3

Games for Model Checking on Automatic Structures

In the previous chapter we used games as a tool to define the semantics of game quantification and to investigate questions in logic. In this chapter we focus on games in their own right.

We start by defining games played on graphs by two players with perfect information. The connection between such games and logic is illustrated on two well-known examples: the game-theoretical semantics of first-order logic where games of finite duration are used, and model-checking of modal μ-calculus where parity games are appropriate. These two examples show that studying the relation to games can both lead to better insight into the expressive power of a logic and also have an algorithmic utility for model checking. This motivates us to look for games for model-checking on automatic structures.

To find an appropriate game model for first-order logic on an automatic structure, we fix a presentation of the structure and investigate the extended logic $\mathsf{FO}[\exists^{\infty}]$. For this setting, we introduce multiplayer games played by two coalitions with opposing objectives and with imperfect information exchanged according to a hierarchical constraint [50]. On the one hand, this constraint is suitable for defining model-checking games for the extended first-order logic, and it is necessary for the problem of establishing the winning coalition to be decidable. On the other hand, this constraint alone is not sufficient for establishing the winners to be decidable.

To identify the properties needed to make hierarchical games decidable, we study a restricted version of these games where players are forced to alternate. We show that this constraint is required both for determinacy of hierarchical games and for decidability of the problem of establishing the winning coalition. Finally, we prove that hierarchical games where players alternate are indeed model-checking games for $\mathsf{FO}[\exists^{\infty}]$ on automatic presentations.

3.1 Games on Graphs and Logic

In the previous chapter we discussed game quantification and used Gale-Stewart games to provide semantics for game formulas. Since we were only

Ł. Kaiser: Logic and Games on Automatic Structures, LNAI 6810, pp. 29–46, 2011.

interested in the existence of winning strategies, we did not give a formal definition of what a game is in that context. In this section we want to take a step back and define games, more precisely games played on graphs. We also give an overview of the well-known connection between two-player zero-sum games with complete information and fixed-point logics.

The intuition behind a two-player zero-sum turn-based game played on a graph is very natural. Two players, let us call them Player 0 and Player 1, play by moving a token around a graph of positions. There is a position singled out in which the game starts and every position is assigned to one of the players. When the token is in a position that belongs to one of the players, this player is required to move by choosing an edge going out from this position. If there are no outgoing edges, the player who can not move loses. If the players manage to keep playing infinitely long, then the winner is decided based on a winning condition that specifies which infinite plays are winning for Player 0 and which for Player 1.

Definition 3.1. *A Büchi, parity, Streett, Rabin or Muller game is given by a tuple* $\mathcal{G} = (V_0, V_1, E, \mathcal{F})$ *where* V_0 *is the set of positions of Player 0 and* V_1*, disjoint from* V_0*, contains the positions of Player 1.* $E \subseteq V \times V$ *is the edge relation denoting possible moves between positions* $V = V_0 \cup V_1$*, and* $\mathcal{F} \subseteq V^\omega$ *is a winning condition, represented in the same way as Büchi, parity, Streett, Rabin and Muller acceptance conditions for automata described in section 1.3.*

To avoid tedious case distinctions, we often assume that all plays are infinite, i.e. that $vE \neq \emptyset$ for all $v \in V$.

You can see that the Gale-Stewart game for a structure $\mathfrak{A}$ can be viewed as a graph game, either as a game on the tree $\mathfrak{T}(A)$ with players alternating their moves or as a game on the complete bipartite graph $A \times A$ with one side belonging to Player 0 and the other to Player 1.

A strategy for player $i \in \{0, 1\}$ in the game $\mathcal{G}$ is a function $\sigma : V^*V_i \to V$ with $(v, \sigma(hv)) \in E$ for all $h \in V^*$ and $v \in V_i$. A play $\pi = v_0 v_1 \dots$ is *consistent with a strategy* σ for player i if $v_{n+1} = \sigma(v_0 \dots v_n)$ for every n such that $v_n \in V_i$. Given strategies σ, ρ for Player 0 and Player 1, respectively, we denote by $\pi_{\sigma,\rho}(v_0)$ the unique play starting in position v_0 which is consistent with both σ and ρ.

We say that a strategy σ is winning for Player 0 from v_0 if for all strategies ρ of the opponent $\pi_{\sigma,\rho}(v_0) \in \mathcal{F}$. Analogously, a strategy ρ is winning for Player 1 from v_0 if for all strategies σ of the opponent $\pi_{\sigma,\rho}(v_0) \notin \mathcal{F}$. The set of all positions from which player i has a winning strategy is called the winning region of player i. A game $\mathcal{G}$ is determined if from every position either Player 0 or Player 1 has a winning strategy. Thus, in a determined game, the game graph can be partitioned into winning regions of Player 0 and Player 1.

In many cases one is interested not only in arbitrary winning strategies, but in strategies of a special kind. One prominent example are *positional strategies*, where the strategy depends only on the current position and not on the

previous positions of the play, i.e. $\sigma(hv) = \sigma(v)$ for any history h. In a stronger version of determinacy one requires the winning strategies to belong to a certain class. For example, games with parity winning conditions are determined in positional strategies [31, 68], i.e. from every position either Player 0 or Player 1 has a positional strategy that is winning against all strategies of the opponent. For games with Muller winning conditions on finitely many priorities a larger class of strategies is needed, namely such where a finite number of memory states is allowed. We investigate various kinds of determinacy and memory for strategies in chapter 4.

There is an intimate connection between zero-sum games and logic. The idea to give semantics to logics using games was mentioned already in the last decade of the 19th century by C.S. Pierce, and about sixty years later Paul Lorenzen gave a game-theoretical semantics for first-order logic. Giving a game-theoretical semantics to a logic means that for the evaluation of a formula φ on a structure $\mathfrak{A}$ one constructs a model-checking game $\mathsf{MC}(\mathfrak{A}, \varphi)$ such that Player 0 has a winning strategy in $\mathsf{MC}(\mathfrak{A}, \varphi)$ from an initial position exactly if $\mathfrak{A} \models \varphi$.

The model-checking game for an FO formula φ on $\mathfrak{A}$ is constructed in a very intuitive way. The positions of the game consist of subformulas of φ together with a valuation of all free variables in the subformula. If the position is of the form $(\varphi_1 \vee \varphi_2, \theta)$ then Player 0 moves either to (φ_1, θ) or to (φ_2, θ). Analogously, from $(\varphi_1 \wedge \varphi_2, \theta)$ Player 1 moves to one of the subformulas. In a position of the form $(\exists x\varphi, \theta)$, Player 0 moves by choosing an element $a \in \mathfrak{A}$. The next position is then $(\varphi, \theta[x \leftarrow a])$. For $(\forall x\varphi, \theta)$, the other player can make analogous moves. When the game reaches a position (φ, θ) for an atomic formula φ, the winner is determined depending on whether or not $\mathfrak{A}, \theta \models \varphi$.

On finite structures first-order logic is often too weak to express properties of interest. Before we proceed to show model-checking games for first-order logic on infinite structures, let us recall how a more expressive logic, the modal fixed-point logic, can be model-checked on finite structures using parity games.

In computer science, real-world systems are often modeled using finite Kripke structures, which are directed graphs labeled by a set of predicates. Formally, a Kripke structure is a tuple $\mathcal{K} = (V, E, P_1, \ldots, P_k)$ with $E \subseteq V \times V$ and $P_i \subseteq V$. Important properties that often need to be checked on such systems include *reachability*, i.e. the question whether a node where a predicate P_i holds can be reached from an initial node, and *safety*, i.e. the question whether nodes where a predicate P_j holds can be avoided on all possible paths from an initial node. These properties are not definable in FO, but there are well-known temporal logics, like the linear time logic LTL and the branching-time logic CTL, which can express these properties. There is an elegant modal logic that subsumes all these temporal logics and can express many interesting properties, the modal μ-calculus L_μ. Formulas φ of L_μ are formed according to the following syntax,

$$\varphi \;=\; P_i \mid \neg P_i \mid X \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \Box\varphi \mid \Diamond\varphi \mid \mu X\varphi \mid \nu X\varphi,$$

and evaluated on a Kripke structure $\mathcal{K}$ using the following semantics.

- $\mathcal{K}, v \models P_i$ whenever $P_i(v)$ holds and $\mathcal{K}, v \models \neg P_i$ in the other case,
- $\mathcal{K}, v \models \varphi \wedge \psi$ ($\varphi \vee \psi$) whenever $\mathcal{K}, v \models \varphi$ and (or) $\mathcal{K}, v \models \psi$,
- $\mathcal{K}, v \models \Diamond\varphi$ whenever there is a $w \in vE$ for which $\mathcal{K}, w \models \varphi$ holds,
- $\mathcal{K}, v \models \Box\varphi$ whenever $\mathcal{K}, w \models \varphi$ for all $w \in vE$,
- $\mathcal{K}, v \models \mu X\varphi$ whenever $(\mathcal{K}, X), v \models \varphi$, where X is the *smallest* subset of V for which the equation $X = \{w \ : \ (\mathcal{K}, X), w \models \varphi\}$ holds,
- $\mathcal{K}, v \models \nu X\varphi$ whenever $(\mathcal{K}, X), v \models \varphi$, where X is the *biggest* subset of V for which the equation $X = \{w \ : \ (\mathcal{K}, X), w \models \varphi\}$ holds.

Note that in the syntax we use X to denote a set variable, while in the definition of semantics we write $(\mathcal{K}, X)$ for the Kripke structure $\mathcal{K}$ extended with the predicate X. The semantics above is well defined only if the smallest and biggest solutions to the fixed-point equation exist, but this is indeed the case due to the monotonicity of all the operators of L_μ.

The modal μ-calculus is a very expressive logic, in fact it can express all MSO-definable properties that are invariant under bisimulation [46], and most properties of practical interest belong to this class. To define a model-checking game $\mathsf{MC}(\mathcal{K}, \varphi)$ for an L_μ formula φ on a Kripke structure $\mathcal{K}$ one proceeds in an analogous way to first-order logic. Player 0 chooses a successor for $\Diamond$ and $\vee$, while Player 1 moves for $\Box$ and $\wedge$. Additionally, to handle fixed-point operators, from any set variable X a new edge is added back to the formula $\mu X\varphi$ or $\nu X\varphi$ where the variable X was introduced. These back-edges make infinite plays possible and it turns out that the winner of such an infinite play is decided depending on whether the outermost fixed-point variable occurring infinitely often in the play is introduced in a μ or in a ν formula. This corresponds exactly to the parity condition and indeed, not only are parity games powerful enough for model-checking L_μ, the converse holds as well, i.e. winning in any parity game (with a fixed number of priorities) can be expressed in the μ-calculus.

The correspondence between L_μ and parity games is not only an interesting extension of the analogous relation between first-order logic and games of finite duration, it also has interesting algorithmic consequences. While it is still open whether there exists a polynomial-time algorithm for model-checking L_μ, all of the most efficient algorithms known so far [47, 48, 86, 49] rely on the representation of the problem as a game. In particular, one very efficient algorithm [86] heavily exploits the structure of the game. This algorithm does not compute the fixed-points in an iterative way, as suggested by the structure of the L_μ formula. Instead, it starts by guessing a positional strategy in the parity game and then it improves this strategy, which often takes fewer steps than the iterative fixed-point evaluation. The fact that the structure of the game can be of algorithmic use is an additional motivation to look for model-checking games for $\mathsf{FO}[\supset]$ on automatic structures.

3.2 Games with Hierarchical Imperfect Information

Our goal in this section is to describe a class of games that will later be used for model-checking first-order logic with the game quantifier on presentations of automatic structures. To define such games we go beyond two-player perfect information games and use multiplayer games with imperfect information. Even though there are multiple players, in the games we define they form two coalitions with strictly opposing objectives. For this reason one could use a different metaphor with just two players for the same class of games. We use the multiplayer setting in this chapter and discuss the other possibilities in the final chapter.

While imperfect information is a standard element of classical game theory, especially for games in extensive form, in computer science games with imperfect information played on graphs have first been studied in the context of alternating Turing machines with private states [76, 77]. At that time only the reachability condition was considered. Algorithmic solutions for imperfect information games with ω-regular winning conditions were presented only recently [21], however only for the case of observable winning conditions.

The standard way to represent imperfect information in games is by means of *information sets*, equivalence relations describing which states can not be distinguished by a given player. We find it more convenient to use a different representation, in which players see some of the actions of their opponents and other actions are hidden. It is possible to transform between these two representations, but the transformation may increase the size of the game.

Definition 3.2. *A hierarchical Büchi, parity, Rabin, Streett or Muller game with actions in a finite set Σ is given by a tuple*

$$(V_{1,\mathrm{I}}, \ldots, V_{N,\mathrm{I}}, V_{1,\mathrm{II}}, \ldots, V_{N,\mathrm{II}},\ \mu,\ \mathcal{F}).$$

The game is played by two coalitions, I *and* II*, each consisting of N players, with the set of players denoted*

$$\Pi = (1,\mathrm{I}), (2,\mathrm{I}), \ldots, (N,\mathrm{I}), (1,\mathrm{II}), (2,\mathrm{II}), \ldots, (N,\mathrm{II})$$

and the arena of the game given by the pairwise disjoint sets of positions of each player, $V_{1,\mathrm{I}}, \ldots, V_{N,\mathrm{I}}, V_{1,\mathrm{II}}, \ldots, V_{N,\mathrm{II}}$. Positions of coalition I *are denoted by $V_{\mathrm{I}} = V_{1,\mathrm{I}} \cup \ldots \cup V_{N,\mathrm{I}}$ and the ones of coalition* II *by $V_{\mathrm{II}} = V_{1,\mathrm{II}} \cup \ldots \cup V_{N,\mathrm{II}}$, with all positions denoted $V = V_{\mathrm{I}} \cup V_{\mathrm{II}}$. The function $\mu : V \times \Sigma \to V$ defines the possible moves, so that when a player chooses an action $a \in \Sigma$ in his position v then the token is moved and the play proceeds to position $\mu(v, a)$. The objective of coalition* I *is given by the winning condition $\mathcal{F} \subseteq V^\omega$, represented in a finite way as a parity, Streett, Rabin or Muller condition, depending on the type of the game.*

When a hierarchical game is played infinitely long, an infinite sequence of actions is taken by the players during the play, which we call the *play actions*

sequence and denote by $\alpha \in \Sigma^\omega$. Conversely, with every play actions sequence α and a starting position v_0, we associate the unique play $\pi_\alpha(v_0)$. It is the infinite sequence of positions that results from making the moves according to α,

$$\pi_\alpha(v_0) = v_0 v_1 \ldots \iff v_{i+1} = \mu(v_i, \alpha[i]) \text{ for all } i \in \mathbb{N}.$$

During the play $\pi_\alpha(v_0)$ we encounter a sequence of players that take the moves in each step, defined by $\Pi_\alpha(v_0)[i] = p \Leftrightarrow \pi_\alpha(v_0)[i] \in V_p$.

In a hierarchical game each player p has to decide on a strategy $\sigma_p : \Sigma^* \to \Sigma$. In a game with perfect information one says that play actions α are consistent with a strategy σ_p in a play starting in v_0 if for each move i taken by player p the action taken is given by the strategy acting on the history of actions, $\alpha[i] = \sigma_p(\alpha|_i)$.

Since the players do not have perfect information, we additionally assume that for each player p there is a *view function* ν_p that extracts the information visible for this player from the history of play actions. More precisely, let $\nu_p : (\Sigma \times \Pi)^* \to \Sigma^*$ be the function that extracts the information visible to player p from the history of play actions labeled by players who took these actions. We say that a sequence of play actions α is consistent with a strategy σ_p of player p in a play starting in v_0 if, for each i for which $\pi_\alpha[i] \in V_p$, it holds that

$$\alpha[i+1] = \sigma_p(\nu_p((\alpha[0], \Pi_\alpha[0]) \ldots (\alpha[i], \Pi_\alpha[i]))).$$

The above definition of views of play history is very general, but we will only use a concrete special case of *hierarchical* view functions. These hierarchical views allow player k in each coalition to see the moves of players $1, \ldots, k$ in both coalitions, but do not allow him to see the moves of players with numbers $j > k$. Formally, for a player $p = (k, c)$, i.e. player number k in coalition c,

$$\nu_p((a_0, p_0)(a_1, p_1) \ldots (a_n, p_n)) = a_{i_1} a_{i_2} \ldots a_{i_l}$$

if for all $i \in \{i_1, \ldots, i_l\}$ the player $p_i = (l, d)$ has number $l \leq k$, and for all other $j \notin \{i_1, \ldots, i_l\}$ the player $p_j = (m, e)$ has number $m > k$.

There is a good reason to use hierarchical view functions, namely that for most other kinds of information flow, determining the winner, even in a reachability game with three players, is undecidable [4, 2].

To define when coalition I wins a hierarchical game we can not require from all players in this coalition to put forth their winning strategies before players in coalition II do, as it is often done in games with perfect information. Intuitively, in that case players with higher numbers would lose their advantage of information as their strategies would be disclosed too early. Therefore, we use the following definition that requires that strategies are given stepwise, level by level in the information hierarchy.

Definition 3.3. *Coalition* I *wins the hierarchical game*

$$(V_{1,\mathrm{I}}, \ldots, V_{N,\mathrm{I}}, V_{1,\mathrm{II}}, \ldots, V_{N,\mathrm{II}},\ \mu,\ \mathcal{F})$$

starting from position v_0 *if the following condition holds. There exists a strategy* $\sigma_{1,\mathrm{I}}$ *for player* $1,\mathrm{I}$*, such that for each strategy* $\sigma_{1,\mathrm{II}}$ *of player* $1,\mathrm{II}$*, there exists a strategy* $\sigma_{2,\mathrm{I}}$*, such that for each strategy* $\sigma_{2,\mathrm{II}}$*,* ...*, there exists a strategy* $\sigma_{N,\mathrm{I}}$*, such that for each strategy* $\sigma_{N,\mathrm{II}}$*, the play actions sequence* α *that starts from* v_0 *and is consistent with all strategies* $\sigma_{1,\mathrm{I}}, \sigma_{1,\mathrm{II}}, \ldots, \sigma_{N,\mathrm{I}}, \sigma_{N,\mathrm{II}}$ *results in a play winning for* I*, i.e.* $\pi_\alpha(v_0) \in \mathcal{F}$*.*

The definition for coalition II is analogous, i.e. there exists a $\sigma_{1,\mathrm{II}}$, such that for all $\sigma_{1,\mathrm{I}}$, ..., the play is winning for II, i.e. $\pi_\alpha(v_0) \notin \mathcal{F}$.

Example 3.4. To get an intuition about the kind of interactions that appear in hierarchical games, let us consider the simple game depicted in Figure 3.1 in two variants. The positions of coalition I are round, the positions of coalition II are square, there are two levels of information, and the positions on the upper level are dotted.

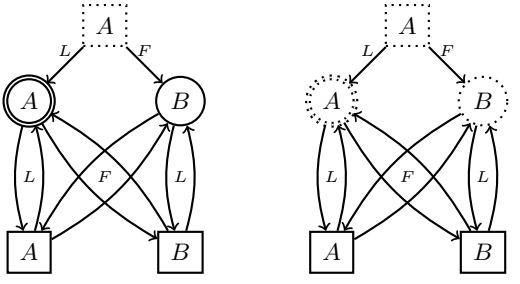

Fig. 3.1. Example of a hierarchical game in two variants

You can think of this game as played using a coin with two sides, A and B. Each of the players can choose to either flip the coin (F) or leave it as it is (L). Formally, there are four players in this game, two in each coalition. The top position belongs to $2,\mathrm{II}$ and the two bottom positions belong to $1,\mathrm{II}$. The game proceeds as follows: first the second player of coalition II chooses either to flip the coin or to leave it intact. Afterward, only the other two players play by either flipping the coin or leaving it as it is. Coalition I wins if the A side of the coin is seen infinitely often in positions where players in coalition I move, as marked in Figure 3.1.

To illustrate the importance of hierarchical information levels we consider two variants of this game. In the first one (left), the bottom strongly connected component belongs to players on the same information level, i.e. to $1,\mathrm{II}$ and $1,\mathrm{I}$. In

this scenario, coalition II can win, because first player 2, II can flip the coin to B and later player 1, II can always repeat the last move of player 1, I.

In the other variant (right), the player in coalition I has more information, i.e. the bottom strongly connected component belongs to 1, II and 2, I, with $V_{1,\mathrm{I}} = \emptyset$. In this case coalition I can win, because the strategy of player 2, I is given after the strategy of 1, II is set. Therefore, player 2, I can assure that the coin will be flipped after each two moves, which guarantees that I holds the coin on the A side infinitely often, independent of the first move of 2, II.

3.3 Alternation of Moves in Hierarchical Games

In games with perfect information it is not necessary to assume that the players move in any fixed order. Moreover, the assumption that players move in an alternating way can be made without loss of generality. We show that this is not the case for hierarchical games. Thus, we define an *alternating hierarchical game* as a hierarchical game, where for each letter $a \in \Sigma$ and each level $i = 1, \ldots, N$ the following alternation conditions hold:

$$v_i \in V_{i,\mathrm{I}} \implies \mu(v_i, a) \in V_{i,\mathrm{II}},$$
$$v_i \in V_{i,\mathrm{II}} \implies \mu(v_i, a) \in V_{(i \bmod N)+1,\mathrm{I}}.$$

To see that non-alternating hierarchical games can not be reduced to alternating ones, let us consider the game depicted in Figure 3.2. The leftmost and the rightmost bottom position is winning for coalition I, while in the other two bottom positions coalition I loses. This simple hierarchical game is not alternating and we show that it is not determined. To win this game, the player on the lower level of information, i.e. 1, I or 1, II, has to predict the move of the opponent, i.e. 1, II or 1, I. In particular, his strategy has to start with an a exactly if the opponent starts with an a. As this holds for players in both coalitions, it leads to a non-determined game as each player can counter the strategy of the opponent, once it is known.

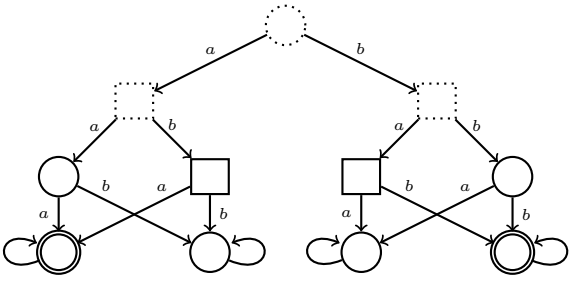

Fig. 3.2. Non-determined hierarchical game

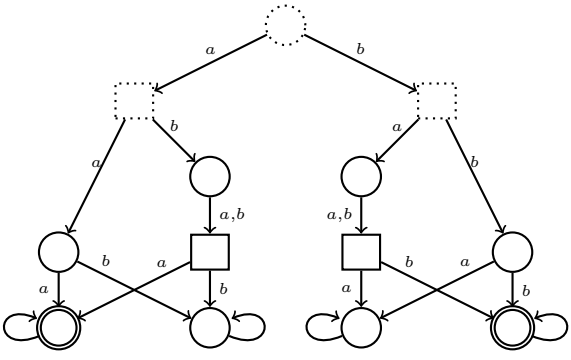

Fig. 3.3. Alternation makes hierarchical games determined

Introducing alternation of moves, even in the simplest possible way, changes this situation. The game depicted in Figure 3.3 is identical to the one in Figure 3.2 except for two additional positions of player 1, I. These positions may seem useless as there is no choice to be made there, but the new game is determined. To convince yourself that, in the extended game, coalition II can indeed win, take the following strategy of player 1, II: let him always play the opposite move to the one that was taken before by player 1, I. For player 2, II take the following strategy: if player 1, I declared that he will play a first, then play b, and else play a first. You can check that these strategies are indeed winning for coalition II, but this is possible only because when constructing the strategy for 1, II the first letter played by 1, I was already known.

Another important difference between alternating and non-alternating hierarchical games is decidability of the problem of establishing whether coalition I wins the game. We show in the next section that this problem is decidable for alternating hierarchical games, and here we prove that in the general non-alternating case it is undecidable. The differences between alternating and non-alternating hierarchical games can be explained on the level of logic and model-checking, as alternating hierarchical games correspond to model-checking on automatic presentations, while non-alternating games correspond to model-checking on presentations that use asynchronous automata, known as rational structures, which have undecidable first-order theory. It is also interesting to observe that the proof of undecidability uses the fact that all players in hierarchical games as we defined them choose actions from the same alphabet Σ. If we assume that in a hierarchical game every player chooses actions from his own alphabet, which does not overlap with the alphabet of any other player, then establishing which coalition wins is decidable even for non-alternating games, cf. [73].

Theorem 3.5. *The question whether coalition* I *wins in a hierarchical Büchi game is undecidable.*

Proof. We reduce the Post correspondence problem for $\overline{u} = u_1, \ldots, u_K$ and $\overline{v} = v_1, \ldots, v_K$, where $u_i, v_i \in \{a, b\}^*$, to the problem whether coalition I wins in the hierarchical game $\mathcal{G}_{\overline{u},\overline{v}}$. The possible actions in $\mathcal{G}_{\overline{u},\overline{v}}$ are $\Sigma = \{a, b, \Box, 1, 2, \ldots, K\}$ and they intuitively correspond to the players choosing letters of the words u_i, v_i, a special delimiter $\Box$, and choosing which word to play next.

In constructing $\mathcal{G}_{\overline{u},\overline{v}}$, we are going to use subgames such that, for a given word u, the subgame enforces that u is played, or else the player that moves loses. Such a subgame has one more position than the length of u, and if the wrong letter is chosen then the move leads to a position where the player loses. There is only one outgoing edge in such a subgame, the one taken when the last letter of u is played. In Figure 3.4 we depicted an example subgame for $u = aba$ and player 1, I, who loses in the rightmost position.

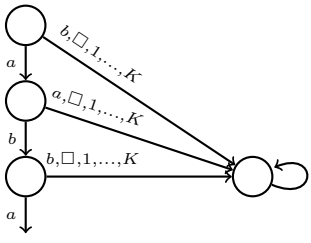

Fig. 3.4. Example subgame for $u = aba$

We start the construction of the game $\mathcal{G}_{\overline{u},\overline{v}}$ with a position belonging to player 3, II with two possible (non-losing) moves. In this position, coalition II can decide if the test will be done for the words $\overline{u}$ or for the words $\overline{v}$. All other positions will be on lower levels of information and we construct them in such a way that coalition I will never be able to deduce in which component the play is taking place.

Each of the two components, for $\overline{u}$ and for $\overline{v}$, starts with a position of player 2, I where this player chooses if he wants to play a word with index $1, \ldots, K$ or the special symbol $\Box$. If the special symbol is chosen, player 1, I must play the same symbol $\Box$ and the play returns back to the position, where 2, I chooses a word. When an index L is chosen, then in each of the components first the word v_L and then the word u_L is played. The difference is that, in the first component (for $\overline{u}$), it is player 2, II who must play v_L and player 1, I must play u_L, while in the other component (for $\overline{v}$), it is player 1, I who must play v_L and player 2, II who must play u_L. After the two words were played, the play returns to the position where 2, I chooses the index of a word to be played. The complete game is depicted in Figure 3.5, using subgames for u_i and v_i.

The winning condition is defined as follows: the special symbol $\Box$ must be chosen by 2, I infinitely often and additionally there must be another action L, different from $\Box$, that is played infinitely often. While this is not directly a

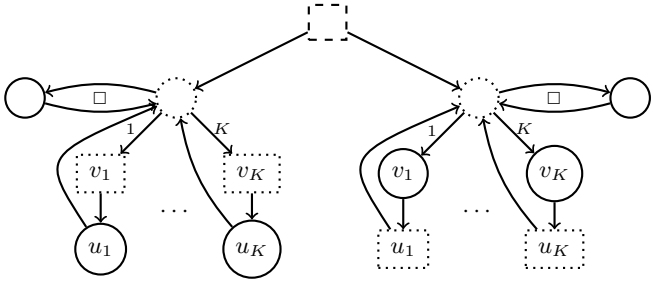

Fig. 3.5. Complete game $\mathcal{G}_{\overline{u},\overline{v}}$

Büchi condition, the game can be transformed into a game with Büchi winning condition. In the modified game, one more position for player 2, I is added in each component, with the same moves as in the original one except for the possibility of choosing □. In the transformed game, when 1, I chooses □ in the only position where he is allowed to do so, the play proceeds from the new position of 2, I where □ is not allowed, thus ensuring that a non-□ action is taken.

Let us first show that if there is a solution for the Post correspondence problem for $\overline{u}$ and $\overline{v}$ then coalition I has winning strategies for $\mathcal{G}_{\overline{u},\overline{v}}$. Indeed, let $i_1, i_2, \ldots, i_M$ be the indices for the solution of the correspondence problem, so that $u_{i_1} u_{i_2} \ldots u_{i_M} = v_{i_1} v_{i_2} \ldots v_{i_M}$. Let player 2, I choose i_1 in his first move, then i_2, i_3, and so on up to i_M, then the special symbol □, and then again i_1, i_2, and so on. Player 1, I is going to play the letters from the word $u_{i_1} u_{i_2} \ldots u_{i_M}$ in turn, and then □, and then again the letters $u_{i_1} u_{i_2} \ldots u_{i_M}$, and □, and so on. Clearly, player 2, I chooses □ and non-□ infinitely often, so to show that coalition I wins we only need to prove that player 1, I will never play the wrong letter in a subgame for some word w. If the play is taking place in the $\overline{u}$ component this is clear from the definition of the strategies given above, as player 1, I plays exactly the words indices of which player 2, I chooses. When the play takes place in the $\overline{v}$ component, the indices chosen by player 2, I force player 1, I to play the words $v_{i_1}, v_{i_2}, \ldots, v_{i_M}$. But since $u_{i_1} u_{i_2} \ldots u_{i_M} = v_{i_1} v_{i_2} \ldots v_{i_M}$, this is equivalent to playing the u_i words with the same indices, which is exactly the strategy that player 1, I uses.

To prove the converse, namely that if there is a winning strategy for coalition I then the correspondence problem has a solution, observe two intuitive facts. First, 2, I can never deduce in which component the play is taking place, because what he can see after each of his moves is the same in both components. Secondly, □ can be played by 2, I only if the words played up to that point have the same length in both components. Otherwise, coalition I would lose as □ can not be played in a subgame for any word.

Formally, let us first fix the only rational strategy for 2, II, namely that if a number L was the most recent action in the play, then 2, II plays v_L, and if

there were other actions from $\{a, b\}^*$ taken after the last time a number L was played, then he plays u_L. Note that the above construction implies that player $2, \mathrm{II}$ knows in which component the play takes place, even if the move of $3, \mathrm{II}$ is not visible for him. With this strategy fixed, the condition that coalition I has a winning strategy for $\mathcal{G}_{\overline{u},\overline{v}}$ means that there exists a strategy σ_1 for player $1, \mathrm{I}$ and a strategy σ_2 for player $2, \mathrm{I}$ such that the play corresponding to these two strategies and the one fixed for $2, \mathrm{II}$ is winning for coalition I, independent of the component chosen by $3, \mathrm{II}$.

Let us first concentrate on the strategy σ_2. Since, according to the winning condition, $\square$ can not be the only action played infinitely often, and in each component the only possible answer to $\square$ is again $\square$, let us assume without loss of generality that the first move taken by σ_2 is not $\square$ and let it be L_1. After choosing L_1 the play goes through v_{L_1} and u_{L_1} and does not stop, since player $2, \mathrm{II}$ uses a fixed strategy that prevents him from losing in a subgame and player $1, \mathrm{I}$ plays a winning strategy. Let us denote by L_2 the next move of $2, \mathrm{I}$, i.e. $L_1 = \sigma_2(\varepsilon)$, $L_2 = \sigma_2(L_1 v_{L_1} u_{L_1})$, and continue the play denoting the subsequent moves of $2, \mathrm{I}$ by $L_2, \ldots, L_M$, up to the point where he plays $\square$. Formally,

$$L_1 = \sigma_2(\varepsilon), \qquad L_{i+1} = \sigma_2(L_1 v_{L_1} u_{L_1} \ldots L_i v_{L_i} u_{L_i}), \qquad L_{M+1} = \square.$$

After extracting the sequence $L_1, \ldots, L_M$ of moves of $2, \mathrm{I}$ from his winning strategy σ_2, let us look at player $1, \mathrm{I}$. This is the only player on information level 1 so he only sees his own previous moves. In this case, the strategy σ_1 is in fact completely described by the word $t \in \{a, b, \square\}^\omega$ such that

$$t[i] = \sigma_1(t|_i) \text{ for all } i \in \mathbb{N}.$$

Due to the structure of the game, no $\square$ can be played by $1, \mathrm{I}$ before $2, \mathrm{I}$ decides to play $\square$, and then $\square$ must be played. Therefore, if w is the prefix of t up to the first occurrence of $\square$, then w is exactly the word played by $1, \mathrm{I}$ while $2, \mathrm{I}$ played the moves $L_1, \ldots, L_M$. But due to the structure of the game $\mathcal{G}_{\overline{u},\overline{v}}$, coalition II can decide if $w = u_{L_1} \ldots u_{L_M}$ or if $w = v_{L_1} \ldots v_{L_M}$. Since we extracted both $L_1, \ldots, L_M$ and w independent of this choice, w has to be good for both cases. Therefore it is the solution for the Post correspondence problem as requested. $\square$

3.4 Model Checking with Hierarchical Games

We observed that non-alternating hierarchical games are neither determined nor decidable, so we concentrate on the alternating version. Indeed, we prove that alternating hierarchical games are exactly the games needed for model-checking $\mathsf{FO}[\supset]$ on presentations of automatic structures.

To start with, observe that in an alternating game every infinite sequence of play actions can be divided into blocks of $2N$ actions, each taken by a different player,

$$\alpha = a_0^{1,\mathrm{I}} a_0^{1,\mathrm{II}} a_0^{2,\mathrm{I}} a_0^{2,\mathrm{II}} \dots a_0^{N,\mathrm{I}} a_0^{N,\mathrm{II}} a_1^{1,\mathrm{I}} \dots a_1^{N,\mathrm{II}} a_2^{1,\mathrm{I}} \dots .$$

Let the $2N$-*split* of these play actions be the tuple of $2N$ words of actions played by each of the players,

$$\mathrm{split}_{2N}(\alpha) = (a_0^{1,\mathrm{I}} a_1^{1,\mathrm{I}} \dots,\ \{a_i^{1,\mathrm{II}}\}_{i\in\mathbb{N}},\ \dots,\ \{a_i^{N,\mathrm{I}}\}_{i\in\mathbb{N}},\ \{a_i^{N,\mathrm{II}}\}_{i\in\mathbb{N}}).$$

Observe that since the set of plays winning for coalition I and starting from a fixed v_0 is ω-regular, also the set of corresponding $2N$-splits of play actions is ω-regular. This is a known property of ω-regular languages, and it can be proved by taking each $2N$th state of the automaton recognizing the plays and making a product with Σ^{2N} to store the states that were omitted from the original automaton. For an alternating hierarchical game $\mathcal{G}$ with winning condition $\mathcal{F}$ let us denote the $2N$ary relation recognizing the $2N$-split of plays winning for coalition I by $W_{\mathrm{I}}^{\mathcal{G},v_0}(\beta_1, \dots, \beta_{2N})$, formally defined by

$$W_{\mathrm{I}}^{\mathcal{G},v_0}(\overline{\beta}) \iff \forall\alpha\ (\ \mathrm{split}_{2N}(\alpha) = \overline{\beta}\ \Rightarrow\ \pi_\alpha(v_0) \in \mathcal{F}\ ).$$

The definition for coalition II is analogous with $\pi_\alpha(v_0)) \notin \mathcal{F}$.

Using the relation W_{I}^{G,v_0} we can express in $\mathsf{FO}[\eth]$ that coalition I wins in the alternating hierarchical game $\mathcal{G}$, which results in the following theorem.

Theorem 3.6. *For any alternating hierarchical game $\mathcal{G}$ and the relation $W_{\mathrm{I}}^{\mathcal{G},v_0}$ defined as above, coalition* I *wins the game $\mathcal{G}$ starting from v_0 if and only if the following formula φ_{I} holds in $(\Sigma^\omega, W_{\mathrm{I}}^{\mathcal{G},v_0})$:*

$$\varphi_{\mathrm{I}} = \eth x_1 y_1 \dots \eth x_N y_N\ W_{\mathrm{I}}^{\mathcal{G},v_0}(x_1, y_1, \dots, x_N, y_N).$$

Proof. Let us recapitulate the definition of coalition I winning a hierarchical game and the semantics of the formula φ_{I}. Coalition I wins $\mathcal{G}$ if there is a strategy σ_1 for player on level 1 of coalition I, so that for any counter-strategy ρ_1, there exists a strategy σ_2, and so on up to σ_N, such that for all ρ_N the resulting play must be won by coalition I. On the other hand, the formula φ_{I}, according to the definition of $\eth$, says that there is a function f_1, so that for all functions g_1, there is a function f_2, and so on up to a function f_N, such that for all g_N, if we construct the words according to $\overline{f}$ and $\overline{g}$ then they form a $2N$-split of a play that is won by coalition I.

As the structure and the final condition in both definitions are equivalent, due to the definition of $W_{\mathrm{I}}^{\mathcal{G},v_0}$, the only remaining task is to show how the functions f_i, g_i and the strategies σ_i, ρ_i are related. It is intuitively clear that the functions and the strategies are closely related, the only difference is that the functions f_i, g_i operate on prefixes of x_i, y_i while the strategies σ_i, ρ_i take all actions of all players $j \leq i$ as arguments, which corresponds to prefixes of all words x_j, y_j with $j \leq i$. Intuitively, this makes no difference since the words x_j, y_j are completely fixed before the function f_i is constructed, and we are going to prove it formally.

Let us construct, given the function f_i, a strategy $\sigma_i^{f_i}$. The strategy $\sigma_i^{f_i}$ applied to a view h of the history of play actions extracts from h the sequences h_{I}^i and h_{II}^i of actions of players i, I and i, II, respectively, and chooses $f_i(h_{\mathrm{I}}^i, h_{\mathrm{II}}^i)$ as the next action. It is possible to extract h_{I}^i and h_{II}^i from h due to the alternation condition, because we know that h is of the form $a_0^{1,\mathrm{I}} a_0^{1,\mathrm{II}} a_0^{2,\mathrm{I}} a_0^{2,\mathrm{II}} \dots a_0^{i,\mathrm{I}} a_0^{i,\mathrm{II}} a_1^{1,\mathrm{I}} \dots$ and the sequences $h_{\mathrm{I}}^i = a_0^{i,\mathrm{I}} a_1^{i,\mathrm{I}} \dots$ and h_{II}^i can be computed by taking every $2i$th position in h starting from $2i-1$ and $2i$, respectively. Note that extracting these sequences would not be possible if it was not clear which player made which move, which we used in the previous proof of undecidability.

Let us now do the converse and construct, given the strategy σ_i, the function $f_i^{\sigma_i}$. For this construction we need to have all the f_j, g_j with $j < i$ already constructed, thus we write $f_i^{\{\sigma_j,\rho_j\}_{j\le i}}$. Using the constructed functions f_j, g_j, we can assume that the words x_j, y_j are already fixed. The result of

$$f_i^{\{\sigma_j,\rho_j\}_{j\le i}}(x_i[0] \dots x_i[n], y_i[0], \dots y_i[n])$$

is given by

$$\sigma_i(x_1[0]y_1[0] \dots x_i[0]y_i[0]x_1[1]y_1[1] \dots x_i[1]y_i[1] \dots x_i[n]y_i[n]).$$

The constructions relating g_i and ρ_i are analogous. Observe that if

$$W_{\mathrm{I}}^{\mathcal{G},v_0}(x_{f_1 g_1} y_{f_1 g_1}, \dots, x_{f_N g_N} y_{f_N g_N})$$

holds for some functions $\overline{f}, \overline{g}$ then, by the above definition, we have that the play $\pi(v_0, \sigma_1^{f_1}, \rho_1^{g_1}, \dots, \sigma_N^{f_N}, \rho_N^{g_N})$ is in $\mathcal{F}$. Moreover, the converse holds as well, i.e. if for some strategies $\overline{\sigma}, \overline{\rho}$ we have

$$\pi(v_0, \sigma_1, \rho_1, \dots, \sigma_N, \rho_N) \in \mathcal{F},$$

then $W_{\mathrm{I}}^{\mathcal{G},v_0}(x_{f_1 g_1} y_{f_1 g_1}, \dots, x_{f_N g_N} y_{f_N g_N})$ holds, where $f_i = f_i^{\{\sigma_j,\rho_j\}_{j\le i}}$ and $g_i = g_i^{\{\sigma_j,\rho_j\}_{j\le i}}$ are the functions constructed above.

This correspondence allows to exploit the similarity of the structure of the definition of the $\mathsf{FO}[\Game]$ formula φ_{I} and the definition of coalition I winning in $\mathcal{G}$. Intuitively, it is enough to insert the transformed functions and strategies into the definition to arrive at a contradiction and finish this proof. To avoid cluttered notation, we formally present only one direction in the case of two levels, the other direction and the proof for more levels is analogous.

Let us assume that φ_{I} holds and coalition I does not win $\mathcal{G}$, formally

(1) $\exists f_1 \forall g_1 \exists f_2 \forall g_2 \; W_{\mathrm{I}}^{\mathcal{G},v_0}(x_{f_1 g_1}, y_{f_1 g_1}, x_{f_2 g_2}, y_{f_2 g_2})$,

(2) $\forall \sigma_1 \exists \rho_1 \forall \sigma_2 \exists \rho_2 \; \pi(\sigma_1, \rho_1, \sigma_2, \rho_2) \notin \mathcal{F}$.

Let us fix f_1 that exists by our first assumption, set $\sigma_1 = \sigma_1^{f_1}$ and fix ρ_1 that exists by the second assumption for this σ_1. Let us now set $g_1 = g_1^{\rho_1}$ and fix

f_2 that exists by the first assumption. Finally, let us set $\sigma_2 = \sigma_2^{f_2}$ and fix ρ_2 that exists by the second assumption. By the previous observation

$$W_{\mathrm{I}}^{\mathcal{G},v_0}(x_{f_1g_1}, y_{f_1g_1}, x_{f_2g_2}, y_{f_2g_2}) \iff \pi(\sigma_1, \rho_1, \sigma_2, \rho_2) \in \mathcal{F},$$

but this contradicts the two assumptions above. □

Observe that the same proof works for the other coalition and an analogous relation $W_{\mathrm{II}}^{\mathcal{G},v_0}$. Thus, the negation normal form of $\mathsf{FO}[\partial]$ corresponds to determinacy of alternating hierarchical games.

Corollary 3.7. *Alternating hierarchical games are determined.*

After we captured winning in alternating games in $\mathsf{FO}[\partial]$ let us do the converse and construct the model-checking game for a given $\mathsf{FO}[\partial]$ formula on an automatic presentation $\mathfrak{A}$. At first, we restrict ourselves to formulas of the form

$$\varphi = \partial x_1 y_1 \partial x_2 y_2 \dots \partial x_N y_N \; R(x_1, y_1, \dots, x_N, y_N)$$

and construct a game so that the *split* of the winning plays will allow us to use the previous theorem.

Intuitively, the construction can be understood as prefixing each variable with all possible letters in the order of information hierarchy and making a step of the automaton when all the variables are prefixed. To define these games precisely, let us take the deterministic automaton for R, denoted $\mathcal{A}_R = (Q, q_0, \delta, \mathcal{F}_R)$, and construct the model-checking game $\mathcal{G}_\varphi$ for φ in the following way.

For each tuple of letters $c_1, d_1, c_2, d_2, \dots, c_M, d_M$ of even length, with $0 \leq M < N$, and for every state $q \in Q$, we have in $\mathcal{G}_\varphi$ the position

$$R^q(c_1x_1, d_1y_1, \dots, c_Mx_M, d_My_M, x_{M+1}, \dots, y_N). \tag{3.1}$$

Moreover, for each tuple $c_1, d_1, c_2, d_2, \dots, c_M, d_M, c_{M+1}$ of odd length, we have the position

$$R^q(c_1x_1, \dots, d_My_M, c_{M+1}x_{M+1}, y_{M+1}, \dots, y_N). \tag{3.2}$$

In each of these positions, the next move is made by the player corresponding to the next variable that is not yet prefixed by a letter, e.g. in position 3.1 it is the player $M+1$ of coalition I who makes the move for x_{M+1} and in position 3.2 it is the player $M+1$ of coalition II. We can formally define the set of positions of players on each level i as $V_{i,\mathrm{I}} = Q \times \Sigma^{2(i-1)}$, $V_{i,\mathrm{II}} = Q \times \Sigma^{2i-1}$.

The moves in $\mathcal{G}_\varphi$ intuitively correspond to the player choosing a letter to prefix his variable with, so for $0 \leq M < N$

$$\mu(R^q(c_1x_1, \dots, d_My_M, x_{M+1}, \dots, y_N), c_{M+1}) =$$

$$R^q(c_1x_1, \dots, d_My_M, c_{M+1}x_{M+1}, y_{M+1}, \dots, y_N),$$

and for $0 \leq M < N - 1$

$$\mu(R^q(c_1x_1, \ldots, c_{M+1}x_{M+1}, y_{M+1}, \ldots, y_N), d_{M+1}) =$$

$$R^q(c_1x_1, \ldots, c_{M+1}x_{M+1}, d_{M+1}y_{M+1}, x_{M+2}, \ldots, y_N).$$

The only special case is the final position $R^q(c_1x_1, d_1y_1, \ldots, c_Nx_N, y_N)$. When player N, II chooses the final letter d_N, it will not be appended, but instead all prefixing letters will be removed and the state of the automaton will be changed as follows, with $\overline{\alpha} = c_1d_1 \ldots c_Nd_N$:

$$\mu(R^q(c_1x_1, d_1y_1, \ldots, c_Nx_N, y_N), d_N) = R^{\delta(q,\overline{\alpha})}(x_1, \ldots, y_N).$$

We derive the winning condition $\mathcal{F}$ of the game $\mathcal{G}_\varphi$ from the acceptance condition $\mathcal{F}_R$ of the automaton for R in the following way. Only the state component of each position in the game is taken into account, i.e. a sequence π of positions of $\mathcal{G}_\varphi$ is in $\mathcal{F}$ if and only if π projected to the state component is in $\mathcal{F}_R$.

To see that the game $\mathcal{G}_\varphi$ is indeed the model-checking game for φ, we use Theorem 3.6 and observe that the $2N$-split of the winning paths in $\mathcal{G}_\varphi$ is exactly the relation R, $W_{\mathrm{I}}^{\mathcal{G}_\varphi, R^{q_0}(x_1,y_1,\ldots,x_N,y_N)} = R$.

In this way, the model-checking game for formulas in the considered form is constructed. As we proved, any formula in $\mathsf{FO}[\eth]$ can be written in negation normal form and additionally, by renaming variables, it can be transformed into prenex normal form. Let us therefore consider a general formula in the form $\varphi = \eth x_1y_1 \ldots \eth x_Ny_N\ \psi(x_1, y_1, \ldots, x_N, y_N)$, where ψ is in negation normal form and does not contain quantifiers. We construct the game $\mathcal{G}_\varphi$ inductively with respect to ψ.

In the case of $\psi(\overline{x}) = R(\overline{x})$ or $\psi(\overline{x}) = \neg R(\overline{x})$ the solution was already presented, when considering $\neg R$ we just have to complement the acceptance condition of the automaton for R. Let us show how to construct the game for Boolean connectives, i.e. for $\psi_1 \wedge \psi_2$ and for $\psi_1 \vee \psi_2$. We want to adhere to the usual convention of model-checking games and to have only one additional position for any Boolean connective. The game for $\psi_1 \circ \psi_2$, where $\circ = \wedge, \vee$, is therefore constructed as follows: we take the two games for ψ_1 and ψ_2 and we add one more position on higher level of information that has two possible moves — to the starting position of ψ_1 and to the starting position of ψ_2. The new position belongs to coalition I when $\circ = \vee$ and to coalition II when $\circ = \wedge$ and in both cases the other coalition does not play on that information level. With the construction described above we face a problem, as the game is not strictly alternating any more, but this time it can be made alternating by adding dummy positions, as presented in Example 3.8.

To formally prove that the resulting games are indeed model-checking games for formulas with Boolean connectives, we replace the connectives with a new variable and the formula with a relation where only the first letter of the

new variable corresponding to the Boolean connective is considered. Then the automaton for such a relation corresponds to the defined game and Theorem 3.6 can be used again.

Example 3.8. To illustrate the construction of model-checking games and the method to overcome the technical problem with non-alternating games mentioned above, let us consider the simple formula $\exists x\ (R_1(x) \wedge R_2(x))$ over $\{a,b\}^\omega$ with $R_1 = \{a^\omega\}$ and $R_2 = \{a,b\}^\omega \backslash \{a^\omega\}$. Both the automaton for R_1 and the one for R_2 has two states and the transition functions are identical. On any b the automata go from q_0 to q_1 and stay there forever. Only the Büchi acceptance conditions differ, with $F_1 = \{q_0\}$ and $F_2 = \{q_1\}$.

In Figure 3.6, the game for this formula is depicted. We show dummy moves for the second player, as formally $\exists x\varphi(x) \equiv \exists xy\varphi(x)$. Note that this is actually a four-player game and the top position belongs to player 2, II. Since the formula is false, coalition II wins this game. Indeed, for coalition I to win, player 1, I would have to present a strategy to visit both of the double-circled vertices infinitely often without knowing in which branch he is, and that is impossible.

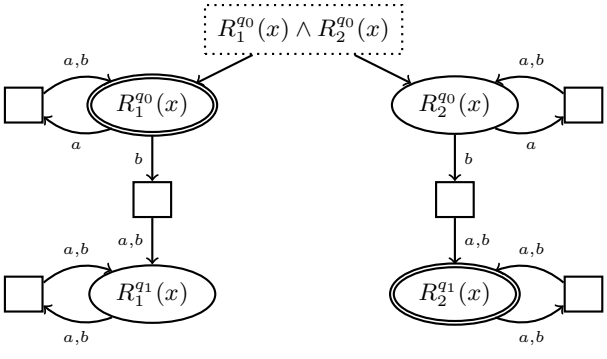

Fig. 3.6. Model-checking game for $\exists x(R_1(x) \wedge R_2(x))$

To fix the problem with alternation, let us add positions where there is no choice for the player. The alternating game for $\psi_1 \circ \psi_2$ is depicted in Figure 3.7. In this game, dummy positions are added there, where it is necessary to make the game alternating. It is clear that winning strategies in these two games can be transferred, as in each move on each level of visibility the players know how many moves on the other levels were made, both in the original game depicted in Figure 3.6 and in the modified one in Figure 3.7.

The tight correspondence between alternating hierarchical games and $\mathsf{FO}[\exists^\infty]$ makes it possible to use our knowledge about this logic to reason about the games. In particular, we can transfer the results about complexity, including the non-elementary lower bound on deciding $\mathsf{FO}[\exists^\infty]$ on automatic presentations, which allows us to conclude with the following corollary.

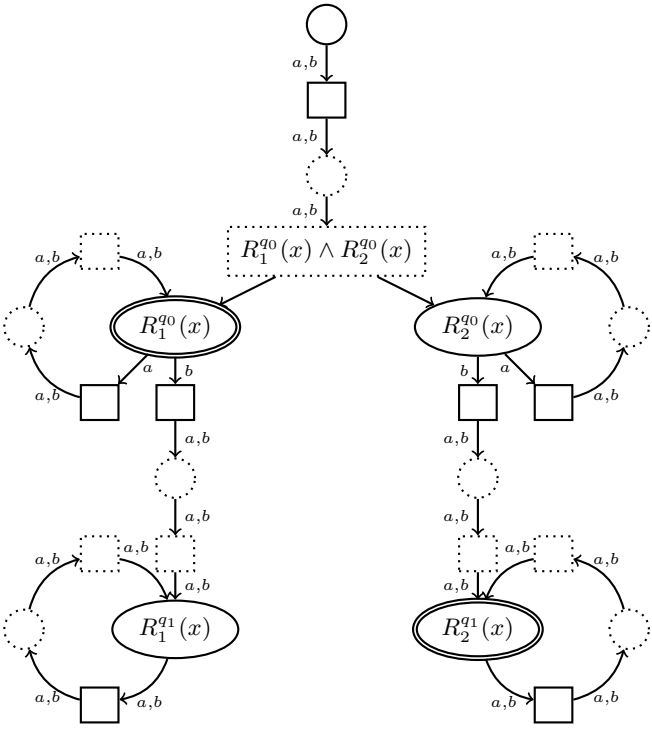

Fig. 3.7. Alternating game for $\exists x(R_1(x) \wedge R_2(x))$

Corollary 3.9. *The question whether coalition* I *wins in an alternating hierarchical game on a finite arena is decidable and has non-elementary complexity when the number of levels is not fixed. It can be decided in* $2k$EXPTIME *for games with at most* k *levels.*