6

Control

Whenever a system S_a fails to conform to its specification S_b, in the sense that $S_a \not\preceq S_b$, we may ask if there exists another system S_c, the controller, such that $S_c \times_{\mathcal{I}} S_a \preceq S_b$ or even $S_c \times_{\mathcal{I}} S_a \cong S_b$. In this chapter we discuss these control problems in the behavioral and similarity contexts. We show how to reduce controller synthesis problems from the behavioral context to the similarity context and we solve the later by computing fixed-points of suitably defined operators. In addition to these general control problems we also present fixed-point solutions specialized for safety and reachability control problems that frequently arise in applications.

Notation

For a set Z, Z^* and Z^ω denote the set of all finite and infinite strings, respectively, obtained by concatenating elements in Z. An element $\mathsf{z} \in Z^*$ can thus be seen as a map $\mathsf{z} : \{0, 1, 2, \ldots, n\} \to Z$ represented by $\mathsf{z} = z_0 z_1 z_2 \ldots z_n$ with $\mathsf{z}(i) = z_i$, $i \in \{0, 1, 2, \ldots, n\}$. Similarly, an element $\mathsf{z} \in Z^\omega$ is a map $\mathsf{z} : \mathbb{N}_0 \to Z$ represented by $\mathsf{z} = z_0 z_1 z_2 \ldots$ with $\mathsf{z}(i) = z_i$, $i \in \mathbb{N}_0$. A string $\mathsf{z} \in L \subseteq Z^* \cup Z^\omega$ is said to be maximal if $\mathsf{z} \in Z^\omega$ or if $\mathsf{z} = z_0 z_1 \ldots z_k \in Z^*$ and there exists no string $\mathsf{w} = w_0 w_1 \ldots w_k w_{k+1} \in L$ satisfying $z_i = w_i$ for $i = 0, 1, \ldots, k$.

The natural projection taking $(x_a, x_b) \in X_a \times X_b$ to $x_a \in X_a$ is denoted by $\pi_a : X_a \times X_b \to X_a$. Similarly, $\pi_b : X_a \times X_b \to X_b$ denotes the natural projection taking $(x_a, x_b) \in X_a \times X_b$ to $x_b \in X_b$. The map $\pi_X : X_a \times X_b \times U_a \times U_b \to X_a \times X_b$ is also a projection and sends the quadruple $(x_a, x_b, u_a, u_b) \in X_a \times X_b \times U_a \times U_b$ to the pair $(x_a, x_b) \in X_a \times X_b$. The set of all subsets of Z, also known as the power set of Z, is denoted by 2^Z.

P. Tabuada, *Verification and Control of Hybrid Systems: A Symbolic Approach*,
DOI: 10.1007/978-1-4419-0224-5_6, © Springer Science + Business Media, LLC 2009

6.1 Feedback composition

The notion of system introduced in Part I made no claims regarding the semantics of the set U of inputs. While for some systems, the elements of U that are fed into a system can be suitably chosen, for other systems this choice is not possible. In the literature, two different approaches to the modeling of U coexist. The set of inputs U can be treated as the disjoint union of U_c and U_d, *i.e.*, $U = U_c \uplus U_d$ with U_c modeling the inputs under the designer's control (controllable) and U_d modeling the inputs beyond the designer's control (uncontrollable). Under this paradigm the effect of controllable and uncontrollable inputs is interleaved or turn-based since a transition $x \xrightarrow{u} x'$ will either be labeled by a controllable or by an uncontrollable input u. The other approach consists in describing U as the product $U = C \times D$ with C modeling the control inputs and D modeling the disturbance or adversarial inputs. In this case, starting from a state x and choosing a control input $c \in C$ leads to a transition $x \xrightarrow{c,d} x'$ in which the reached state x' depends on the choice of disturbance input $d \in D$ which is unknown and thus assumed adversarial. In this paradigm, the effect of the control and disturbance is concurrent instead of being interleaved or turn-based. We follow the concurrent approach since this is the natural paradigm for continuous-time control systems and it will be inherited by its finite-state models discussed in Parts III and IV. We do not model disturbance inputs explicitly but rather implicitly through the nondeterminism of the transition relation. This means that the disturbance has the power to decide which c-successor of a state x is reached when a control input c is chosen at the state x.

The notion of controller can be formalized in several different ways. We could regard a controller as a mechanism that determines which input should be fed into the system being controlled based on observed states[1]. This intuitive description has one important limitation: there may be more than one input that leads to a correct or desirable behavior. We thus revise the concept of controller to a mechanism that determines which inputs can be fed to the controlled system based on a sequence of observed outputs. Mathematically, this can be described by a map:

$$\phi : X^* \to 2^U$$

transforming sequences of outputs into sets of inputs. A sequence of transitions:

$$x_0 \xrightarrow{u_0} x_1 \xrightarrow{u_1} x_2 \xrightarrow{u_2} \dots \xrightarrow{u_{n-1}} x_n$$

would then be an internal behavior of the controlled system provided that $u_k \in \phi(x_0 x_1 \dots x_k)$ for every $k \in \{0, 1, \dots, n-1\}$. Although this notion of controller is conceptually very pleasing, for operational reasons we restrict attention, in this chapter, to controllers $\phi : X^* \to 2^U$ that can be described

[1] To simplify the discussion, we assume $Y = X$ and $H = 1_X$.

by a finite-state system S_c. To understand how this can be done we need to digress into feedback composition.

If the effect of applying $\phi : X_a^* \to 2^{U_a}$ to a system S_a is to be described by $S_c \times_{\mathcal{I}} S_a$, we need to elaborate on the kind of interconnection relation that is appropriate for control. Among the several different possibilities we shall require $\mathcal{I}$ to be the extended relation R^e of an alternating simulation relation R from S_c to S_a. This choice renders the results that follow conceptually simple.

Definition 6.1 (Feedback composition). *A system S_c is said to be* feedback composable *with a system S_a if there exists an alternating simulation relation R from S_c to S_a. When S_c is feedback composable with S_a, the feedback composition of S_c and S_a, with interconnection relation $\mathcal{F} = R^e$, is given by $S_c \times_{\mathcal{F}} S_a$.*

The term feedback is justified by the following interpretation of $S_c \times_{\mathcal{F}} S_a$. Assume that $S_c \times_{\mathcal{F}} S_a$ is at the state $(x_c, x_a) \in R$. Controller S_c offers to execute any of the inputs $u_c \in U_c(x_c)$. System S_a responds by selecting any input $u_a \in U_a(x_a)$ satisfying $(x_c, x_a, u_c, u_a) \in \mathcal{F}$ and by taking any transition $x_a \xrightarrow[a]{u_a} x'_a$ labeled by the chosen input u_a. This transition then triggers a matching transition by the controller. This means that S_c measures the new state x'_a of S_a and takes a transition $x_c \xrightarrow[c]{u_c} x'_c$ satisfying $(x'_a, x'_c) \in R$. Existence of the matching transition is guaranteed by the fact that R is an alternating simulation relation. We can thus interpret an internal behavior of $S_c \times_{\mathcal{F}} S_a$ as being the result of a feedback process during which the controller offers a set of inputs, measures the state of S_a, updates its own state, offers again a new set of inputs based on its updated state, and so on. Although it would be more appropriate to use the term state-feedback, given that S_c has access to the states of S_a, we use feedback for brevity. To emphasize this feedback interpretation, the interconnection relation R^e is denoted by $\mathcal{F}$.

The next example illustrates the notion of feedback composition.

Example 6.2. Consider the system S_a displayed in Figure 6.1 and assume that we want to eliminate all the internal behaviors containing transitions of the form $x_{a1} \xrightarrow[a]{\mathtt{a}} x_{a1}$ or containing transitions of the form $x_{a0} \xrightarrow[a]{\mathtt{b}} x_{a0}$. This objective can be achieved by resorting to the controller S_c also represented in Figure 6.1. The required alternating simulation relation is given by:

$$\{(x_{c0}, x_{a0}), (x_{c1}, x_{a1}), (x_{c2}, x_{a2})\}.$$

The feedback composed system $S_c \times_{\mathcal{F}} S_a$ is also depicted in Figure 6.1 and it can be seen that it is equal, up to a relabeling of states and inputs, to S_c. Therefore, the controller enforces the desired requirements on S_a.

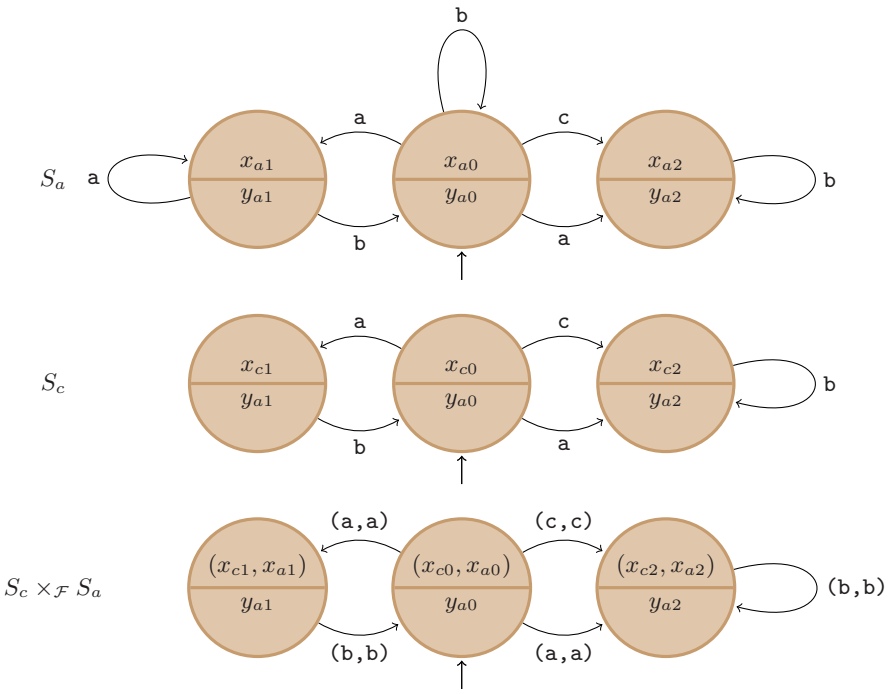

Fig. 6.1. From top to bottom we have: system S_a, controller S_c, and the feedback composed system $S_c \times_{\mathcal{F}} S_a$.

To understand the need to require the existence of an alternating simulation relation from S_c to S_a, let us attempt to use system S_d in Figure 6.2 as a controller. The relation $R = \{(x_{d0}, x_{a0}), (x_{d1}, x_{a1})\}$ is an obvious simulation relation from S_d to S_a but not an alternating simulation relation. Although the composition $S_d \times_{\mathcal{I}} S_a$ is well defined for the interconnection relation:

$$\mathcal{I} = \{(x_d, x_a, u_d, u_a) \in X_d \times X_a \times U_d \times U_a \mid (x_d, x_a) \in R\},$$

the transition $(x_{d0}, x_{a0}) \xrightarrow[da]{a,a} (x_{d1}, x_{a2})$ is not present in $S_d \times_{\mathcal{I}} S_a$ even though it is labeled by the same input as the transition $(x_{d0}, x_{a0}) \xrightarrow[da]{a,a} (x_{d1}, x_{a1})$ which is present in $S_d \times_{\mathcal{I}} S_a$. This means that an implementation of $S_d \times_{\mathcal{I}} S_a$ requires a synchronization procedure between S_d and S_a that is not purely based on inputs. $\triangleleft$

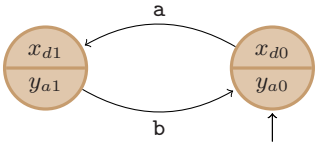

Fig. 6.2. Candidate controller for system S_a in Figure 6.1.

The following result, which is also valid for other forms of composition, explains how feedback composition can restrict the behavior of systems.

Proposition 6.3. *Let S_a and S_b be systems with $Y_a = Y_b$ and let $\mathcal{I}$ be an interconnection relation satisfying:*

$$(x_a, x_b) \in \pi_X(\mathcal{I}) \implies H_a(x_a) = H_b(x_b).$$

Then, the following holds:

- $S_a \times_{\mathcal{I}} S_b \preceq_{\mathcal{S}} S_b$;
- $S_b \times_{\mathcal{I}} S_a \preceq_{\mathcal{S}} S_a$.

Proof. The proof consists in routinely checking that the relations:

$$\{((x_a, x_b), x_b') \in X_{ab} \times X_b \mid x_b = x_b'\}$$
$$\{((x_b, x_a), x_a') \in X_{ba} \times X_a \mid x_a = x_a'\}$$

are simulation relations from $S_a \times_{\mathcal{I}} S_b$ to S_b and from $S_b \times_{\mathcal{I}} S_a$ to S_a, respectively. □

Feedback composition not only restricts the behavior of the system to be controlled but also its initial states. Recall that (x_c, x_a) is an initial state of $S_c \times_{\mathcal{F}} S_a$ if x_c is an initial state of S_c, x_a is an initial state of S_a, and $(x_c, x_a) \in R$. Therefore, it suffices that R does not relate x_a to an initial state of S_c to prevent x_a from being part of an initial state of $S_c \times_{\mathcal{F}} S_a$. The introduced notion of feedback composition thus assumes that the controller has the possibility of initializing S_a. As this assumption may not hold in many situations, we also show how to generalize the results in this chapter to the case where S_a cannot be initialized.

6.2 Safety games

We start by considering a very simple class of control problems whose objective is to design a controller S_c for a system S_a so that $S_c \times_{\mathcal{F}} S_a$ is nonblocking and $\text{Reach}(S_c \times_{\mathcal{F}} S_a) \subseteq W$ for some set $W \subseteq Y_a$. If we regard W as a set of *safe* outputs, the objective of S_c is then to render W invariant for the behaviors in $\mathcal{B}^\omega(S_c \times_{\mathcal{F}} S_a)$, thus keeping the composed system safe. This class

of control problems are termed *safety games* since the controller S_c arises as the solution of a game played against an opponent that tries to prevent the composed system from being safe.

Definition 6.4 (Safety game). *Let S_a be a system with $Y_a = X_a$ and $H_a = 1_{X_a}$, and let $W \subseteq X_a$ be a set of safe states. The* safety game *for system S_a and specification set W asks for the existence of a controller S_c such that:*

1. *S_c is feedback composable with S_a;*
2. *$S_c \times_{\mathcal{F}} S_a$ is nonblocking;*
3. *$\varnothing \neq \mathcal{B}^{\omega}(S_c \times_{\mathcal{F}} S_a) \subseteq W^{\omega}$.*

A safety game is said to be solvable when S_c exists.

The requirement $Y_a = X_a$ and $H_a = 1_{X_a}$ is made without loss of generality since the general case where $Y_a \neq X_a$ can be reduced to this one. We shall elaborate on this fact once we know how to solve safety games. Note that the third requirement in the preceding definition is equivalent to $\mathrm{Reach}(S_c \times_{\mathcal{F}} S_a) \subseteq W$ since a behavior $y_0 y_1 y_2 \ldots$ in W^{ω} necessarily satisfies $y_i \in W$ for every $i \in \mathbb{N}_0$ and vice-versa.

Safety games can be solved by constructing a suitable operator:

$$F_W : 2^{X_a} \to 2^{X_a}$$

for any specification set $W \subseteq X_a$. A fixed-point of this operator provides a collection of states from which it is possible to control system S_a so as to remain in W. The operator F_W:

$$F_W(Z) = \{x_a \in Z \mid x_a \in W \text{ and } \exists u_a \in U_a(x_a) \quad \varnothing \neq \mathrm{Post}_{u_a}(x_a) \subseteq Z\}$$

captures the essence of safety games in the sense that the set $F_W(Z)$ contains all the states $x_a \in Z \cap W$ for which all the u_a-successors of x_a are in Z. The next result shows that a maximal fixed-point of F_W exists and relates the solvability of safety games to fixed-points of F_W.

Proposition 6.5. *Let S_a be a system with $Y_a = X_a$ and $H_a = 1_{X_a}$, and let $W \subseteq X_a$ be a set of safe states. The operator $F_W : 2^X \to 2^X$ satisfies:*

1. *$Z \subseteq Z'$ implies $F_W(Z) \subseteq F_W(Z')$;*
2. *if the safety game for system S_a and specification set W is solvable, then the maximal fixed-point Z of F_W satisfies $Z \cap X_{a0} \neq \varnothing$.*

Proof. The first assertion follows directly from the definition of F_W.

To prove the second assertion, assume that a solution S_c to the safety game exists, let K be the set of all states reachable in $S_c \times_{\mathcal{F}} S_a$, and let $Z' = \mathrm{Reach}(S_c \times_{\mathcal{F}} S_a)$. Note that $K \subseteq X_a \times X_b$ while $\pi_a(K) = Z' \subseteq X_a$. We claim that $X_{a0} \cap Z' \neq \varnothing$ and $Z' \subseteq F(Z')$. The first claim is proved by noting

that the second and third requirement in the definition of safety game imply $K \cap X_{ca0} \neq \varnothing$ and thus:

$$Z' \cap X_{a0} = \pi_a(K) \cap \pi_a(X_{ca0}) \supseteq \pi_a(K \cap X_{ca0}) \neq \varnothing.$$

The second claim can be proved as follows. Let $x_a \in Z'$ and let $x_c \in X_c$ be such that $(x_c, x_a) \in K$. Since state (x_c, x_a) is reachable in $S_c \times_{\mathcal{F}} S_a$ and since S_c is a solution to the safety game, there must exist $(u_c, u_a) \in U_{ca}(x_c, x_a)$ such that $(x_c, x_a) \xrightarrow[ca]{u_c, u_a} (x'_c, x'_a)$ with $x'_a \in W$. Moreover, $(x'_c, x'_a) \in K$ which implies $x'_a \in Z'$. We now invoke the definition of feedback composition to conclude that every transition $x_a \xrightarrow[a]{u_a} x''_a$ in S_a labeled by the same input u_a gives rise to a transition $(x_c, x_a) \xrightarrow[ca]{u_c, u_a} (x''_c, x''_a)$ in $S_c \times_{\mathcal{F}} S_a$. Necessarily, $(x''_c, x''_a) \in K$ and thus $x''_a \in Z'$. Hence, we conclude the existence of an input $u_a \in U_a(x_a)$ for which $\varnothing \neq \mathrm{Post}_{u_a}(x_a) \subseteq Z'$. According to the definition of F_W, $x_a \in F_W(Z')$ and the second claim is proved. Finally, by definition of F_W we always have $F_W(Z') \subseteq Z'$ so that $F_W(Z') = Z'$ and Z' is a fixed-point of F_W. Since the second assertion in the proposition holds for any fixed-point Z' of F_W, it also holds for its maximal fixed-point whose existence is a consequence of the first assertion in the proposition. □

A controller solving a safety game with specification set W can always be constructed from the information contained in a fixed-point Z of F_W satisfying $Z \cap X_{a0} \neq \varnothing$. One possibility is the controller:

$$S_c = (X_c, X_{c0}, U_a, \xrightarrow[c]{}) \tag{6.1}$$

defined by:

- $X_c = Z$;
- $X_{c0} = Z \cap X_{a0}$;
- $x_c \xrightarrow[c]{u_a} x'_c$ if $\varnothing \neq \mathrm{Post}_{u_a}(x_c) \subseteq Z$,

and where $\mathrm{Post}_{u_a}(x_c)$ refers to the u_a-successors in S_a. It is a simple matter to check that the relation defined by all the pairs $(x_c, x_a) \in X_c \times X_a$ with $x_c = x_a$ is an alternating simulation relation from S_c to S_a. According to Proposition 6.3, $S_c \times_{\mathcal{F}} S_a \preceq_{\mathcal{S}} S_c$ and we can interpret the result of composing S_c with S_a as the elimination of all the transitions labeled by inputs for which the corresponding successor sets are not contained in Z. Intuitively, S_c forces the behavior of $S_c \times_{\mathcal{F}} S_a$ to remain in $Z^\omega \subseteq W^\omega$.

A complete characterization of the solutions to safety games can now be obtained by noting that it follows from the results in the Appendix that the maximal fixed-point of F_W can be obtained by iterating F_W.

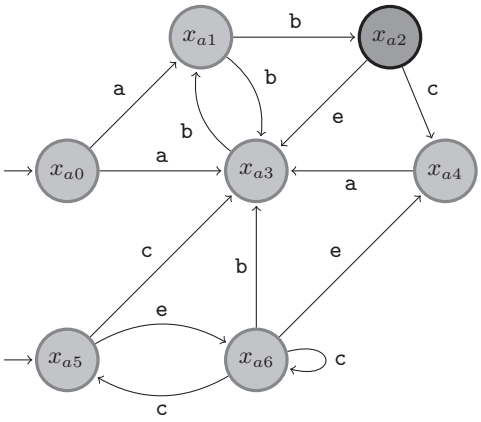

Fig. 6.3. System S_a for Example 6.7.

Theorem 6.6. *Let S_a be a system with $Y_a = X_a$ and $H_a = 1_{X_a}$, and let $W \subseteq X_a$ be a set of safe states. The safety game for system S_a and specification set W is solvable iff the maximal fixed-point Z of the operator F_W satisfies $Z \cap X_{a0} \neq \varnothing$. Moreover, Z can be obtained as:*

$$Z = \lim_{i \to \infty} F_W^i(X_a).$$

When $Z \cap X_{a0} \neq \varnothing$, a solution to the safety game is given by the controller (6.1).

Example 6.7. To illustrate Theorem 6.6 consider the finite-state system S_a in Figure 6.3 and let W be the set of all light-colored states. The maximal fixed-point of F_W can be obtained by iterating F_W and the result of this iteration is shown in Figure 6.4.

After 5 iterates of F_W a fixed-point is reached. The resulting set Z defines a controller that restricts the inputs to e at the state x_{a5} and to c at state x_{a6}. The reader can verify that this choice of inputs prevents the behavior of S_a to leave W^ω. The feedback composition of S_c with S_a, displayed in Figure 6.5, results in a finite-state system equal to S_c if we identify the states (x_{a5}, x_{a5}) and (x_{a6}, x_{a6}) with the states x_{a5} and x_{a6}, and if we identify the inputs (e, e) and (c, c) with the inputs e and c, respectively. $\lhd$

The controller (6.1) is completely determined by a given fixed-point of F_W. When we use the maximal fixed-point of F_W, (6.1) becomes the best possible controller in the sense that any other controller solving the same safety problem would be more restrictive.

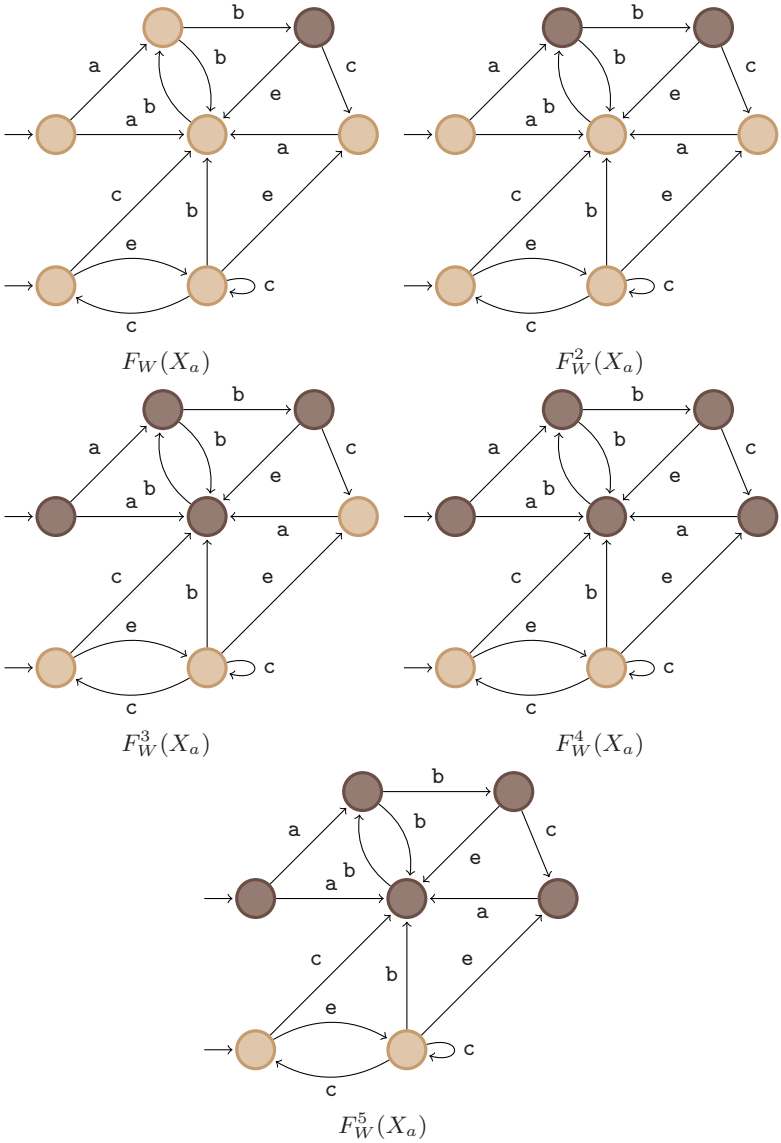

Fig. 6.4. Iterates of F_W. Dark-colored states correspond to the image of F_W.

Proposition 6.8. *Let S_a be a system with $Y_a = X_a$ and $H_a = 1_{X_a}$, and let $W \subseteq X_a$ be a set of safe states. For any controller S_d solving the safety game for system S_a and specification set W we have:*

$$S_d \times_{\mathcal{G}} S_a \preceq_{\mathcal{S}} S_c \times_{\mathcal{F}} S_a$$

where S_c is the controller (6.1) defined by the maximal fixed-point of F_W.

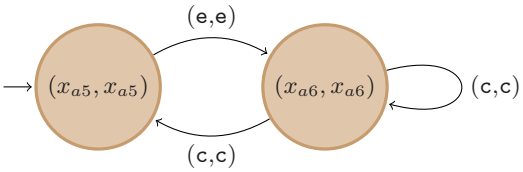

Fig. 6.5. System $S_c \times_{\mathcal{F}} S_a$ for Example 6.7.

Proof. It was shown in the proof of Proposition 6.5 that for any controller S_d solving the safety game, the set $Z' = \text{Reach}(S_d \times_{\mathcal{G}} S_a)$ is a fixed-point of F_W. Therefore, $Z' \subseteq Z$ where Z is the maximal fixed-point of F_W. This suggests that the relation:

$$R = \{((x_d, x'_a), (x_c, x_a)) \in X_{da} \times X_{ca} \mid x'_a = x_a\}$$

is a simulation relation from $S_d \times_{\mathcal{G}} S_a$ to $S_c \times_{\mathcal{F}} S_a$. The proof consists in showing that this is indeed the case. Before starting we note that by definition of S_c and since $H_a = 1_{X_a}$, $(x_c, x_a) \in X_{ca}$ iff $x_c = x_a$.

Consider the first requirement in Definition 4.7 and let $(x_{d0}, x_{a0}) \in X_{da0}$. Then, $x_{a0} \in Z' \subseteq Z$ and $x_{a0} \in X_{a0}$. By definition of S_c and by definition of feedback composition, $(x_{a0}, x_{a0}) \in X_{ca0}$. Consequently, the pair $((x_{d0}, x_{a0}), (x_{a0}, x_{a0}))$ belongs to R.

The second requirement follows directly from the definition of R.

To prove the third requirement let $((x_d, x_a), (x_a, x_a)) \in R$ and assume that $(x_d, x_a) \xrightarrow[da]{u_d, u_a} (x'_d, x'_a)$ in $S_d \times_{\mathcal{G}} S_a$. Since S_d is a controller for the safety problem we necessarily have $\varnothing \neq \text{Post}_{u_a}(x_a) \subseteq Z' \subseteq Z$. But by definition of S_c, for every such input u_a we have $u_a \in U_c(x_a)$. Therefore, $(x_a, x_a) \xrightarrow[ca]{u_a, u_a} (x'_a, x'_a)$ in $S_c \times_{\mathcal{F}} S_a$ and by definition of R, $((x'_d, x'_a), (x'_a, x'_a)) \in R$. □

Theorem 6.6 can be generalized to the case where the initial state of S_a cannot be initialized. The modification amounts to replace the condition $Z \cap X_{a0} \neq \varnothing$, which requires the existence of at least one initial state from which S_c can operate, to $X_{a0} \subseteq Z$, which requires that S_c can operate from every initial state in X_{a0}.

The apparently more general problem of synthesizing a controller S_c to enforce $\mathcal{B}^\omega(S_c \times_{\mathcal{F}} S_a) \subseteq W^\omega$ when $Y_a \neq X_a$ and $H_a \neq 1_{X_a}$ can be reduced to the problem studied in this section. It suffices to consider a new safe set $W' \subseteq X_a$ defined by $W' = H_a^{-1}(W)$ and to apply Theorem 6.6 to system $(X_a, X_{a0}, U_a, \xrightarrow[a]{}, X_a, 1_{X_a})$ and specification set W'.

6.3 Reachability games

While the objective of safety games is to keep the behaviors of the composed system within a safe set, reachability games ask for a certain set W of outputs to be reached. As in the previous section we consider only the case where $H_a = 1_{X_a}$ since the general case can be reduced to this one by suitably redefining W.

Definition 6.9 (Reachability game). *Let S_a be a system satisfying $Y_a = X_a$ and $H_a = 1_{X_a}$, and let $W \subseteq X_a$ be a set of states. The* reachability game *for system S_a and specification set W asks for the existence of a controller S_c such that:*

1. *S_c is feedback composable with S_a;*
2. *for every maximal behavior $\mathsf{y} \in \mathcal{B}(S_c \times_{\mathcal{F}} S_a) \cup \mathcal{B}^{\omega}(S_c \times_{\mathcal{F}} S_a)$ there exists $k \in \mathbb{N}_0$ such that $\mathsf{y}(k) = y_k \in W$.*

A reachability game is said to be solvable when S_c exists.

The second condition in the definition of reachability game requires that any infinite behavior $\mathsf{y} = y_0 y_1 \ldots$ of $S_c \times_{\mathcal{F}} S_a$ visits the set W in finite time. Moreover, it also requires that any finite behavior $y_0 y_1 \ldots y_l$ that cannot be extended to an infinite behavior, visits W before or when reaching a blocking state. In particular, no nonblocking condition is imposed since the objective is simply to reach W in finitely many steps. Once states in W are reached, no further requirements are imposed by the reachability game. More demanding requirements, such as reaching a set of states W and remaining within W thereafter, can be obtained by combining safety with reachability.

Similarly to safety games, reachability games can also be given a fixed-point characterization. For any $W \subseteq X_a$ we can define the operator:

$$G_W : 2^{X_a} \to 2^{X_a}$$

by:

$$G_W(Z) = \{x_a \in X_a \mid x_a \in W \text{ or } \exists u_a \in U_a(x_a) \quad \varnothing \neq \mathrm{Post}_{u_a}(x_a) \subseteq Z\}.$$

It is not difficult to see that for any $W \subseteq X_a$, the inclusion $Z \subseteq Z'$ implies $G_W(Z) \subseteq G_W(Z')$, thus guaranteeing the existence of a unique minimal fixed-point of G_W. Several different controllers solving the reachability game can be constructed from a fixed-point Z of G_W for which $Z \cap X_{a0} \neq \varnothing$.

Among the several possible solutions, we consider the controller:

$$S_c = (X_c, X_{c0}, U_a, \underset{c}{\longrightarrow}) \tag{6.2}$$

defined as:

- $X_c = Z$;
- $X_{c0} = Z \cap X_{a0}$;
- $x_c \xrightarrow[c]{u_a} x_c'$ if there exists a $k \in \mathbb{N}$ such that $x_c \notin G_W^k(\varnothing)$ and $\varnothing \neq \mathrm{Post}_{u_a}(x_c) \subseteq G_W^k(\varnothing)$,

and where $\mathrm{Post}_{u_a}(x_c)$ refers to the u_a-successors in S_a. Moreover, one can easily verify that the relation defined by all the pairs $(x_c, x_a) \in X_c \times X_a$ with $x_c = x_a$ is an alternating simulation relation from S_c to S_a. Similarly to safety games, the solution of reachability games can be fully characterized in terms of the fixed-points of G_W.

Theorem 6.10. *Let S_a be a system with $Y_a = X_a$ and $H_a = 1_{X_a}$, and let $W \subseteq X_a$ be a set of states. The reachability game for S_a and specification set W is solvable iff the minimal fixed-point Z of the operator G_W satisfies $Z \cap X_{a0} \neq \varnothing$. Moreover, Z can be obtained as:*

$$Z = \lim_{i \to \infty} G_W^i(\varnothing).$$

When $Z \cap X_{a0} \neq \varnothing$, a solution to the reachability game is given by the controller (6.2).

Example 6.11. Consider again the finite-state system in Figure 6.3 and assume that W consists of the single state x_{a4}. The computation of the fixed-point of G_W by iteration is presented in Figure 6.6. The resulting controller (6.2) is displayed in Figure 6.7. Note that state x_{a2} is not helpful for this particular problem since it is not reachable. However, it may be useful when this set of states, corresponding to the minimal fixed-point of G_W, is used as the starting point for further design problems. $\lhd$

For reachability games there is no optimal controller in the sense of Proposition 6.8. This observation is illustrated in Example 6.12.

Example 6.12. Consider the system S_a in Figure 6.8 where the set W consists of the state x_2. Let S_c be any finite-state controller solving the reachability problem for system S_a and let k be the maximum number of times[2] that an internal behavior of $S_c \times_{\mathcal{F}} S_a$ visits the state x_1 before reaching the state x_2. We can always construct a controller S_d that allows the internal behaviors of $S_d \times_{\mathcal{G}} S_a$ to visit x_1 any number of times smaller than or equal to $k+1$ before reaching x_2. Clearly, S_d is less restrictive than S_c which shows that a minimally restrictive controller does not exist. $\lhd$

[2] Such number exists since both S_c and S_a are finite-state systems.

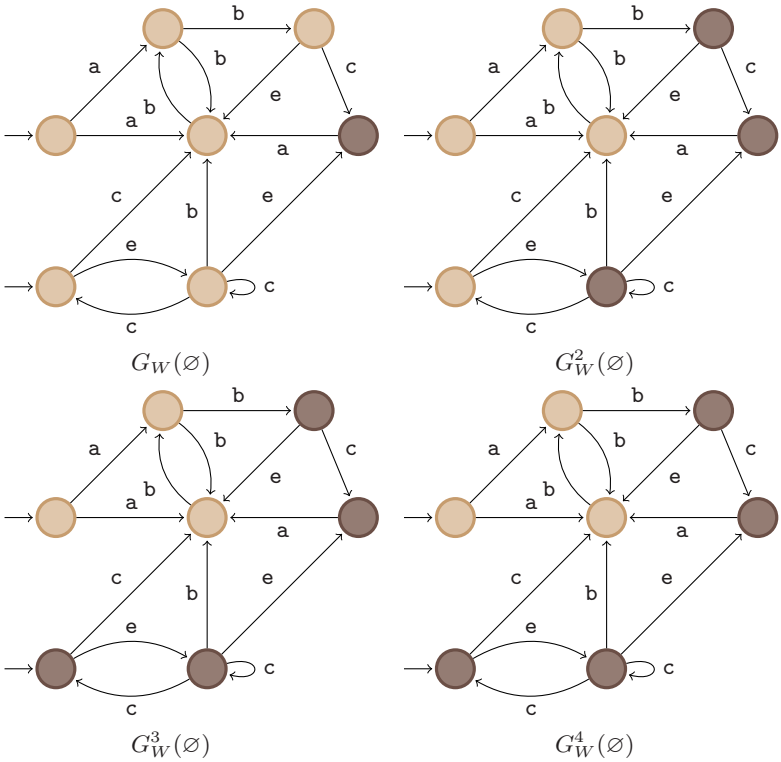

Fig. 6.6. Iterates of G_W. Dark-colored states correspond to the image of G_W.

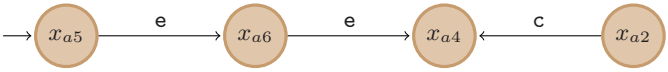

Fig. 6.7. Controller S_c for Example 6.11.

Analogously to safety games, Theorem 6.10 can be generalized to the case where the initial states of S_a cannot be initialized by the controller. This generalization consists in replacing $Z \cap X_{a0} \neq \varnothing$ with the stronger condition $X_{a0} \subseteq Z$ guaranteeing that no initial state of S_a is eliminated in the composition with the controller.

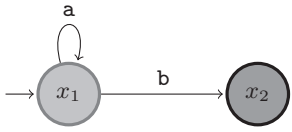

Fig. 6.8. System S_a for Example 6.12. The set W consists of the state x_2.

6.4 Behavioral games

Safety games are special instances of behavioral games. If S_b is a system such that $\mathcal{B}^\omega(S_b) = W^\omega$, then the safety game specified by S_a and W can be reformulated as the existence of a controller S_c such that $S_c \times_\mathcal{F} S_a \preceq_\mathcal{B} S_b$.

Definition 6.13 (Behavior inclusion game). *Let S_a be a system and let S_b be a system specification satisfying $Y_b = Y_a$. The* behavior inclusion game *for system S_a and specification system S_b asks for the existence of a controller S_c such that:*

1. *S_c is feedback composable with S_a;*
2. *$S_c \times_\mathcal{F} S_a$ is nonblocking;*
3. *$S_c \times_\mathcal{F} S_a \preceq_\mathcal{B} S_b$.*

A behavior inclusion game is said to be solvable when S_c exists.

In Chapter 4 we saw that under reasonable assumptions the specification system is output deterministic. Therefore, by Proposition 4.11, the third requirement in the definition of behavior inclusion games can be converted to $S_c \times_\mathcal{F} S_a \preceq_\mathcal{S} S_b$. Replacing behavior inclusion with simulation leads to similarity games which are discussed in detail in the next section. Behavioral games, where the stronger requirement $S_c \times_\mathcal{F} S_a \cong_\mathcal{B} S_b$ is to be enforced, can also be transformed into similarity games with the requirement $S_c \times_\mathcal{F} S_a \cong_\mathcal{S} S_b$. These games are also discussed in the next section.

6.5 Similarity games

The controller synthesis problem in a similarity context is called a *simulation game*.

Definition 6.14 (Simulation game). *Let S_a be a system and let S_b be a system specification satisfying $Y_b = Y_a$. The* simulation game *for system S_a and specification system S_b asks for the existence of a controller S_c such that:*

1. *S_c is feedback composable with S_a;*
2. *$S_c \times_\mathcal{F} S_a$ is nonblocking;*
3. *$S_c \times_\mathcal{F} S_a \preceq_\mathcal{S} S_b$.*

A simulation game is said to be solvable when S_c exists.

Simulation games can be solved by using an extension of the operator F introduced in Chapter 5. The operator F_C:

$$F_C : 2^{X_a \times X_b} \to 2^{X_a \times X_b}$$

in which the subscript C refers to control, is defined by $(x_a, x_b) \in F_C(W)$, for

some $W \subseteq X_a \times X_b$, if the following three conditions are satisfied:

1. $H_a(x_a) = H_b(x_b)$;
2. $(x_a, x_b) \in W$;
3. there exists $u_a \in U_a(x_a)$ such that for every $x'_a \in \mathrm{Post}_{u_a}(x_a)$ there exists a transition $x_b \xrightarrow[b]{u_b} x'_b$ in S_b with $(x'_a, x'_b) \in W$.

As before, $Z \subseteq Z'$ implies $F_C(Z) \subseteq F_C(Z')$ so that F_C has a unique maximal fixed-point which can be used to construct a solution to the simulation game whenever $Z \cap (X_{a0} \times X_{b0}) \neq \varnothing$. In this case we can define the controller:

$$S_c = (X_c, X_{c0}, U_a, \xrightarrow[c]{}, Y_a, H_c) \tag{6.3}$$

by:

- $X_c = Z$;
- $X_{c0} = Z \cap (X_{a0} \times X_{b0})$;
- $(x_a, x_b) \xrightarrow[c]{u_a} (x'_a, x'_b)$ in S_c if the following three conditions hold:
  1. $(x'_a, x'_b) \in Z$;
  2. $x_b \xrightarrow[b]{u_b} x'_b$ in S_b for some $u_b \in U_b(x_b)$;
  3. $x_a \xrightarrow[a]{u_a} x'_a$ in S_a for some $u_a \in U_a(x_a)$ such that for all $x''_a \in \mathrm{Post}_{u_a}(x_a)$ there exists a transition $x_b \xrightarrow[b]{u'_b} x''_b$ in S_b with $(x''_a, x''_b) \in Z$;
- $H_c(x_a, x_b) = H_a(x_a)$.

The reader should verify that the definition of F_C ensures that the relation:

$$R = \{((x_a, x_b), x'_a) \in Z \times X_a \mid x_a = x'_a\}$$

is an alternating simulation relation from S_c to S_a.

The previous discussion can be summarized in the following result characterizing the solution to behavior inclusion games.

Theorem 6.15. *Let S_a be a system and let S_b be a system specification with $Y_b = Y_a$. The simulation game for system S_a and specification system S_b is solvable iff the maximal fixed-point Z of the operator F_C satisfies $Z \cap (X_{a0} \times X_{b0}) \neq \varnothing$. Moreover, Z can be obtained as:*

$$Z = \lim_{i \to \infty} F_C^i(X_a \times X_b).$$

When $Z \cap (X_{a0} \times X_{b0}) \neq \varnothing$, a solution to the simulation game is given by the controller (6.3).

Proof. It was already shown, by defining explicitly the controller S_c in (6.3), that existence of a fixed-point Z of F_C satisfying $Z \cap (X_{a0} \times X_{b0}) \neq \varnothing$ leads to a solution of the simulation game.

The converse implication can be proved by noting that from any controller S_c solving the simulation game and from the corresponding simulation relation R from $S_c \times_{\mathcal{F}} S_a$ to S_b we can construct a relation $R' \subseteq X_a \times X_b$ defined by $(x_a, x_b) \in R'$ if there exists $x_c \in X_c$ such that $((x_c, x_a), x_b) \in R$. It is now simple to verify that R' is a fixed-point of F_C. The crucial inclusion is $R' \subseteq F_C(R')$ and the key step is to show that any $(x_a, x_b) \in R'$ satisfies the third requirement in the definition of F_C. We focus on this step. Let $(x_a, x_b) \in R'$ and recall that by definition of R' there exists $x_c \in X_c$ such that $((x_c, x_a), x_b) \in R$. Since $S_c \times_{\mathcal{F}} S_a$ is nonblocking, there exists an input $(u_c, u_a) \in U_{ca}(x_c, x_a)$. Moreover, it follows from the definition of feedback composition, that for every $x'_a \in \mathrm{Post}_{u_a}(x_a)$ there exists a transition $(x_c, x_a) \xrightarrow[ca]{u_c, u_a} (x'_c, x'_a)$ in $S_c \times_{\mathcal{F}} S_a$. But as R is a simulation relation from $S_c \times_{\mathcal{F}} S_a$ to S_b, for every transition $(x_c, x_a) \xrightarrow[ca]{u_c, u_a} (x'_c, x'_a)$ in $S_c \times_{\mathcal{F}} S_a$ there exists a transition $x_b \xrightarrow{u'_b} x'_b$ in S_b satisfying $((x'_c, x'_a), x'_b) \in R$. We thus conclude the existence of $u_a \in U_a(x_a)$ such that for every $x'_a \in \mathrm{Post}_{u_a}(x_a)$ there exists a transition $x_b \xrightarrow{u'_b} x'_b$ in S_b satisfying $(x'_a, x'_b) \in R'$ which is precisely the third requirement in the definition of F_C. □

Example 6.16. To illustrate the construction of S_c we revisit the models for the bus fare machine used in Example 4.3 and displayed in Figure 4.1 and Figure 4.2. Although $S_a \cong_{\mathcal{B}} S_b$, system S_a is not simulated by system S_b. We thus seek a controller solving the simulation game for system S_a and specification system S_b. The iteration of F_C starts with the set $X_a \times X_b$ and terminates with the fixed-point Z after two iterations.

$$
\begin{aligned}
F_C^0(X_a \times X_b) = \{&(x_{a0}, x_{b0}), (x_{a0}, x_{b1}), (x_{a0}, x_{b2}), (x_{a0}, x_{b3}), (x_{a1}, x_{b0}),\\
&(x_{a1}, x_{b1}), (x_{a1}, x_{b2}), (x_{a1}, x_{b3}), (x_{a2}, x_{b0}), (x_{a2}, x_{b1}),\\
&(x_{a2}, x_{b2}), (x_{a2}, x_{b3})\},\\
F_C^1(X_a \times X_b) = \{&(x_{a0}, x_{b0}), (x_{a1}, x_{b1}), (x_{a1}, x_{b3}), (x_{a2}, x_{b2})\},\\
F_C^2(X_a \times X_b) = \{&(x_{a0}, x_{b0}), (x_{a1}, x_{b1}), (x_{a1}, x_{b3}), (x_{a2}, x_{b2})\}.
\end{aligned}
$$

The corresponding controller S_c is displayed in Figure 6.9 and $S_c \times_{\mathcal{F}} S_a$ is shown in Figure 6.10. The simulation relation from $S_c \times_{\mathcal{F}} S_a$ to S_b is given by the fixed-point Z of F_C. Note that the state (x_{a1}, x_{b3}), albeit not reachable, can be useful when $S_c \times_{\mathcal{F}} S_a$ is the starting point for further designs. ◁

The operator F_C can be seen as a control generalization of the operator F defined in Chapter 5. If there exists a simulation relation from system S_a to system S_b, then the maximal fixed-point F_C coincides with the maximal fixed-point of F. However, if no simulation relation from S_a to S_b exists, then

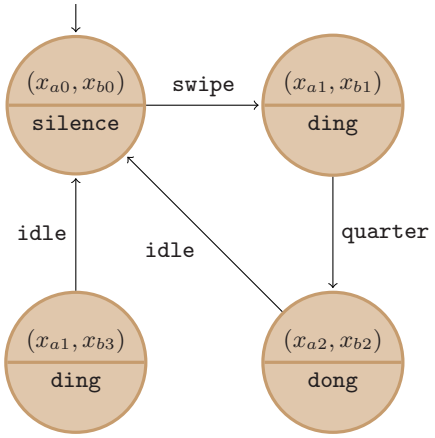

Fig. 6.9. Controller S_c for Example 6.16.

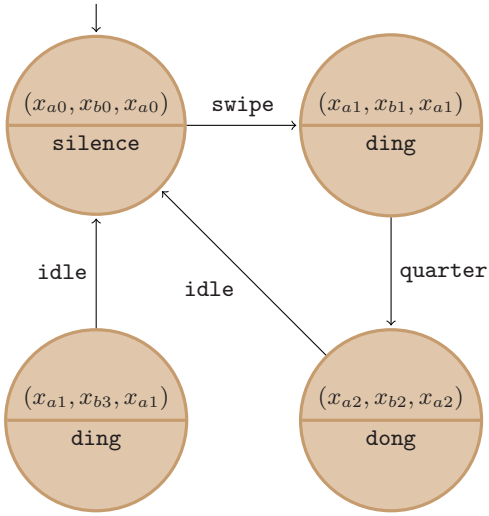

Fig. 6.10. Composed system $S_c \times_{\mathcal{F}} S_a$ for Example 6.16.

$X_{a0} \not\subseteq \pi_a(Z \cap (X_{a0} \times X_{b0}))$ for the maximal fixed-point Z of F. In contrast, the maximal fixed-point of F_C provides a simulation relation from a restricted version of S_a to S_b whenever a solution for the simulation game exists. Such restricted version can then be constructed as $S_c \times_{\mathcal{F}} S_a$ for a controller S_c.

The iteration of the operator F_C provides an algorithm which is guaranteed to terminate in time polynomial in $|X_a||X_b|$ for finite-state systems. Moreover, the controller S_c constructed from the maximal fixed-point of F_C in (6.3) is optimal in the sense that it minimally restricts S_a in order to enforce the specification.

Proposition 6.17. *Let S_a be a system and let S_b be a system specification with $Y_b = Y_a$. For any controller S_d solving the simulation game for system S_a and specification system S_b we have:*

$$S_d \times_{\mathcal{G}} S_a \preceq_{\mathcal{S}} S_c \times_{\mathcal{F}} S_a$$

where S_c is the controller (6.3) defined by the maximal fixed-point of F_C.

Proof. It was shown in the proof of Theorem 6.15 that any controller S_d solving the simulation game leads to a fixed-point Z' of F_C. Moreover, S_c is completely determined by the maximal fixed-point Z of F_C. If we denote by R and R' the simulation relations from $S_c \times_{\mathcal{F}} S_a$ and $S_d \times_{\mathcal{G}} S_a$, respectively, to S_b, it follows from the maximality of Z that the relation defined by the pairs $((x_d, x'_a), (x_c, x_a)) \in X_{da} \times X_{ca}$ for which there exists a state $x_b \in X_b$ such that $((x_d, x'_a), x_b) \in R'$ and $((x_c, x_a), x_b) \in R$ is the desired simulation relation from $S_d \times_{\mathcal{G}} S_a$ to $S_c \times_{\mathcal{F}} S_a$. The rest of the proof consists in routinely checking that all the requirements in Definition 4.7 are satisfied and is left to the reader. □

Theorem 6.15 assumes that the initial states of S_a can be initialized by the controller. When this is not the case we need to replace $Z \cap (X_{a0} \times X_{b0}) \neq \varnothing$ with $X_{a0} = \pi_a(Z \cap (X_{a0} \times X_{b0}))$ in Theorem 6.15 to ensure S_c can operate from any initial state of S_a.

The more demanding bisimulation games require the composed system $S_c \times_{\mathcal{F}} S_a$ to be bisimilar to the specification.

Definition 6.18 (Bisimulation game). *Let S_a be a system and let S_b be a system specification satisfying $Y_b = Y_a$. The* bisimulation game *for system S_a and specification system S_b asks for the existence of a controller S_c such that:*

1. *S_c is feedback composable with S_a;*
2. *$S_c \times_{\mathcal{F}} S_a \cong_{\mathcal{S}} S_b$.*

A simulation game is said to be solvable when S_c exists.

Note that no nonblocking requirement is imposed on $S_c \times_{\mathcal{F}} S_a$ since a state in $S_c \times_{\mathcal{F}} S_a$ is blocking iff it is bisimulated by a blocking state in S_b. Hence, the existence or absence of blocking states is completely determined by the specification S_b.

Before molding bisimulation games into a fixed-point computation we make two observations. First, if there exists a controller S_c rendering $S_c \times_{\mathcal{F}} S_a$ bisimilar to S_b, it follows that $S_b \preceq_{\mathcal{S}} S_a$ since $S_b \cong_{\mathcal{S}} S_c \times_{\mathcal{F}} S_a \preceq_{\mathcal{S}} S_a$ in virtue of Proposition 6.3. The second observation notes that for any input $u_a \in U_a(x_a)$ enabled by the controller S_c and for any $x'_a \in \mathrm{Post}_{u_a}(x_a)$ there must exist a matching transition $x_b \xrightarrow[b]{u_b} x'_b$ in S_b.

The preceding observations motivate the definition of the operator:

$$G_C : 2^{X_a \times X_b} \to 2^{X_a \times X_b}$$

given by $(x_a, x_b) \in G_C(W)$, for some $W \subseteq X_a \times X_b$, if the following three conditions are satisfied:

1. $H_a(x_a) = H_b(x_b)$;
2. $(x_a, x_b) \in W$;
3. for every transition $x_b \xrightarrow[b]{u_b} x'_b$ in S_b there exists an input $u_a \in U_a(x_a)$ satisfying:
   a) there exists $x'_a \in \mathrm{Post}_{u_a}(x_a)$ with $(x'_a, x'_b) \in W$;
   b) for every $x''_a \in \mathrm{Post}_{u_a}(x_a)$ there exists a transition $x_b \xrightarrow[b]{u'_b} x''_b$ in S_b with $(x''_a, x''_b) \in W$.

A controller based on a fixed-point Z of G_C can be constructed whenever $\pi_b(Z \cap (X_{a0} \times X_{b0})) = X_{b0}$. Under this assumption, one possible controller is:

$$S_c = (X_c, X_{c0}, U_a, \xrightarrow[c]{}, Y_a, H_c) \tag{6.4}$$

defined by:

- $X_c = Z$;
- $X_{c0} = Z \cap (X_{a0} \times X_{b0})$;
- for every transition $x_b \xrightarrow[b]{u_b} x'_b$ in S_b, $(x_a, x_b) \xrightarrow[c]{u_a} (x'_a, x'_b)$ in S_c if the following two conditions hold:
  1. $(x'_a, x'_b) \in Z$;
  2. $x_a \xrightarrow[a]{u_a} x'_a$ in S_a for some $u_a \in U_a(x_a)$ such that for all $x''_a \in \mathrm{Post}_{u_a}(x_a)$ there exists a transition $x_b \xrightarrow[b]{u'_b} x''_b$ in S_b with $(x''_a, x''_b) \in Z$;
- $H_c(x_a, x_b) = H_a(x_a)$.

Controller S_c is defined so as to make the relation:

$$\{((x_a, x_b), x'_a) \in Z \times X_a \mid x_a = x'_a\}$$

an alternating simulation relation from S_c to S_a. Arguing as we did for simulation games we arrive at the following result characterizing the solution of bisimulation games.

Theorem 6.19. *Let S_a be a system and let S_b be a system specification with $Y_b = Y_a$. The bisimulation game for system S_a and specification system S_b is solvable iff the maximal fixed-point Z of the operator G_C satisfies $\pi_b(Z \cap (X_{a0} \times X_{b0})) = X_{b0}$. Moreover, Z can be obtained as:*

$$Z = \lim_{i \to \infty} G_C^i(X_a \times X_b).$$

When $\pi_b(Z \cap (X_{a0} \times X_{b0})) = X_{b0}$, a solution to the simulation game is given by the controller (6.4).

For finite-state systems, a fixed-point of G_C is reached after finitely many iterations and the bisimulation game is solvable in time polynomial in $|X_a||X_b|$. Although the operator G_C can also be used for infinite-state systems, a fixed-point may not be reached in finitely many steps unless one is working with an infinite-state system satisfying additional assumptions such as the ones described in Part III.

In situations where it is not possible to initialize S_a we can still apply Theorem 6.19 by strengthening it with the requirement $\pi_a(Z \cap (X_{a0} \times X_{b0})) = X_{a0}$.

6.6 Notes

Problems of control in the behavioral context have been studied in the discrete-event systems community since the pioneering work of Ramadge and Wonham [RW87, RW89]. The main results of this line of work can now be found in several books [KG95, CL99]. Similar problems were independently solved in the context of reactive software synthesis [PR89a, PR89b]. Except for [QL91], the corresponding simulation and bisimulation games have been addressed much more recently and using very different mathematical formalizations [MT02, AVW03, Tab04, ZKJ06, Tab08b]. The use of alternating simulation relations to formalize feedback composition and the systematic exposition based on fixed-points appears to be new.

The adroit reader certainly noticed the reachability problem to be different from all the other control problems considered in this chapter: its solution is given by a minimal and not a maximal fixed-point, and no least restrictive controller exists. Reachability is an instance of a liveness property as opposed to safety. See, *e.g.*, [AS87] for definitions of safety and liveness. This distinction between safety and liveness properties also occurs in verification problems and makes verification a much more interesting topic than what can be judged by the superficial treatment in Chapter 5.

Worth mentioning is also the similarity between the definition of the operator G_C, used to solve bisimulation games, and the definition of alternating simulation relation. This is no coincidence since the solutions of bisimulation games can be completely characterized in terms of certain alternating simulation relations, see [Tab04, Tab08b].