bisimilar. Otherwise, they are not.

https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press

**Example 3.1.11 (Bisimilarity)** In Figure 3.6, an example of a bisimulation relation on transition systems is given. (Note that, technically, the two transition systems are embedded in one transition-system space.) Related states are connected by a dashed line. Since the initial states of the two transition systems are related, they are bisimilar.

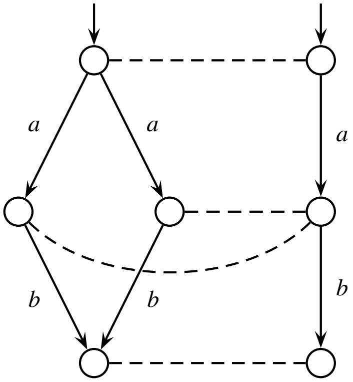


Fig. 3.6. Example of a bisimulation.

**Example 3.1.12 (Bisimilarity)** Figure 3.7 recalls two by now well-known transition systems. It should not come as a surprise that they are not bisimilar. States that can possibly be related are connected by a dashed line. The states where the behaviors of the two systems differ are indicated by dotted lines labeled with a question mark. None of the two indicated pairs of states satisfies the transfer conditions of Definition 3.1.10 (Bisimilarity).

So far, bisimilarity is just a relation on transition systems. However, it has already been mentioned that it is meant to serve as a notion of equality. For that purpose, it is necessary that bisimilarity is an equivalence relation. It is not difficult to show that bisimilarity is indeed an equivalence relation.

**Theorem 3.1.13 (Equivalence)** Bisimilarity is an equivalence.

*Proof* Let $S$ be the set of states of a transition-system space. Proving that a relation is an equivalence means that it must be shown that it is reflexive, symmetric, and transitive. First, it is not hard to see that the relation $R = \{(s, s) \mid s \in S\}$ is a bisimulation relation. This implies that $s \underline{\leftrightarrow} s$ for any (transition system induced by) state $s \in S$. Second, assume that $s \underline{\leftrightarrow} t$ for states $s, t \in S$. If $R$ is a bisimulation relation such that $(s, t) \in R$, then the relation

https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press

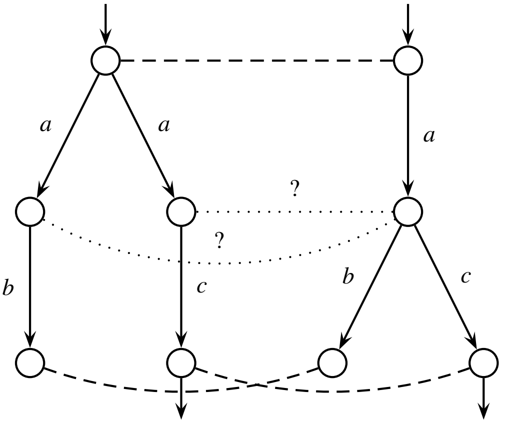


Fig. 3.7. Two transition systems that are not bisimilar.

$R' = \{(v, u) \mid (u, v) \in R\}$ is a bisimulation relation as well. Moreover, as $(s, t) \in R$, obviously $(t, s) \in R'$. Hence, $t \underline{\leftrightarrow} s$, proving symmetry of $\underline{\leftrightarrow}$. Finally, for transitivity of $\underline{\leftrightarrow}$, it must be shown that the relation composition of two bisimulation relations results in a bisimulation relation again. Let $s$, $t$, and $u$ be states in $S$, and let $R_1$ and $R_2$ be bisimulation relations with $(s, t) \in R_1$ and $(t, u) \in R_2$. The relation composition $R_1 \circ R_2$ is a bisimulation with $(s, u) \in R_1 \circ R_2$, implying transitivity of $\underline{\leftrightarrow}$. The detailed proof is left as an exercise to the reader (Exercise 3.1.6). □

The discussion leading from the notion of language equivalence to the introduction of bisimilarity is illustrative for a more general problem. When designing a semantic framework based on transition systems, one has to choose a meaningful equivalence on transition systems. The question of what equivalence is most suitable depends on the context and is often difficult to answer. In fact, the question is considered so important that it has generated its own field of research, called comparative concurrency semantics. In Chapter 12, the problem of choosing an appropriate equivalence is investigated in more detail. In the other chapters of this book, bisimilarity serves as the main semantic equivalence.

To end this section, several relevant subclasses of transition systems are defined.

**Definition 3.1.14 (Deadlock)** A state $s$ of a transition system as defined in Definition 3.1.5 (Transition system) is a *deadlock state* if and only if it does not have any outgoing transitions and it does not allow successful termination,

https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press

i.e., if and only if for all $a \in L$, $s \overset{a}{\not\to}$, and $s \notin {\downarrow}$. A transition system *has a deadlock* if and only if it has a deadlock state; it is *deadlock free* if and only if it does not have a deadlock.

An important property that has already been suggested when motivating the notion of bisimilarity is that bisimilarity preserves deadlocks. Exercise 3.1.5 asks for a proof of this fact.

**Definition 3.1.15 (Regular transition system)** A transition system is *regular* if and only if both its sets of states and of transitions are *finite*.

The reader familiar with automata theory will notice that a regular transition system is simply a finite automaton. The fact that finite automata define the class of regular languages explains the name 'regular transition system'.

**Definition 3.1.16 (Finitely branching transition system)** A transition system is *finitely branching* if and only if each of its states has only *finitely many outgoing transitions*; if any of its states has infinitely many outgoing transitions, the transition system is said to be infinitely branching.

Note that a regular transition system is by definition finitely branching.

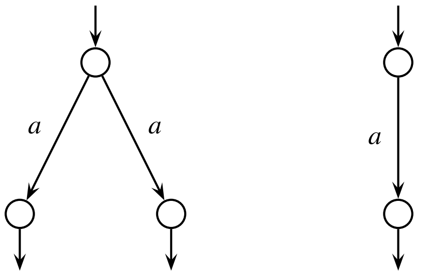
So far, it has not been made precise what are the states in a transition-system space. In the remainder, however, the states in a transition-system space consist of terms from a particular equational theory, as introduced in the previous chapter.

### Exercises

3.1.1 Consider the transition-system space of Figure 3.1. Draw the transition system with root 5.

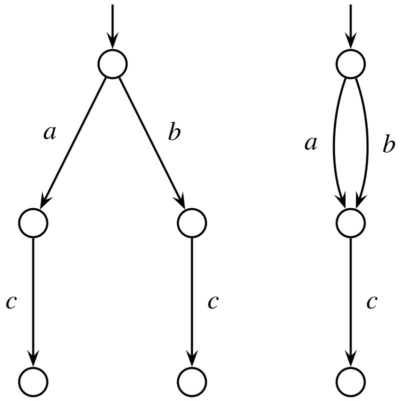
3.1.2 Are the following pairs of transition systems bisimilar? If so, give a bisimulation between the two systems; otherwise, explain why they are not bisimilar.

(a)

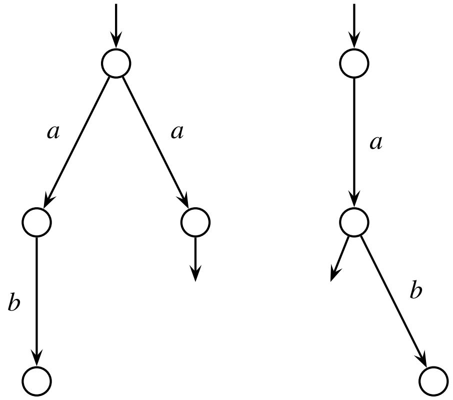


https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press

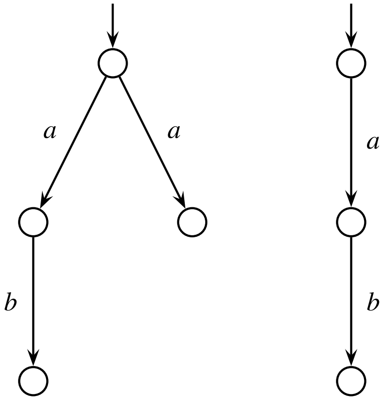
(b)



(c)

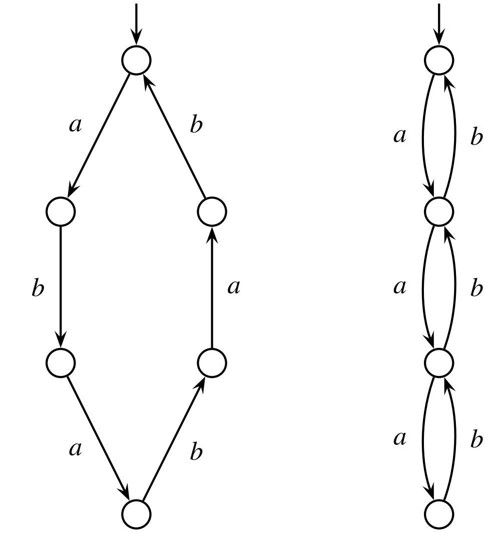


(d)

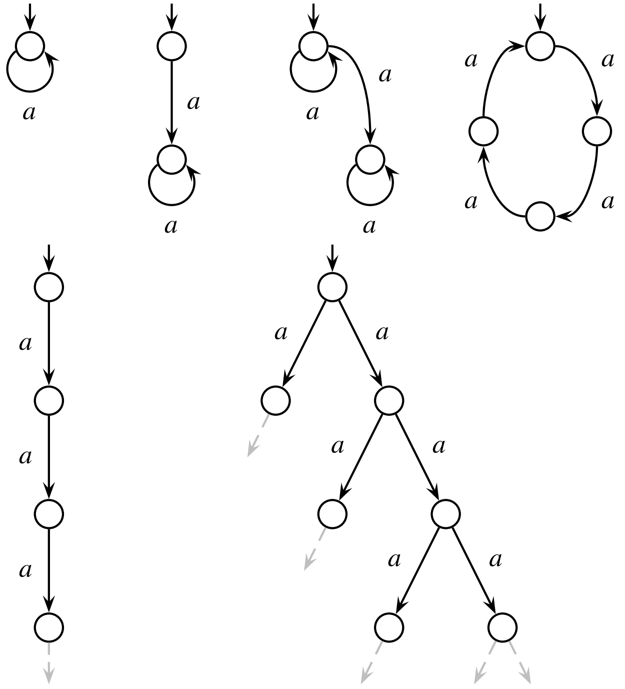


https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press

(e)



3.1.3 Give a bisimulation between any two of the following six transition systems.



https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press

3.1.4 Recall Definitions 3.1.14 (Deadlock), 3.1.15 (Regular transition system), and 3.1.16 (Finitely branching transition system). Which of the transition systems of Exercises 3.1.2 and 3.1.3 are deadlock free? Which ones are regular? Which ones are finitely branching?

3.1.5 Recall Definition 3.1.14 (Deadlock). Let $s$ and $t$ be two *bisimilar* transition systems in some transition-system space. Show that $s$ has a deadlock if and only if $t$ has a deadlock.

3.1.6 Prove that the relation composition of two bisimulation relations is again a bisimulation.

3.1.7 Formalize the game-theoretic characterization of bisimilarity using concepts from game theory (see e.g. (Osborne & Rubinstein, 1994)).

3.1.8 Prove that the union of all possible bisimulation relations on a transition-system space is again a bisimulation relation. This is called the *maximal* bisimulation relation.

3.1.9 Let a set $C$ be given, the set of *colors*. A *coloring* of a transition-system space $(S, L, \rightarrow, \downarrow)$ is a mapping from $S$ to $C$, so each node has a color. A *colored trace* starting from $s \in S$ is a sequence $(c_0, a_1, c_1, \ldots, a_k, c_k)$ such that there are states $s_1, \ldots, s_k$ in $S$ with $s \overset{a_1}{\rightarrow} s_1 \ldots \overset{a_k}{\rightarrow} s_k$, $s$ has color $c_0$ and $s_i$ has color $c_i$ $(1 \leq i \leq k)$. A coloring is *consistent* if two nodes have the same color only if the same colored traces start from them and if they are both terminating or both non-terminating. Prove that two nodes are bisimilar exactly when there is a consistent coloring in which both nodes have the same color.

3.1.10 Recall Definition 2.3.26 (Isomorphism of algebras). On a transition-system space, the notion of isomorphism can be defined as follows: two transition systems $s, t$ are isomorphic if and only if there is a bijective function $f$ between the set of states reachable from $s$ and the set of states reachable from $t$ such that $u \overset{a}{\rightarrow} v \Leftrightarrow f(u) \overset{a}{\rightarrow} f(v)$ and $u\downarrow \Leftrightarrow f(u)\downarrow$. Prove that isomorphism is an equivalence relation on the transition-system space that is finer than bisimilarity, i.e., prove that isomorphic transition systems are bisimilar and give an example of two transition systems that are bisimilar but not isomorphic.

## 3.2 Structural operational semantics

In the previous section, the notion of a transition-system space was introduced. In this section, this is expanded upon by turning the set of states into a set of

https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press

(closed) terms over some equational theory. Thus, it is shown how transition-system spaces can be used to provide equational theories with operational semantics.

**Example 3.2.1 (Operational semantics)** An illustration of what is involved is given by considering the equational theory $T_1 = (\Sigma_1, E_1)$ of Example 2.2.5. As shown in Example 2.2.11, each closed term over this theory is derivably equal to a closed term of the form $\mathbf{s}^m(\mathbf{0})$ for some natural number $m$. These terms $\mathbf{s}^m(\mathbf{0})$ can be transformed into a transition system as follows: the term $\mathbf{0}$ is a terminating state, and there is a transition from state $\mathbf{s}^{n+1}(\mathbf{0})$ to state $\mathbf{s}^n(\mathbf{0})$ for each natural number $n$. For simplicity, just one label is considered here, denoted 1. In this way, the state $\mathbf{s}^m(\mathbf{0})$ is characterized by having a sequence of exactly $m$ transitions to a terminating state. Figure 3.8 shows the transition system induced by closed term $\mathbf{s}^2(\mathbf{0})$.

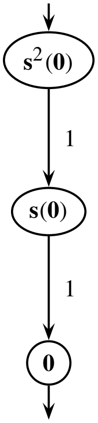


Fig. 3.8. The transition system corresponding to closed $\Sigma_1$-term $\mathbf{s}^2(\mathbf{0})$.

The general idea is to define transition systems for all closed $\Sigma_1$-terms of theory $T_1$ such that two closed terms, i.e., states, are bisimilar exactly when these terms are derivably equal in the equational theory. The desired transition-system space can be constructed in the following way: any transition for a certain closed term led by a certain function symbol is derived from the transitions of the arguments of the function symbol. The same holds for the termination predicate: whether or not a term is a terminating state is derived from the termination of the arguments of the leading function symbol. In other words, transitions and termination are determined based on the structure of terms. Since the transition-systems framework has an operational flavor, the

https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press

resulting semantics of the equational theory is often referred to as a *structural operational semantics*. The rules defining all possible transitions and terminating states in a transition-system space constructed in this way are called *deduction rules*. The conditions on the arguments of the leading function symbol are called the *premises* of the deduction rule. Deduction rules without premises are often referred to as *axioms*.

For the example theory $T_1$, first, there are two axioms, corresponding to the two cases already described above: $\mathbf{0}$ is a terminating state, and each term that starts with a successor function symbol can do a step. Formally, for any $\Sigma_1$-term $x$,

$$\mathbf{0}{\downarrow} \qquad \mathbf{s}(x) \xrightarrow{1} x.$$

Note that the second axiom holds for arbitrary (open) $\Sigma_1$-terms; however, the transition-system space that will be derived from these rules is built from closed terms only.

Second, the addition function symbol is given an operational semantics. As mentioned, each closed $\Sigma_1$-term is derivably equal to a closed term of the form $\mathbf{s}^m(\mathbf{0})$ for some natural number $m$. The aim to obtain a semantics such that two derivably equal terms correspond to bisimilar transition systems can be achieved by ensuring that each closed term that is derivably equal to a closed term of the form $\mathbf{s}^m(\mathbf{0})$ has a single path of exactly $m$ transitions to a terminating state. Deduction rules for the function symbol $\mathbf{a}$ can therefore be formulated as follows. For any $\Sigma_1$-terms $x, x', y, y'$,

$$\frac{y \xrightarrow{1} y'}{\mathbf{a}(x, y) \xrightarrow{1} \mathbf{a}(x, y')} \quad \frac{x \xrightarrow{1} x',\ y{\downarrow}}{\mathbf{a}(x, y) \xrightarrow{1} x'} \quad \frac{x{\downarrow},\ y{\downarrow}}{\mathbf{a}(x, y){\downarrow}}.$$

The first rule says that whenever a term $y$ has a transition to term $y'$, then $\mathbf{a}(x, y)$ has a transition to $\mathbf{a}(x, y')$ for any arbitrary term $x$. The second rule states that whenever a term $x$ can do a step to $x'$ and term $y$ cannot do a step, the term $\mathbf{a}(x, y)$ can do a step to $x'$. The idea of these two rules is that first the second argument of a term with leading symbol $\mathbf{a}$ does all its steps; if the second argument cannot do any further steps, then the first argument continues with its steps. Of course, it is also possible to first let the first argument do its steps and then the second. In principle, it is even possible to allow arbitrary interleavings of the steps of the two arguments. However, the above rules follow Axioms PA1 and PA2 given in Table 2.1. This choice simplifies the proofs of certain properties that are given in the remainder. The third and last rule given above says that $\mathbf{a}(x, y)$ is a terminating state if both $x$ and $y$ are terminating states.

By repeatedly applying the deduction rules given so far, it can be established

https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press

that $\mathbf{a}(\mathbf{s}(\mathbf{0}), \mathbf{s}(\mathbf{0}))$ has the expected sequence of two transitions to a terminating state. From the leftmost axiom given above, for example, it is immediately clear that $\mathbf{0}\downarrow$. From the second axiom, it follows that $\mathbf{s}(\mathbf{0}) \xrightarrow{1} \mathbf{0}$. From these two facts, using the first deduction rule for the addition function symbol, it is obtained that $\mathbf{a}(\mathbf{s}(\mathbf{0}), \mathbf{0}) \xrightarrow{1} \mathbf{0}$. Continuing in this way results in the transition system given in Figure 3.9.

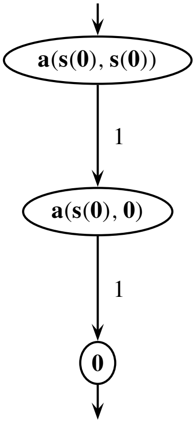


Fig. 3.9. The transition system corresponding to $\mathbf{a}(\mathbf{s}(\mathbf{0}), \mathbf{s}(\mathbf{0}))$.

In Example 2.2.9, it has already been shown that

$$T_1 \vdash \mathbf{a}(\mathbf{s}(\mathbf{0}), \mathbf{s}(\mathbf{0})) = \mathbf{s}(\mathbf{s}(\mathbf{0})).$$

From the two transition systems given in Figures 3.8 and 3.9, it immediately follows that

$$\mathbf{a}(\mathbf{s}(\mathbf{0}), \mathbf{s}(\mathbf{0})) \leftrightarroweq \mathbf{s}(\mathbf{s}(\mathbf{0})),$$

where, as explained before, closed terms are interpreted as states in a transition-system space or initial states of a transition system. The above two results conform to the aim that two closed $\Sigma_1$-terms are bisimilar if and only if these terms are derivably equal in the equational theory $T_1$.

To complete the example, consider the following deduction rules that define the transitions involving the function symbol $\mathbf{m}$. For any $\Sigma_1$-terms $x, x', y$, and $y'$,

$$\frac{x \xrightarrow{1} x', y \xrightarrow{1} y'}{\mathbf{m}(x, y) \xrightarrow{1} \mathbf{a}(\mathbf{m}(x, y'), x')} \quad \frac{x\downarrow}{\mathbf{m}(x, y)\downarrow} \quad \frac{y\downarrow}{\mathbf{m}(x, y)\downarrow}.$$

The reader is urged to try to understand these rules. Also in this case the rules

https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press

guarantee that each closed term has a transition system consisting of a single path of transitions to a terminating state, where the length of the path is the 'natural-number interpretation' of the term.

The pair of a signature and a set of deduction rules for terms over this signature is called a *deduction system*. The deduction system developed in this example forms the basis for a transition-system space. The set of states of the transition-system space is the set of *closed* $\Sigma_1$-terms, $\mathcal{C}(\Sigma_1)$ (see Definition 2.2.3 (Terms)); the set of action labels is the singleton $\{1\}$. The transition relation is the *smallest* relation in $\mathcal{C}(\Sigma_1) \times \{1\} \times \mathcal{C}(\Sigma_1)$ satisfying the deduction rules given above. The requirement that it is the smallest relation guarantees that no transitions are present that cannot be derived from the deduction rules. Similarly, the set of terminating states is the smallest set of states satisfying the deduction rules.

Unfortunately, the transition-system space given in this example is not an algebra in the sense of Section 2.3, and it cannot be turned into a model for theory $T_1$ (see Definition 2.3.8 (Model)) without some additional work. In the remainder, it is shown how the transition-system space of this example can be turned into a model of $T_1$; then, it is also possible to formally prove the claim that two closed $\Sigma_1$-terms are bisimilar if and only if they are derivably equal.

The following definitions precisely introduce the notions of a deduction system and its induced transition-system space.

**Definition 3.2.2 (Deduction system)** A *term deduction system*, or simply a *deduction system*, is a pair $(\Sigma, R)$, where $\Sigma$ is a signature as defined in Definition 2.2.1 (Signature), and where $R$ is a set of *rules*. A *rule* is of the form

$$\frac{\Phi}{\psi},$$

where $\psi$ is a so-called *formula* (called the *conclusion* of the rule), and where $\Phi$ is a *set* of formulas (called the *premises* of the rule). A formula is either a *transition* $s \overset{a}{\to} t$ or a *termination* $t\!\downarrow$, for $\Sigma$-terms $s, t$ and a label $a$ taken from some given set of labels $L$. The term $s$ in a transition $s \overset{a}{\to} t$ is called the *source* of the transition; term $t$ is called the *target*. The term in a termination is referred to as the source of the termination. If the set of premises $\Phi$ is empty, then a rule is called an *axiom*.

**Definition 3.2.3 (Transition-system space induced by a deduction system)** Assume $(\Sigma, R)$ is a term deduction system; furthermore, assume that $L$ is the set of labels occurring in the set of rules $R$. The transition-system space induced by $(\Sigma, R)$ is the quadruple $(\mathcal{C}(\Sigma), L, \to, \downarrow)$ where $\to \subseteq \mathcal{C}(\Sigma) \times L \times$

https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press

$\mathcal{C}(\Sigma)$ and $\downarrow \subseteq \mathcal{C}(\Sigma)$ contain a formula of $(\Sigma, R)$ if and only if this formula is a closed substitution instance of a conclusion of a rule in $R$ of which all the closed instances of the premises are also in $\rightarrow$ or $\downarrow$. In other words, a closed formula $\phi$ (i.e., a formula containing only closed terms) is an element of $\rightarrow$ or $\downarrow$ exactly when there is a rule $\frac{\Phi}{\psi}$ and a substitution $\sigma$ such that $\phi \equiv \psi[\sigma]$ and the formulas $\chi[\sigma]$, for each $\chi \in \Phi$, are elements of $\rightarrow$ or $\downarrow$. (Recall the notion of substitution from Definition 2.2.6; also recall that $\equiv$ denotes syntactical identity on terms. In the current definition, both substitution and syntactical identity are overloaded to formulas, but that should not cause confusion.)

The example and the definitions given so far show that a deduction system is a way to turn the set of closed terms over a signature into a transition-system space and, thus, the individual closed terms into transition systems. The next step is to show how such a transition-system space can be transformed into a *model* of an equational theory over that signature, in the sense of Section 2.3. The key idea is to find an equivalence relation on the transition-system space that matches the notion of derivability in the equational theory, and to use this equivalence to construct a quotient algebra as defined in Definition 2.3.18; this quotient algebra is then a model of the equational theory. It should not come as a surprise that for the example theory $T_1$ bisimilarity is the equivalence notion matching derivability.

Consider Definition 2.3.18 (Quotient algebra). It reveals two important aspects. First, the starting point of a quotient algebra is an algebra (see Definition 2.3.1 (Algebra)); second, the equivalence relation must be a congruence on this algebra (see Definition 2.3.16 (Congruence)).

Concerning the first aspect, as already mentioned, a transition-system space is not an algebra. However, if a transition-system space is constructed from a term deduction system, as illustrated in Example 3.2.1, then it is straightforward to turn the space into an algebra of transition systems. Note that the set of closed terms over any signature with the constants and function symbols of the signature forms an algebra, often referred to as the term algebra. According to Definition 3.1.5 (Transition system), the closed terms over the signature of a term deduction system can be interpreted as transition systems in the transition-system space induced by the deduction system, which means that the algebra of these closed terms can be seen as an algebra of transition systems.

**Example 3.2.4 (Algebra of transition systems)** Consider again the example equational theory $T_1 = (\Sigma_1, E_1)$ of Example 2.2.5. Based on the transition-system space introduced in Example 3.2.1, the set of closed $\Sigma_1$-terms $\mathcal{C}(\Sigma_1)$

https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press

with constant **0** and functions **s**, **a**, and **m** forms an algebra of transition systems, referred to as $\mathbb{A}_1$.

Considering the second aspect mentioned above, it turns out that it is often also not too difficult to show that bisimilarity is a congruence on an algebra of transition systems constructed from a term deduction system. Theorem 3.1.13 implies that bisimilarity is an equivalence relation on transition systems. Thus, if it is possible to prove the second condition in Definition 2.3.16 (Congruence) for bisimilarity as well, it is possible to construct a quotient algebra from an algebra of transition systems and bisimilarity. In general, proving that a bisimilarity is a congruence on some algebra of transition systems can be quite tedious. However, it turns out that if the rules of the term deduction system used for constructing this algebra adhere to a certain simple format, then the desired congruence result follows automatically. The following definition introduces this format.

**Definition 3.2.5 (Path format)** Let $(\Sigma, R)$ be a deduction system. Consider the following restrictions on a deduction rule in $R$:

(i) any target of any transition in the premises is just a single variable;

(ii) the source of the conclusion is either a single variable or it is of the form $f(x_1, \ldots, x_n)$, for some natural number $n$, with $f \in \Sigma$ an $n$-ary function symbol and the $x_i$ variables;

(iii) the variables in the targets of the transitions in the premises and the variables in the source of the conclusion are all distinct, i.e., two targets of transitions in the premises are not the same variable and the variables in the source of the conclusion are all different, and no target of a premise occurs in the source of the conclusion.

If a rule satisfies these conditions, then it is said to be in *path* format. A deduction system is in path format if all its rules are.

**Example 3.2.6 (Path format)** It is easily seen that the deduction system for equational theory $T_1$ introduced in Example 3.2.1 is in path format.

The exercises at the end of this section contain some examples illustrating that the restrictions in the above definition are necessary in order to ensure that the desired congruence result holds. Almost all deduction systems given in this book turn out to be in path format.

The following theorem is given without proof. The interested reader can find a proof in (Baeten & Verhoef, 1993).

https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press

**Theorem 3.2.7 (Congruence theorem)** Consider a deduction system in path format. Bisimilarity is a congruence on the induced algebra of transition systems.

**Example 3.2.8 (Congruence, quotient algebra)** Consider once again the equational theory $T_1 = (\Sigma_1, E_1)$. It follows from the above congruence theorem and the fact that the deduction system for $T_1$ of Example 3.2.1 is in path format (Example 3.2.6) that bisimilarity is a congruence on the algebra $\mathbb{A}_1$ of transition systems introduced in Example 3.2.4. Thus, at this point, it is possible to construct a quotient algebra. Following Definition 2.3.18 (Quotient algebra), let $\mathbb{M}_1$ be the quotient algebra $\mathbb{A}_{1/\leftrightarrow}$ with domain $[\mathcal{C}(\Sigma_1)]_{\leftrightarrow}$, constant $[\mathbf{0}]_{\leftrightarrow}$ and functions $\mathbf{s}_{\leftrightarrow}$, $\mathbf{a}_{\leftrightarrow}$, and $\mathbf{m}_{\leftrightarrow}$. It turns out that this quotient algebra is a model of theory $T_1$, as defined in Definition 2.3.8; it can even be shown that this model is ground-complete, as defined in Definition 2.3.23.

Recall that it is our goal to prove that two closed $\Sigma_1$-terms are bisimilar if and only if they are derivably equal in $T_1$. The attentive reader might see that this result follows from the soundness and ground-completeness of theory $T_1$ for model $\mathbb{M}_1$ of the above example. Thus, the following two theorems formally prove these soundness and ground-completeness results. The proofs themselves can be skipped on a first reading (or on any reading, for that matter). They are included for the sake of completeness and to show the interested reader what is involved in soundness and ground-completeness proofs.

**Theorem 3.2.9 (Soundness of $T_1$)** Algebra $\mathbb{M}_1$ of Example 3.2.8 is a model of theory $T_1$ using the standard interpretation of constant $\mathbf{0}$ and functions $\mathbf{s}$, $\mathbf{a}$, and $\mathbf{m}$ explained in Definition 2.3.18 (Quotient algebra).

*Proof* Definition 2.3.8 (Model) implies that for each axiom $s = t$ of theory $T_1$ it must be shown that $\mathbb{M}_1 \models s = t$. Only the proof for Axiom PA2 is given; the proofs for the other axioms are left as Exercise 3.2.7.

Recall the notations concerning quotient algebras introduced in Section 2.3. Let $p$ and $q$ be arbitrary closed $\Sigma_1$-terms. Definition 2.3.6 (Validity) states that the following equality must be proven: $\mathbf{a}_{\leftrightarrow}([p]_{\leftrightarrow}, \mathbf{s}_{\leftrightarrow}([q]_{\leftrightarrow})) = \mathbf{s}_{\leftrightarrow}(\mathbf{a}_{\leftrightarrow}([p]_{\leftrightarrow}, [q]_{\leftrightarrow}))$. It follows from Definition 2.3.18 (Quotient algebra) that it therefore must be shown that $[\mathbf{a}(p, \mathbf{s}(q))]_{\leftrightarrow} = [\mathbf{s}(\mathbf{a}(p, q))]_{\leftrightarrow}$ and, thus, that $\mathbf{a}(p, \mathbf{s}(q)) \leftrightarrow \mathbf{s}(\mathbf{a}(p, q))$.

To prove the desired bisimilarity, it suffices to give a bisimulation relation that contains all the pairs $(\mathbf{a}(p, \mathbf{s}(q)), \mathbf{s}(\mathbf{a}(p, q)))$ for arbitrary closed terms $p$ and $q$. Let $R = \{(\mathbf{a}(p, \mathbf{s}(q)), \mathbf{s}(\mathbf{a}(p, q))) \mid p, q \in \mathcal{C}(T_1)\} \cup \{(p, p) \mid p \in$

https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press

$\mathcal{C}(T_1)\}$. It needs to be proven that all pairs in $R$ satisfy the transfer conditions of Definition 3.1.10 (Bisimilarity). These conditions trivially hold for all pairs $(p, p)$ with $p$ a closed $\Sigma_1$-term. This means that only elements of $R$ of the form $(\mathbf{a}(p, \mathbf{s}(q)), \mathbf{s}(\mathbf{a}(p, q)))$ with $p, q$ in $\mathcal{C}(T_1)$ need to be considered. Assume that $(\mathbf{a}(p, \mathbf{s}(q)), \mathbf{s}(\mathbf{a}(p, q)))$ is such an element.

(i) Assume that $\mathbf{a}(p, \mathbf{s}(q)) \xrightarrow{1} r$ for some $r \in \mathcal{C}(T_1)$. Inspection of the deduction rules given in Example 3.2.1 and the fact that $\mathbf{s}(q) \xrightarrow{1} q$ show that necessarily $r \equiv \mathbf{a}(p, q)$. Since also $\mathbf{s}(\mathbf{a}(p, q)) \xrightarrow{1} \mathbf{a}(p, q)$, the fact that $(\mathbf{a}(p, q), \mathbf{a}(p, q)) \in R$ proves that the first transfer condition is satisfied.

(ii) Assume that $\mathbf{s}(\mathbf{a}(p, q)) \xrightarrow{1} r$ for some $r \in \mathcal{C}(T_1)$. It easily follows from the deduction rules of Example 3.2.1 that $r \equiv \mathbf{a}(p, q)$. Since also $\mathbf{a}(p, \mathbf{s}(q)) \xrightarrow{1} \mathbf{a}(p, q)$ and $(\mathbf{a}(p, q), \mathbf{a}(p, q)) \in R$, also this transfer condition is satisfied.

(iii) Assume that $\mathbf{a}(p, \mathbf{s}(q))\downarrow$. Since $\mathbf{s}(q)$ cannot terminate, this yields a contradiction. Thus, $\mathbf{a}(p, \mathbf{s}(q))$ cannot terminate, which means that the third transfer condition is trivially satisfied.

(iv) Since also $\mathbf{s}(\mathbf{a}(p, q))$ cannot terminate, also the final transfer condition is satisfied, completing the proof.

□

Based on Proposition 2.3.9 (Soundness) and Definitions 2.3.15 (Equivalence classes) and 2.3.18 (Quotient algebra), the above theorem has two immediate corollaries. The second corollary shows that two closed $\Sigma_1$-terms that are derivably equal in theory $T_1$ are bisimilar.

**Corollary 3.2.10 (Soundness)** Let $s$ and $t$ be $\Sigma_1$-terms. If $T_1 \vdash s = t$, then $\mathbb{M}_1 \models s = t$.

**Corollary 3.2.11 (Soundness)** Let $p$ and $q$ be closed $\Sigma_1$-terms. If $T_1 \vdash p = q$, then $p \leftrightarrow q$.

As mentioned, it can be shown that the model $\mathbb{M}_1$ is ground-complete for theory $T_1$, and thus that two bisimilar closed $\Sigma_1$-terms are derivably equal in $T_1$. The proof of this completeness result needs two auxiliary properties. The first one states that a closed $\Sigma_1$-term that corresponds to a termination state is derivably equal to $\mathbf{0}$. The second one states that a closed $\Sigma_1$-term $p$ that can make a step towards another closed term $q$ is derivably equal to $\mathbf{s}(q)$. These two results are crucial in the ground-completeness result because they allow a reduction of the problem to basic $\Sigma_1$-terms as defined in Definition 2.2.10.

https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press

**Lemma 3.2.12** For any closed $\Sigma_1$-term $p$, $p{\downarrow}$ implies that $T_1 \vdash p = \mathbf{0}$.

*Proof* The proof goes via structural induction on $p$. Assume that $p{\downarrow}$.

(i) Assume $p \equiv \mathbf{0}$. Obviously, $T_1 \vdash p = \mathbf{0}$, satisfying the property.

(ii) Assume $p \equiv \mathbf{s}(q)$ for some $q \in \mathcal{C}(T_1)$. Since under this assumption not $p{\downarrow}$, the property follows trivially in this case.

(iii) Assume $p \equiv \mathbf{a}(q,r)$ for some $q, r \in \mathcal{C}(T_1)$. From $p{\downarrow}$ and the deduction rules in Example 3.2.1, it follows that $q{\downarrow}$ and $r{\downarrow}$. Induction gives that $T_1 \vdash q = \mathbf{0}$ and $T_1 \vdash r = \mathbf{0}$. Hence, $T_1 \vdash p = \mathbf{a}(q,r) = \mathbf{a}(\mathbf{0},\mathbf{0}) \overset{\mathrm{PA1}}{=} \mathbf{0}$, completing the proof in this case.

(iv) Assume $p \equiv \mathbf{m}(q,r)$ for some $q, r \in \mathcal{C}(T_1)$. From $p{\downarrow}$ and the deduction rules of Example 3.2.1, it follows that (a) $q{\downarrow}$ or (b) $r{\downarrow}$.

(a) Induction gives that $T_1 \vdash q = \mathbf{0}$. Hence, using Exercise 2.2.8 and the conservativity result of Proposition 2.2.17, $T_1 \vdash p = \mathbf{m}(q,r) = \mathbf{m}(r,q) = \mathbf{m}(r,\mathbf{0}) \overset{\mathrm{PA3}}{=} \mathbf{0}$, completing this case.

(b) Induction gives that $T_1 \vdash r = \mathbf{0}$. Hence, $T_1 \vdash p = \mathbf{m}(q,r) = \mathbf{m}(q,\mathbf{0}) \overset{\mathrm{PA3}}{=} \mathbf{0}$, completing also this final case.

□

**Lemma 3.2.13** For any closed $\Sigma_1$-terms $p$ and $p'$, $p \xrightarrow{1} p'$ implies that $T_1 \vdash p = \mathbf{s}(p')$.

*Proof* The proof goes via structural induction on $p$. Assume that $p \xrightarrow{1} p'$.

(i) Assume $p \equiv \mathbf{0}$. In this case $p \overset{1}{\not\to}$, so the property is satisfied.

(ii) Assume $p \equiv \mathbf{s}(q)$ for some $q \in \mathcal{C}(T_1)$. Since $\mathbf{s}(q) \xrightarrow{1} q$, it follows that $p' \equiv q$. Hence $T_1 \vdash p = \mathbf{s}(q) = \mathbf{s}(p')$, proving the property in this case.

(iii) Assume $p \equiv \mathbf{a}(q,r)$ for some $q, r \in \mathcal{C}(T_1)$. From $p \xrightarrow{1} p'$ and the deduction rules in Example 3.2.1, it follows that (a) $p' \equiv \mathbf{a}(q,r')$ with $r \xrightarrow{1} r'$ or (b) $p' \equiv q'$ with $q \xrightarrow{1} q'$ and $r{\downarrow}$.

(a) Induction gives that $T_1 \vdash r = \mathbf{s}(r')$. Hence, $T_1 \vdash p = \mathbf{a}(q,r) = \mathbf{a}(q,\mathbf{s}(r')) \overset{\mathrm{PA2}}{=} \mathbf{s}(\mathbf{a}(q,r')) = \mathbf{s}(p')$, completing this case.

(b) Induction gives that $T_1 \vdash q = \mathbf{s}(q')$. Furthermore, Lemma 3.2.12 implies that $T_1 \vdash r = \mathbf{0}$. Hence, $T_1 \vdash p = \mathbf{a}(q,r) \overset{\mathrm{PA1}}{=} q = \mathbf{s}(q') = \mathbf{s}(p')$, completing also this case.

https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press

(iv) Assume $p \equiv \mathbf{m}(q, r)$ for some $q, r \in \mathcal{C}(T_1)$. From $p \xrightarrow{1} p'$ and the deduction rules of Example 3.2.1, it follows that $p' \equiv \mathbf{a}(\mathbf{m}(q, r'), q')$ with $q \xrightarrow{1} q'$ and $r \xrightarrow{1} r'$. Induction yields that $T_1 \vdash q = \mathbf{s}(q')$ and $T_1 \vdash r = \mathbf{s}(r')$. Hence, $T_1 \vdash p = \mathbf{m}(q, r) = \mathbf{m}(q, \mathbf{s}(r')) \overset{\text{PA4}}{=} \mathbf{a}(\mathbf{m}(q, r'), q) = \mathbf{a}(\mathbf{m}(q, r'), \mathbf{s}(q')) \overset{\text{PA2}}{=} \mathbf{s}(\mathbf{a}(\mathbf{m}(q, r'), q')) = \mathbf{s}(p')$, completing the proof.

□

**Theorem 3.2.14 (Ground-completeness of $T_1$)** Theory $T_1$ is a ground-complete axiomatization of the model $\mathbb{M}_1$, i.e., for any closed $\Sigma_1$-terms $p$ and $q$, $\mathbb{M}_1 \models p = q$ implies $T_1 \vdash p = q$.

*Proof* Assume that $\mathbb{M}_1 \models p = q$. It must be shown that $T_1 \vdash p = q$. In Example 2.2.13, it has been shown that each closed $\Sigma_1$-term can be written as a basic $\Sigma_1$-term, as defined in Definition 2.2.10. Thus, assume that $p_1$ and $q_1$ are basic $\Sigma_1$-terms such that $T_1 \vdash p = p_1$ and $T_1 \vdash q = q_1$. It follows from Corollary 3.2.11 that $p \leftrightarroweq p_1$ and $q \leftrightarroweq q_1$. Since $\mathbb{M}_1 \models p = q$, also $p \leftrightarroweq q$. The fact that bisimilarity is an equivalence implies that $p_1 \leftrightarroweq q_1$. Assuming that it can be shown that $T_1 \vdash p_1 = q_1$, it follows that $T_1 \vdash p = p_1 = q_1 = q$, which is the desired result.

The proof that $T_1 \vdash p_1 = q_1$ under the assumption that $p_1 \leftrightarroweq q_1$ is a structural-induction proof on the structure of $p_1$. Since $p_1$ is a basic term, only two cases need to be considered.

(i) Assume $p_1 \equiv \mathbf{0}$. It follows both that $T_1 \vdash p_1 = \mathbf{0}$ and that $p_1 \downarrow$. Since $p_1 \leftrightarroweq q_1$, the third transfer condition of Definition 3.1.10 (Bisimilarity) yields that $q_1 \downarrow$. Lemma 3.2.12 yields that $T_1 \vdash q_1 = \mathbf{0}$. Thus, $T_1 \vdash p_1 = \mathbf{0} = q_1$, proving the desired property in this case.

(ii) Assume $p_1 \equiv \mathbf{s}(r)$ for some basic $\Sigma_1$-term $r$. Then, $p_1 \xrightarrow{1} r$. Lemma 3.2.13 yields that $T_1 \vdash p_1 = \mathbf{s}(r)$. Since $p_1 \leftrightarroweq q_1$, the first transfer condition of Definition 3.1.10 (Bisimilarity) yields that $q_1 \xrightarrow{1} s$ for some $s \in \mathcal{C}(T_1)$ such that $r \leftrightarroweq s$. In fact, it follows from the structure of basic terms that $s$ must be a basic $\Sigma_1$-term. Induction yields that $T_1 \vdash r = s$. From $q_1 \xrightarrow{1} s$ and Lemma 3.2.13, it follows that $T_1 \vdash q_1 = \mathbf{s}(s)$. Combining the results obtained so far gives $T_1 \vdash p_1 = \mathbf{s}(r) = \mathbf{s}(s) = q_1$, proving the desired property also in this case.

□

The ground-completeness result has an immediate corollary that proves the claim that two bisimilar closed $\Sigma_1$-terms are derivably equal in $T_1$.

https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press

**Corollary 3.2.15 (Ground-completeness)** Let $p$ and $q$ be closed $\Sigma_1$-terms. If $p \leftrightarrow q$, then $T_1 \vdash p = q$.

Consider again equational theory $T_1$ of Example 2.2.5. In Example 2.2.15, this theory has been extended with an extra binary function symbol **e**, yielding theory $T_2$. It has already been shown that this extension does not influence what equalities between closed terms can be derived from the original theory $T_1$ (Proposition 2.2.17 (Conservative extension)). In the context of the current section, which gives an operational semantics of $T_1$ in terms of transition systems, it is interesting to investigate to what extent the soundness and ground-completeness results for $T_1$ can be used to obtain similar results for the extended theory $T_2$, hopefully reducing the amount of work needed to obtain a ground-complete model for $T_2$.

In Definitions 2.2.16 and 2.2.19, a notion of equational conservativity is introduced. A similar notion can be defined for deduction systems, and it turns out that such an operational conservativity notion is useful in obtaining soundness and ground-completeness results for theories extending some other theory. Note that a deduction system can be extended by extending the signature and/or the deduction rules in question, but also by extending the set of labels that are used in the transitions. Furthermore, observe from Definition 3.2.3 (Transition-system space induced by a deduction system) that only transitions between closed terms over the signature of a deduction system are of interest. Hence, it is not necessary to distinguish two notions of extension and conservativity as in the equational context. Informally, a deduction system conservatively extends another deduction system if and only if it does not add any transitions or terminations with respect to closed terms of the original system compared to those already present in that original system.

**Definition 3.2.16 (Operational conservative extension)** Let $D_1 = (\Sigma_1, R_1)$ over a set of labels $L_1$ and $D_2 = (\Sigma_2, R_2)$ over a set of labels $L_2$ be deduction systems as defined in Definition 3.2.2. Deduction system $D_2$ is an *operational conservative extension* of system $D_1$ if and only if

(i) $\Sigma_2$ contains $\Sigma_1$, $R_2$ contains $R_1$ and $L_2$ contains $L_1$, and

(ii) a transition $p \overset{a}{\rightarrow} q$ or termination $p\downarrow$, with $p$ a closed $\Sigma_1$-term, $a \in L_2$ and $q$ a closed $\Sigma_2$-term, is in the transition-system space induced by $D_1$ exactly when it is in the transition-system space induced by $D_2$ (see Definition 3.2.3). This implies that if $p$ is a $\Sigma_1$-term and $p \overset{a}{\rightarrow} q$, then necessarily $a \in L_1$ and $q$ is a $\Sigma_1$-term.

In order to distinguish the notions of conservative extension for equational

https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press

theories and deduction systems, a conservative extension of equational theories is sometimes called an *equational* conservative extension.

The notion of an operational conservative extension can be used in ground-completeness proofs for equational theories extending some base theory. The conditions of the definition can be weakened in two ways, still achieving these results, but the present definition suffices for most of the applications that are considered in this book. First of all, it is not needed that all transitions are preserved exactly, but only that the resulting transition systems are bisimilar (since the results are used to establish facts about bisimulation models), and secondly, in some restricted circumstances it can be allowed to add new transitions for some old terms.

Under certain assumptions it is possible to prove operational conservativity in a straightforward way, as shown by the theorem given below. The theorem uses the following notion of source-dependency. Informally, the meaning of source-dependency can be explained as follows. The conclusion of a rule in a deduction system defines a transition of its source term. The rule is said to be source-dependent when all the variables in the rule originate from the source of the conclusion.

**Definition 3.2.17 (Source-dependency)** Let $(\Sigma, R)$ be a deduction system. It is defined inductively when a variable in a rule in $R$ is *source-dependent*:

(i) all variables in the source of the conclusion are source-dependent;

(ii) if $s \overset{a}{\to} t$ is a premise of a rule, and all variables in $s$ are source-dependent, then all variables in $t$ are source-dependent.

If all variables in a rule are source-dependent, then the rule is said to be source-dependent.

**Example 3.2.18 (Source-dependency)** Consider the deduction system given in Example 3.2.1. Any of the rules with a termination as its conclusion is source-dependent based on item (i) of the above definition. The rule for **s** is source-dependent because the target of the conclusion contains only variable $x$ which is introduced in the source. The first rule for the **a** function symbol is source-dependent because variables $x$ and $y$ are introduced in the source of the conclusion, implying that they are source-dependent, and variable $y'$ is the target of a premise with only the source-dependent variable $y$ in its source, implying that also $y'$ is source-dependent. Similar reasoning shows that also the other two rules in the deduction system are source-dependent.

https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press

As an example of a rule which is not source-dependent, consider the rule $x{\downarrow}\,/\,\mathbf{0}{\downarrow}$. The variable $x$ appearing in the premise is not source-dependent, as it does not appear in the (source of the) conclusion.

The following theorem is given without proof. The interested reader can find a proof in (Verhoef, 1994).

**Theorem 3.2.19 (Operational conservativity)** Let $D_1 = (\Sigma_1, R_1)$ and $D_2 = (\Sigma_2, R_2)$ be deduction systems such that $\Sigma_2$ contains $\Sigma_1$ and $R_2$ contains $R_1$. Deduction system $D_2$ is an *operational conservative extension* of system $D_1$ if

(i) all rules in $R_1$ are source-dependent, and

(ii) for each added rule $r \in R_2 \backslash R_1$, either the source of the conclusion is not a $\Sigma_1$-term, or the rule has a premise of the form $s \overset{a}{\to} t$ with $s$ a $\Sigma_1$-term such that all variables of $s$ occur in the source of the conclusion and either label $a$ is new or $t$ is not a $\Sigma_1$-term.

The second clause in this definition ensures that each new rule is ‘fresh’, i.e., cannot be applied to old terms. This means that new transitions cannot be added to old terms. When this is done in Chapter 9, Section 9.6, nevertheless, the present theory cannot be used any longer.

**Example 3.2.20 (Operational conservativity)** Consider theory $T_2$ of Example 2.2.15. It is possible to provide an operational semantics for this theory by extending the deduction system for $T_1$ given in Example 3.2.1 with deduction rules for the **e** function symbol. For any $\Sigma_2$-terms $x$, $x'$, $y$, $y'$, and $z$,

$$\frac{x \overset{1}{\to} x',\ y \overset{1}{\to} y',\ \mathbf{e}(x, y') \overset{1}{\to} z}{\mathbf{e}(x, y) \overset{1}{\to} \mathbf{a}(\mathbf{m}(\mathbf{e}(x, y'), x'), z)}$$

and

$$\frac{x{\downarrow},\ y \overset{1}{\to} y'}{\mathbf{e}(x, y){\downarrow}} \qquad \frac{y{\downarrow}}{\mathbf{e}(x, y) \overset{1}{\to} \mathbf{0}}.$$

In particular the first rule might be difficult to understand. It can be clarified by a comparison to the algebra of natural numbers $(\mathbf{N}, +, \times, exp, succ, 0)$, which is an extension of the algebra of Example 2.3.3 with the exponentiation function $exp$. (Note that it is straightforward to show that this algebra is a model of theory $T_2$; see Exercise 2.3.3.) Assume that for any $n, m \in \mathbf{N}$, $exp(n, m)$ is written $n^m$; further assume that multiplication is not explicitly written. In terms of the algebra of natural numbers, the first rule above could be formulated as follows: if $n = 1 + n'$, $m = 1 + m'$, and $n^{m'} = 1 + l$, then $n^m = 1 + (n^{m'}n' + l)$, for $l, n, n', m, m' \in \mathbf{N}$. The following simple calculation

https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press

shows that this is correct: $n^m = n^{1+m'} = nn^{m'} = (1+n')n^{m'} = n^{m'} + n'n^{m'} = 1 + l + n'n^{m'} = 1 + (n^{m'}n' + l)$. Note that the second and third rule can be interpreted in a similar way: if $n = 0$ and $m = 1+m'$ (i.e., $m \neq 0$), then $n^m = 0$, and if $m = 0$, then $n^m = 1 + 0 = 1$ ($n, m, m' \in \mathbf{N}$). (At this point, it may be interesting to have another look at the deduction rules in Example 3.2.1.)

The rules given in this example extend the deduction system given in Example 3.2.1. Since all the deduction rules of Example 3.2.1 are source-dependent (see Example 3.2.18) and since the source of the conclusion in all the extra rules concerns the new function symbol **e**, it follows in a straightforward way from Theorem 3.2.19 (Operational conservativity) that the extended deduction system is an operational conservative extension of the deduction system of Example 3.2.1.

There are close connections between extensions of deduction systems and extensions of equational theories. Theorem 3.2.21 (Conservativity) formalizes this relation and Theorem 3.2.26 (Ground-completeness) given below builds on this result. The two theorems are used several times in the remainder of this book.

The following theorem establishes conditions under which structural operational semantics can be used to determine that one equational theory is a conservative ground-extension of another one.

**Theorem 3.2.21 (Conservativity)** Let $D_1 = (\Sigma_1, R_1)$ and $D_2 = (\Sigma_2, R_2)$ be deduction systems. Let $T_1 = (\Sigma_1, E_1)$ and $T_2 = (\Sigma_2, E_2)$ be equational theories over the same signatures with $E_1$ a subset of $E_2$. Then $T_2$ is an equational conservative ground-extension of $T_1$ if the following conditions are satisfied:

(i) $D_2$ is an operational conservative extension of $D_1$,

(ii) $T_1$ is a ground-complete axiomatization of the bisimulation model induced by $D_1$, and

(iii) $T_2$ is a sound axiomatization of the bisimulation model induced by $D_2$.

**Example 3.2.22 (Conservativity)** It is interesting to see how this theorem can be applied to the two example theories $T_1$ and $T_2$. The aim is to show that $T_2$ is a conservative ground-extension of $T_1$, as defined in Definition 2.2.19. In Examples 3.2.1 and 3.2.20, two deduction systems for $T_1$ and $T_2$ have been given. Example 3.2.20 furthermore has shown that the deduction system for $T_2$ is an operational conservative extension of the deduction system for $T_1$. Theorems 3.2.9 and 3.2.14 show that $T_1$ is a sound and ground-complete axiomatization

https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press

of the algebra of transition systems induced by the term deduction system for $T_1$. Thus, only the third condition in the above theorem needs to be satisfied in order to prove that $T_2$ is an equational conservative extension of $T_1$.

**Example 3.2.23 (Model)** Consider again theory $T_2$ of Example 2.2.15 and its deduction system of Examples 3.2.1 and 3.2.20. It is not difficult to verify that this system is in path format, as defined in Definition 3.2.5. Thus, Theorem 3.2.7 (Congruence theorem) implies that bisimilarity is a congruence on the induced algebra of transition systems. This result, in turn, means that the quotient algebra modulo bisimilarity is well-defined. Let $\mathbb{M}_2$ be this quotient algebra. It turns out that this algebra is a model of $T_2$.

**Theorem 3.2.24 (Soundness of $T_2$)** Algebra $\mathbb{M}_2$ of the previous example is a model of theory $T_2$ using the standard interpretation of constant **0** and functions **s**, **a**, **m**, and **e** explained in Definition 2.3.18 (Quotient algebra).

*Proof* According to Definition 2.3.8 (Model), it must be shown that, for each axiom $s = t$ of $T_2$, $\mathbb{M}_2 \models s = t$. The proof for the axioms already present in $T_1$ carries over directly from the proof of Theorem 3.2.9 (Soundness of $T_1$) and Exercise 3.2.7. The proof for Axioms PA5 and PA6 that are new in $T_2$ is based on the following two bisimulation relations:

$$\{(\mathbf{e}(p, \mathbf{0}), \mathbf{s}(\mathbf{0})) \mid p \in \mathcal{C}(T_2)\} \cup \{(\mathbf{0}, \mathbf{0})\}$$

and

$$\{(\mathbf{e}(p, \mathbf{s}(q)), \mathbf{m}(\mathbf{e}(p, q), p)) \mid p, q \in \mathcal{C}(T_2)\} \cup \{(p, p) \mid p \in \mathcal{C}(T_2)\}.$$

The proof that these two relations are indeed bisimulation relations as defined in Definition 3.1.10 is left as Exercise 3.2.8. □

Combining Theorem 3.2.24 with the observations made in Example 3.2.22 implies that all three conditions of Theorem 3.2.21 (Conservativity) are satisfied. Thus, the following proposition, which is a restatement of Proposition 2.2.17, follows immediately.

**Proposition 3.2.25 (Conservativity)** Theory $T_2$ of Table 2.2 is an equational conservative ground-extension of theory $T_1$ of Table 2.1.

When the extension is of a particular kind, namely when newly introduced syntax can be eliminated from closed terms, then ground-completeness of the extended theory can be established incrementally. This is formulated in the following theorem.

https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press

**Theorem 3.2.26 (Ground-completeness)** Assume that $D_1 = (\Sigma_1, R_1)$ and $D_2 = (\Sigma_2, R_2)$ are deduction systems. Let $T_1 = (\Sigma_1, E_1)$ and $T_2 = (\Sigma_2, E_2)$ be equational theories over the same signatures with $E_1$ a subset of $E_2$. Then $T_2$ is a ground-complete axiomatization of the model induced by $D_2$ if in addition to the conditions of Theorem 3.2.21 (Conservativity) the following condition is satisfied:

(iv) theory $T_2$ has the elimination property with respect to $T_1$, i.e., for each closed $\Sigma_2$-term $p$, there is a closed $\Sigma_1$-term $q$ such that $T_2 \vdash p = q$.

Let us return one more time to the running example of theories $T_1$ of Table 2.1 and $T_2$ of Table 2.2. It has already been argued that these two theories with their deduction systems satisfy the conditions of Theorem 3.2.21 (Conservativity). Proposition 2.2.20 (Elimination) implies that theory $T_2$ has the elimination property with respect to $T_1$. Thus, the following completeness result follows immediately from the above theorem.

**Theorem 3.2.27 (Ground-completeness of $T_2$)** Theory $T_2$ is a ground-complete axiomatization of model $\mathbb{M}_2$, i.e., for any closed $\Sigma_2$-terms $p$ and $q$, $\mathbb{M}_2 \models p = q$ implies $T_2 \vdash p = q$.

It goes without saying that a completeness result based on Theorem 3.2.26 (Ground-completeness), such as the one given for Theorem 3.2.27 (Ground-completeness of $T_2$), for example, is preferable over a completeness result from scratch. Thus, the results given in this section motivate a modular setup of equational theories, starting from some minimal theory and subsequently defining interesting extensions.

### Exercises

3.2.1 Add to the deduction system given in Example 3.2.1 the rule

$$\frac{p \overset{1}{\rightarrow} \mathbf{0}}{\mathbf{s}(p) \overset{ok}{\rightarrow} \mathbf{0}} \; .$$

Show that $\mathbf{s}(\mathbf{a}(\mathbf{0}, \mathbf{0})) \underline{\leftrightarrow} \mathbf{s}(\mathbf{0})$ but $\mathbf{s}(\mathbf{s}(\mathbf{a}(\mathbf{0}, \mathbf{0}))) \not\underline{\leftrightarrow} \mathbf{s}(\mathbf{s}(\mathbf{0}))$. Thus, bisimilarity is not a congruence. Note that the added rule violates the first clause in the definition of the *path* format.

3.2.2 Add to the deduction system given in Example 3.2.1 the axiom

$$\mathbf{m}(x, x) \overset{ok}{\rightarrow} \mathbf{0} \; .$$

Show that $\mathbf{a}(\mathbf{0}, \mathbf{0}) \leftrightarrows \mathbf{0}$ but $\mathbf{m}(\mathbf{a}(\mathbf{0}, \mathbf{0}), \mathbf{0}) \not\leftrightarrows \mathbf{m}(\mathbf{0}, \mathbf{0})$. Thus, bisimilarity is not a congruence. Note that the added rule violates the second clause in the definition of the *path* format.

3.2.3 Add to the deduction system given in Example 3.2.1 the axiom

$$\mathbf{a}(x, \mathbf{a}(y, z)) \xrightarrow{ok} \mathbf{0} \ .$$

Show that $\mathbf{a}(\mathbf{0}, \mathbf{0}) \leftrightarrows \mathbf{0}$ but $\mathbf{a}(\mathbf{0}, \mathbf{a}(\mathbf{0}, \mathbf{0})) \not\leftrightarrows \mathbf{a}(\mathbf{0}, \mathbf{0})$. Thus, bisimilarity is not a congruence. Note that the added rule violates the second clause in the definition of the *path* format.

3.2.4 Add to the deduction system given in Example 3.2.1 the rule

$$\frac{p \xrightarrow{1} x, \ q \xrightarrow{1} x}{\mathbf{a}(p, q) \xrightarrow{ok} \mathbf{0}} \ .$$

Show that $\mathbf{s}(\mathbf{a}(\mathbf{0}, \mathbf{0})) \leftrightarrows \mathbf{s}(\mathbf{0})$ but $\mathbf{a}(\mathbf{s}(\mathbf{0}), \mathbf{s}(\mathbf{a}(\mathbf{0}, \mathbf{0}))) \not\leftrightarrows \mathbf{a}(\mathbf{s}(\mathbf{0}), \mathbf{s}(\mathbf{0}))$. Thus, bisimilarity is not a congruence. Note that the added rule violates the third clause in the definition of the *path* format.

3.2.5 Add to the deduction system given in Example 3.2.1 the rule

$$\frac{x \xrightarrow{1} y}{\mathbf{m}(x, y) \xrightarrow{ok} \mathbf{0}} \ .$$

Show that $\mathbf{s}(\mathbf{a}(\mathbf{0}, \mathbf{0})) \leftrightarrows \mathbf{s}(\mathbf{0})$ but $\mathbf{m}(\mathbf{s}(\mathbf{a}(\mathbf{0}, \mathbf{0})), \mathbf{0}) \not\leftrightarrows \mathbf{m}(\mathbf{s}(\mathbf{0}), \mathbf{0})$. Thus, bisimilarity is not a congruence. Note that the added rule violates the third clause in the definition of the *path* format.

3.2.6 Add to the term deduction system with as only rule $x\downarrow \ / \ \mathbf{0}\downarrow$ a new constant $\mathbf{1}$. Show that this extension is not operationally conservative.

3.2.7 Complete the proof of Theorem 3.2.9 (Soundness of $T_1$) by proving the validity of Axioms PA1, PA3, and PA4.

3.2.8 Complete the proof of Theorem 3.2.24 (Soundness of $T_2$) by showing that the two relations given for Axioms PA5 and PA6 are bisimulation relations as defined in Definition 3.1.10.

## 3.3 Bibliographical remarks

In the literature, the term transition system is used in two different meanings: the meaning that is given here, and in the meaning of a transition-system space. The notion of a transition system is closely related to an automaton in language theory (see e.g. (Linz, 2001)), the notion of a synchronization tree in CCS (see e.g. (Milner, 1980)), or the notion of a process graph (see e.g. (Bergstra & Klop, 1985)). The notion of bisimulation as given here is from (Park, 1981),

https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press

and is closely related to the notion of strong equivalence from CCS (Hennessy & Milner, 1980). For more information on a game characterization of bisimulation, see (Oguztuzun, 1989). For further information on comparative concurrency semantics, see (Van Glabbeek, 1990).

The notion of choice as given in transition systems will, as becomes more clear further on, be subject to influence from outside, from the environment. Non-deterministic choice is a means to show a choice that cannot be influenced. Some theories, like CSP (Hoare, 1985), have two different constructs for the two types of choice.

This chapter gives an introduction to the theory of structural operational semantics (often abbreviated to SOS). This theory finds its origins in the article (Plotkin, 1981), later reprinted in the special issue (Aceto & Fokkink, 2004), where a lot more information on SOS theory can be found. A good overview of SOS theory is (Aceto *et al.*, 2001). The *path* format was introduced in (Baeten & Verhoef, 1993). The material on conservative extensions originates in (Verhoef, 1994). For some recent developments in this area, see (Mousavi *et al.*, 2007).

https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press

https://doi.org/10.1017/CBO9781139195003.007 Published online by Cambridge University Press