

# Automata and Compositional Strategies in Extensive Form Games

Soumya Paul<sup>1</sup>, R. Ramanujam<sup>2</sup>(✉), and Sunil Simon<sup>3</sup>

<sup>1</sup> IRIT, Université Paul Sabatier, Toulouse, France

<sup>2</sup> Institute of Mathematical Sciences, Chennai, India  
jam@imsc.res.in

<sup>3</sup> Indian Institute of Technology, Kanpur, India

**Abstract.** We study automata as memory structure for “online” strategizing in extensive form games. By online strategizing we mean a model in which players start with potential (partial) strategies that are generic plans for (local) subgames and dynamically compose and switch between them. We consider such strategizing to be relevant for a theory of play. We suggest that for sufficiently large games and resource limited players, the game is better modelled as an infinite horizon game, and thus the study is carried out in games of infinite duration on finite game arenas. We show how strategy switching can be realised by finite state transducers and how they can be used to answer questions on stability of strategies.

**Keywords:** Memory in strategies · Strategy specifications · Strategy switching · Infinite games on finite graphs

## 1 Overview

The seminal paper of John von Neumann [45] begins with a section titled **Great Simplification**. In it, he brilliantly lays down a rationale that has dominated game theory: there is no loss of generality in assuming that a rational player chooses his strategy before the game begins, since a strategy lets him specify a choice for every possible historical situation he might find himself in during the game. So von Neumann concludes that each player must choose his strategy *without being informed of the other players’ strategic choices*. Indeed this is the great simplification<sup>1</sup> that led to normal form game representations and much of game theory as we know it.

In the case of finite extensive form games, this abstraction works very nicely. Even if the game is one of imperfect information, such an abstraction helps us to ignore the extensive temporal structures of games and concentrate on outcome based analysis. Equilibrium theory helps us predict how a rational player would choose; in a prescriptive sense, the theory tries to give good advice to the decision maker. The strategy in an equilibrium profile can be seen as such an advice.

---

<sup>1</sup> due originally to Émile Borel [11]; von Neumann was only developing the idea.

Difficulties arise in the case of games with multiple equilibria, and this has been extensively discussed in the literature [35]. An entirely different kind of trouble arises when **computational** considerations matter. In particular, suppose that the advice oracle that the strategy represents is to be a computer program, then the notion of strategy as any complete plan for all player moves needs re-examination. To take an admittedly ridiculous example, consider a game in which a player has a binary choice, and the strategy defined by: when it's the player's turn to move and the game position is node  $n$  (under some fixed ordering of tree nodes), play 1 if the  $n^{\text{th}}$  Turing machine (under a fixed enumeration of Turing machines) halts on empty input, and play 0 otherwise. This would indeed be a strategy, and may even survive elimination of dominated strategies for particular outcomes, but not only does this function lack reason, it is also unimplementable (as a computer program). The space of functions from player nodes to available choices is indeed too rich. Taking implementability as a criterion, we can look for algorithms that compute equilibrium strategies and such an *algorithmic game theory* [34] is being developed by computer scientists.

On the other hand, *epistemic game theory* [38] attempts to study the rationale underlying a strategy, examining the reasons that underlie the choice made by a player at a game position. Such a viewpoint refuses the offer made by the great simplification and delves into the temporal extensive game structure. This is particularly important when a player finds the history of play on an off-equilibrium path. Given that an opponent has deviated from a choice dictated by an optimal strategy, how should the player expect the opponent to play in future? Several solutions have been proposed in the literature, such as forward induction [38] but the main point is that such solutions involve *online* strategizing, during course of play, rather than the offline strategies given by the great simplification.

Aumann and Dreze [3] make a strong case for the focus of game theory to shift from the existence of equilibria to a prescriptive theory that would advise a player how to play in any particular situation, in light of the history that led to the situation. In a series of articles, van Benthem [6–9] has called for a *theory of play*, which includes not only the deliberative aspects of pre-game strategizing but also reasoning in the game. This involves consideration of a range of events that occur during play: players' observations, information received about other players, etc.; these cause a revision of player beliefs and expectations and affect strategizing.

It is this thread, that of online strategizing that takes into account the temporal extensive structure, that we take up in this article, with an emphasis on computational considerations as suggested above. That is, we move away from pre-game selection of strategies to strategies constructed during course of play by an automaton. This leads us to a study of compositional structure in strategies. The restriction to finite state devices highlights the *memory* needed by a computationally limited player who must select observations to record during play and can see only a part of the future, as opposed to an agent with unbounded memory who has access to the entire past and future.

When the game is given as a finite tree, it seems pointless to talk of an agent with limited memory, since any finite state device can code up the tree in its memory. However, if the game tree is sufficiently large, such as in the game of Chess, the player would see only an abstraction of the tree. In general, if the temporal extensive structure of the game is large relative to the memory capability of players, strategizing by the player is affected by the abstraction of the future. Indeed, as van Benthem [8] argues, in any game where we only know top-level structure, a ‘fuzzy’ view of the future is unavoidable. In the study of game playing programs in Artificial Intelligence, it is customary to work with local game analysis, along with general heuristic values for unanalyzed later parts of the game [25].

The remark about abstracting the future raises an important question: how should such a game be represented, and how can the player strategize in such a game? The answer we provide is in the spirit of Rubinstein [43]: a game with a “very long horizon” is best modelled as a game with *infinite horizon*. According to Rubinstein [43], a game should be seen as a description of relevant factors as perceived by the players, and not as a presentation of the physical rules of the game. If a player strategizes in a game as if it has an unbounded horizon, an infinite tree represents the game better.<sup>2</sup> Therefore we present the game arena as a finite graph and the game as an infinite tree obtained by unfolding the graph. The game arena can be seen as a finite presentation of a set of rules; players look for patterns, and based on the occurrence of patterns and past information (recorded selectively), they make choices. Such a consideration leads us to the realm of **regular infinite games**.

In our model, a player enters the game arena with information on the game structure and on other players’ skills, as well as an initial set of possible strategies to employ. As the play progresses, she makes observations and accordingly revises strategies, *switches* from one to another, perhaps even devises new strategies that she hadn’t considered before. The dynamics of such interaction eventually leads to some strategies being eliminated, and some becoming stable. It is this process that we wish to study using automata.

This chapter continues the line of work initiated in [42] on a compositional structure in strategies realised by automata. Strategy switching is emphasized here, and the rationale for why a player following a strategy might switch to another one online. We study another form of switching as well: a player who cannot decide between two strategies may, rather than committing to one or the other, choose to go back and forth between the two, following either of the two nondeterministically. This can be seen as a nondeterministic analogue of *mixed strategies*, without specifying a probability distribution. In the end, the player is not following either of the two strategies but a nondeterministic mix of both. Thus the central premise of this chapter is that exploring structure in strategies is worthwhile from a logical perspective, and that automata theory is helpful in this regard.

---

<sup>2</sup> Indeed the considerations of online strategizing and compositional structure seem more relevant for such temporally large games. Arguably, for sufficiently small games, pre-game deliberation might suffice.

Infinite games have a long history, and their study has led to some beautiful set theory and topology. Regular infinite games have received attention from theoretical computer science in the last two decades, principally due to their relevance to game models of reactive system design and verification. Notions such as determinacy of win/lose games, equilibria, and computation of equilibrium strategies have been worked out for these games [22]. A central thread of this research is the adequacy of *finite memory* strategies for standard solution concepts.

What we discuss in this article is the realisation of compositional strategies, involving switching and response to other players' behaviour, as finite state transducers, thus explicating their memory structure. In itself, this is not surprising, given the limited expressive power of the specification language of strategies. However, the construction is interesting since it shows ways by which strategies may be combined algorithmically.

What questions can one study in such a model? Once player objectives and preferences are specified, natural questions relate to existence of best response, synthesis of such strategies etc., and for a logical analysis, axiomatic characterizations of formulas specifying how a strategy may ensure an outcome for a player. These questions are addressed in [40] and [42] and not taken up here. Instead we focus only on implications of strategy switching. When players may switch from one strategy to another, there is an associated *dynamics* of strategies and patterns in game evolution. Is it the case that, after online exploration of opponents' behaviour, a player settles down to a specific strategy and does not switch any further? This can be seen as a form of stability in strategizing.

In fact, when a player no longer considers a strategy at all after a point in game evolution, this may have consequences as well. When all players stop considering a strategy similarly, it may simply get eliminated from the game, leading to a new game. When all strategies available to a player are eliminated, a player may be forced out of the game. Such questions are especially relevant in the context of bargaining and negotiations, as evidenced in many political contexts. Questions of this nature were studied in [36]. We do not study such dynamic game forms, but merely point out that strategy switching can lead to many interesting stability questions and that realization of strategies by finite state transducers can be used to answer such questions.

In what follows, we give a brief introduction to infinite games, and since they need to be finitely specified, finite graph representation of (regular) infinite games. Then we discuss the structure of strategy specifications, and we see that switching introduces some conceptual difficulty. We then show how strategy specifications can be implemented using automata. We discuss related work at the end of the article.

## 2 Infinite Games

In this section we give an introduction to infinite games on finite graphs. The main aim here is to set up the preliminaries and point to the rich literature

on automata as strategies in regular infinite games. These automata, in turn, provide us a tool for strategy composition that we take up later on. *Automata, Logics and Infinite Games* [22] is a good source for an introduction to infinite games on finite graphs, as well as strategies as finite state automata.

### 2.1 Game Model

When we consider games of unbounded duration, a natural question is how such a game is presented. While this question is generally not easy to answer, in the special case of *regular* infinite games, we can conceive of the game tree as an unfolding of a finite graph with cycles. Thus games on finite graphs give a finite presentation for games of unbounded duration: we call these game arenas.

**Game Arena.** A game arena is a structure  $\mathcal{G} = (W, \rightarrow, w_0, \lambda)$  over a finite set of moves (or actions)  $\Sigma$ , where:

- $W$  is a finite set of game positions.
- $\rightarrow : W \times \Sigma \times W$  is an edge relation which satisfies the condition: if  $(w, a, w') \in \rightarrow$  and  $(w, a, w'') \in \rightarrow$  then  $w' = w''$ .
- $w_0 \in W$  is the initial game position.
- $\lambda : W \rightarrow N$  is the turn function which associates each game position with a player.

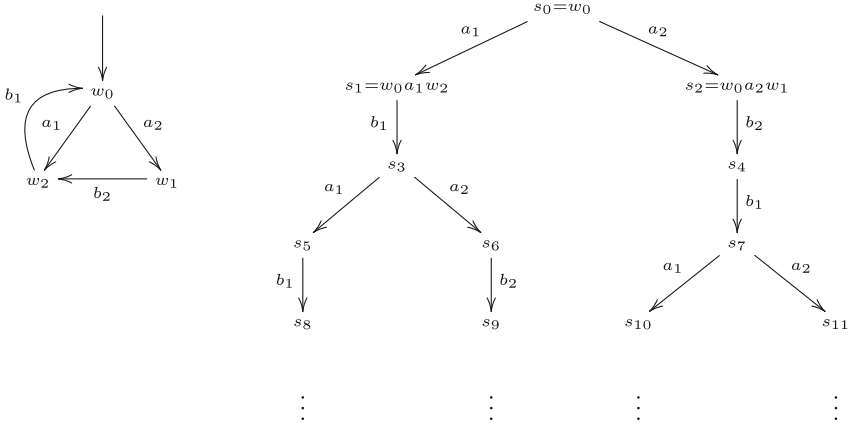
We often denote  $(w, a, v) \in \rightarrow$  by  $w \xrightarrow{a} v$ . For a node  $w$ , let  $\vec{w} = \{v \in W \mid w \xrightarrow{a} v \text{ for some } a \in \Sigma\}$ . For technical convenience, we assume that for all  $w \in W$ ,  $\vec{w} \neq \emptyset$ , that is, there are no ‘dead-ends’. For  $i \in N$ , let  $W^i = \{w \in W \mid \lambda(w) = i\}$ . A play in the game arena  $\mathcal{G}$  starts by placing a token on  $w_0$  and proceeds as follows: at any stage if the token is at a position  $w$  and  $\lambda(w) = i$  then player  $i$  picks an action which is enabled for her at  $w$ , and the token is moved to  $v$  where  $w \xrightarrow{a} v$ . Formally, a play in  $\mathcal{G}$  is an infinite path  $\rho = w_0 a_0 w_1 a_1 \dots$  such that for all  $j \geq 0$ ,  $w_j \xrightarrow{a_j} w_{j+1}$ . Let  $Plays(\mathcal{G})$  denote the set of all plays in  $\mathcal{G}$ . A history  $h$  is a finite path in the arena. For a history  $h = w_0 a_0 w_1 a_1 \dots w_k$  we denote by  $last(h)$  the last element of  $h$  that is  $last(h) = w_k$ .

A subarena of  $\mathcal{G}$  is a subgraph of  $\mathcal{G}$  with no dead-ends.

**Extensive Form Game.** The (infinite) extensive form game tree  $T_{\mathcal{G}}$  associated with  $\mathcal{G}$  is obtained by the tree unfolding of  $\mathcal{G}$  which we define below.

Given a game arena  $\mathcal{G} = (W, \rightarrow, w_0, \lambda)$ , the tree unfolding of  $\mathcal{G}$  is the least tree structure  $T_{\mathcal{G}} = (S, \Rightarrow, s_0, \hat{\lambda})$  where  $S \subseteq (W \times \Sigma)^* W$  and  $\Rightarrow : S \times \Sigma \rightarrow S$  satisfies the condition:

- $w_0 \in S$ .
- If  $s = (w_0, a_0) \dots w_k \in S$  and  $w_k \xrightarrow{a} w'$  then  $s' = (w_0, a_0) \dots (w_k, a) w' \in S$  and  $s \xRightarrow{a} s'$ .



**Fig. 1.** Game arena and tree unfolding

Further, for a node  $s = (w_0, a_0) \dots w_k \in S$ ,  $\widehat{\lambda}(s) = \lambda(w_k)$ .

Figure 1 illustrates a game arena and its tree unfolding. A node  $s$  in  $T_{\mathcal{G}}$  denotes a finite partial play in the arena. Thus  $s = w_0 a_0 w_1 a_1 \dots w_k$  and  $last(s) = w_k$ . Note that for any node  $s$ ,  $last(s) \in W$ . The prefix relation  $s \sqsubseteq s'$  specifies that the node  $s'$  is reachable from  $s$  in the tree. For  $i \in N$ , let  $S^i = \{s \in S \mid \widehat{\lambda}(s) = i\}$ .

## 2.2 Strategies

A strategy  $\mu^i$  for player  $i$  specifies for each partial play ending in a game position of player  $i$  which action to choose. Thus  $\mu^i$  is a map  $\mu^i : (W \times \Sigma)^* W^i \rightarrow \Sigma$ . A play  $\rho : w_0 a_0 w_1 \dots$  is said to be consistent with a strategy  $\mu^i$  if for all  $j, 0 \leq j$ , we have  $\lambda(w_j) = i$  implies  $\mu^i(\rho) = a_j$ . A strategy  $\mu^i$  can also be viewed as a labelled tree  $T_{\mathcal{G}}^{\mu^i} = (T_{\mathcal{G}}, m)$  where  $m : S^i \rightarrow \Sigma$  such that for all  $s \in S^i$ ,  $m(s) = \mu^i(s)$ . Let  $Strat^i$  denote the set of all strategies of player  $i$  in  $\mathcal{G}$ .

A strategy profile  $\bar{\mu}$  consists of a tuple of strategies, one for each player. For  $i \in N$ , let  $\mu^{-i} = (\mu^1, \dots, \mu^{i-1}, \mu^{i+1}, \dots, \mu^n)$ . A play  $\rho$  is consistent with a strategy profile  $\bar{\mu}$  if  $\rho$  is consistent with  $\mu^i$  for all  $i \in N$ . It is easy to check that for a strategy profile  $\bar{\mu}$ , there exists a unique play in  $\mathcal{G}$  which is consistent with  $\bar{\mu}$ . This can be thought of as the play generated by  $\bar{\mu}$ . We denote this play by  $\rho_{\bar{\mu}}$ .

Note that according to the definition, a strategy can in principle depend on the complete history of play and in general need not be computable. For computationally bounded players it is not possible to implement or even choose to play such an arbitrarily defined strategy. In this context, the following two types of strategies are of particular interest:

- **Memoryless** (positional) strategies: These are strategies for which the next move depends only on the current game position and not on the history of

play. Thus the map  $\mu^i : W^i \rightarrow \Sigma$  prescribes the same action for all partial plays ending at the same game position. That is, for all  $\rho, \rho'$  such that  $last(\rho) = last(\rho')$ ,  $\mu^i(\rho) = \mu^i(\rho')$ .

- **Bounded memory strategies:** These are strategies where the dependence of the next move to the history of the play can be kept track of by a finite set of states. Such strategies can be represented using finite state machines equipped with an output function.

Once again, it needs to be noted bounded memory strategies form a subclass of the space of possible strategies. Their attractiveness lies in having a presentation by way of finite state transducers.

**Finite State Transducer.** Given a game arena  $\mathcal{G} = (W, \rightarrow, w_0, \lambda)$ , a finite state transducer for player  $i$  is a tuple  $\mathcal{A}^i = (Q, \delta, o, I)$  where

- $Q$  is a finite set of states.
- $\delta : Q \times W \times \Sigma \rightarrow 2^Q$  is a nondeterministic transition function.
- $o : Q \times W^i \rightarrow \Sigma$  is the output function of the transducer.
- $I \subseteq Q$  is the set of initial states.

Let  $\mu^i$  be a strategy of player  $i$  and  $T_{\mathcal{G}}^{\mu^i} = (T_{\mathcal{G}}, m)$  be the corresponding strategy tree. A run of  $\mathcal{A}^i$  on  $T_{\mathcal{G}}^{\mu^i}$  is a  $Q$  labelled tree  $T = (S, \Rightarrow, s_0, m, f)$ , where  $f : S \rightarrow Q$  is a map defined as follows:  $f(s_0) = q_0$ , and for any  $s_k$  where  $s_k \xrightarrow{a} s'_k$ , we have  $f(s'_k) \in \delta(f(s_k), last(s_k), a_k)$ . A  $Q$  labelled tree  $T$  is accepted by  $\mathcal{A}^i$  if for every tree node  $s \in S$  where  $s \in S^i$ ,  $m(s) = a$  implies  $o(f(s), last(s)) = a$ .

We say a strategy tree  $T_{\mathcal{G}}^{\mu^i}$  is accepted by  $\mathcal{A}^i$  if  $\mathcal{A}^i$  has an accepting run on  $T_{\mathcal{G}}^{\mu^i}$ . For a state  $q$  and a tree node  $s$ , we often use the notation  $o(q, s)$  to denote  $o(q, last(s))$ . If the transition function of  $\mathcal{A}^i$  is deterministic then it is easy to see that the strategy generated by  $\mathcal{A}^i$  is unique.

Given a history  $s = w_0 a_0 w_1 \dots w_k$ , a strategy  $\mu^i[s]$  for player  $i$  after  $s$  is a map  $\mu^i[s] : s(W \times \Sigma)^* W^i \rightarrow \Sigma$ . Let  $T_{\mathcal{G}}^{\mu^i[s]}$  be the strategy tree corresponding to  $\mu^i[s]$ . We denote the set of all strategies for player  $i$  after  $s$  by  $Strat^i(s)$ . A run of  $\mathcal{A}^i$  on  $T_{\mathcal{G}}^{\mu^i[s]}$  is a  $Q$  labelled tree  $T[s]$  as defined earlier, with root node  $s$ . We say that the  $Q$  labelled tree  $T[s]$  is accepted by  $\mathcal{A}^i$  if for every tree node  $s'$  where  $s \sqsubseteq s'$  and  $\hat{\lambda}(s') = i$ , if  $m(s') = a$  then  $o(f(s'), last(s')) = a$ . We denote by  $Lang(\mathcal{A}^i, s)$  the set of all strategy trees  $T_{\mathcal{G}}^{\mu^i[s]}$  which are accepted by  $\mathcal{A}^i$ .

### 2.3 Objectives

**Players' Objectives.** In games where the outcome is binary and every player either wins or loses, a natural way of specifying players' objectives is to associate with each player  $i \in N$  a set  $\Phi_i \subseteq Plays(\mathcal{G})$  with the interpretation that a play  $\rho$  is winning for player  $i$  iff  $\rho \in \Phi_i$ . Note that the objectives could be overlapping,

i.e., for player  $i$  and  $j$  it is possible to have  $\Phi_i \cap \Phi_j \neq \emptyset$ . A game is then specified as the pair  $G = (\mathcal{G}, \{\Phi_i\}_{i \in N})$ .

Given a game  $G = (\mathcal{G}, \{\Phi_i\}_{i \in N})$ , we call a strategy  $\mu^i$  for player  $i$  winning if for all plays  $\rho$  conforming to  $\mu^i$ ,  $\rho \in \Phi_i$ . In other words, a strategy  $\mu^i$  is winning if for every profile  $\mu^{-i}$  of the other players, the play  $\rho$  generated by  $\bar{\mu}$ , we have:  $\rho \in \Phi_i$ .

Two player games are games in which the number of players are restricted to two, i.e. the set of players  $N = \{1, 2\}$ . For such games, we often use the notation  $i$  and  $\bar{i}$  to denote the players where  $\bar{i} = 2$  when  $i = 1$  and  $\bar{i} = 1$  when  $i = 2$ . Two player game in which the players' objectives are strictly complementary are called zero sum games. Formally, these are games where the set of plays can be partitioned into disjoint sets  $\Phi_1$  and  $\Phi_2$ . This implies that when a play  $\rho$  is winning for player  $i$ , it is not winning for player  $\bar{i}$ .

In general, when games are not strictly winning or losing for players, we have a preference order  $\preceq^i \subseteq (Plays(\mathcal{G}) \times Plays(\mathcal{G}))$  for each player  $i \in N$ . Note that this generalizes winning sets, since a winning set  $\Phi_i$  defines a preference in the obvious manner: player  $i$  prefers plays in  $\Phi_i$  over those in  $Plays(\mathcal{G}) \setminus \Phi_i$ .

Note that the specification of objectives is infinite, and we are interested in *finitely presented* objectives, to be described below.

**Solution Concepts.** In the case of win/lose games the natural notion of solving a game is to determine whether a player has a winning strategy. A two player zero sum game  $G$  is said to be **determined** if there exists a player  $i \in \{1, 2\}$  such that  $i$  has a winning strategy in  $G$ .

One of the early results which helped highlight the relationship between determinacy and the topological properties of the winning set  $\Phi$  is the Gale-Stewart theorem [17]. Here the game arena is understood to be an infinite graph, but the other notions remain the same. The theorem asserts that every game where  $\Phi$  is an open set is determined. This result was later improved by Martin [30] to show determinacy for games with Borel objectives: a large class of subsets of topological spaces, of importance to mathematical analysis. The important consequence for games on finite arenas is that all regular objectives to be considered below are determined.

For games with overlapping objectives, a popular solution concept is that of the Nash equilibrium.

- Given a profile  $\bar{\mu}$ , the strategy  $\mu^i$  of player  $i$  is a best response to  $\mu^{-i}$  if  $\forall \nu^i \in Strat^i, \rho_{(\nu^i, \mu^{-i})} \preceq^i \rho_{\bar{\mu}}$ .
- A strategy profile  $\bar{\mu}$  is said to be in equilibrium if for all  $i \in N$ ,  $\mu^i$  is a best response to  $\mu^{-i}$ .

Existence of Nash equilibria in infinite games is much less studied. Indeed, it is unclear whether other notions of equilibria might be more appropriate for games of unbounded duration.



**Finitely Presented Objectives.** Since we are interested in resource limited players, the preference orders themselves need to be finitely presented. A player's decision to prefer one play over another is limited by her observations of the two plays. With finite resources, a player can observe either only a finite prefix of the infinite play, or repetitive behaviour (if any). When the game arena is finite, every infinite play must settle down eventually within a strongly connected component of the arena, and hence preferences over plays can be presented as an ordering over connected components. In general, player's preferences are associated with loops or cycles in the arena. Since a player records her observations as play proceeds she may remember how play arrived at a specific loop as well, or how many times a particular loop was traversed before entering another. However, these observations and counting are limited by the player's resources.

These observations lead us to the notion of an evaluation automaton for each player. We present this for winning sets: a play is in the winning set of player  $i$  only if it is accepted by the evaluation automaton for player  $i$ .

A finite deterministic evaluation automaton over the input alphabet  $W \times \Sigma$  is a tuple  $\mathbb{A} = (Q, \Delta, r_0, Acc)$  where

- $Q$  is a finite set of states.
- $\Delta : (Q \times W \times \Sigma) \rightarrow Q$  is the transition function.
- $r_0 \in Q$  is the initial state.
- $Acc$  specifies the acceptance condition.

The run of  $\mathbb{A}$  on an infinite sequence  $\rho : w_0 a_0 w_1 \dots$  is a sequence of states  $\varphi_\rho : r_0 r_1 \dots$  such that for all  $j \geq 0$ ,  $r_{j+1} = \Delta(r_j, w_j, a_j)$ . Let  $Inf(\varphi_\rho)$  denote the set of states occurring infinitely often in  $\varphi$ . The most commonly used acceptance conditions are the following requirements on  $Inf(\varphi)$ :

- **Reachability condition:** For a set of designated states  $R \subseteq Q$ ,  $\Phi_i = \{\rho = w_0 a_0 w_1 a_1 \dots \mid \exists k \text{ with } w_k \in R\}$ .
- **Büchi condition [12]:** For a set of “good states”  $B \subseteq Q$ ,  $Inf(\varphi_\rho) \cap B \neq \emptyset$ . In other words, some final state occurs infinitely often in the run  $\varphi_\rho$ .
- **Muller condition [33]:** For a family  $\mathcal{F} \subseteq 2^Q$ ,  $\bigvee_{F \in \mathcal{F}} Inf(\varphi_\rho) = F$ . This requires that the set of states occurring infinitely often in the run  $\varphi_\rho$  forms a set in  $\mathcal{F}$ .
- **Parity condition:** Let  $c : Q \rightarrow \{1, \dots, k\}$  (where  $k \in \mathbb{N}$ ). The run  $\rho$  is accepting iff  $\min\{c(r) \mid r \in Inf(\varphi_\rho)\}$  is even.

In terms of expressiveness it is known that a Parity condition can be translated to a Muller condition and vice-versa but this can lead to a blow-up in the size of the arena. The deterministic Büchi condition is strictly less expressive than the Muller or the Parity condition and the Reachability condition is strictly less expressive than the Büchi condition.

**Algorithmic Questions.** Now that we have finite game arenas and finite presentation of objectives, we can ask algorithmic questions on them.

- **Verification question:** Given a game  $G = (\mathcal{G}, \{\Phi_i\}_{i \in N})$  and player  $i$ , does player  $i$  have a winning strategy in  $G$ ?

- **Synthesis question:** Given a game  $G = (\mathcal{G}, \{\Phi_i\}_{i \in N})$  and player  $i$ , is it possible to construct a winning strategy for player  $i$  (when it exists)?

The determinacy of two player zero sum games with regular objectives follows from Martin’s theorem since regular objectives fall in the second level of the Borel hierarchy. However, this result does not suffice for algorithmic purposes since the winning strategies employed depends on the complete history of the play, and therefore require infinite memory. The seminal result of Büchi and Landweber [13] showed that when players’ objectives are presented as Muller conditions, the winner can be determined and that the winning strategy can be effectively synthesised in bounded memory strategies. For parity games on the other hand, memoryless strategies suffice for winning [16, 31]. In other words, these results showed that it is possible to solve the verification and synthesis questions for games with regular objectives.

The existence of Nash equilibrium for games with regular win-lose objectives follows from the result of [15]. The main idea here is the effective use of **threat strategies** whereby a player deviating from the equilibrium profile is punished by others to receive the outcome which she can guarantee on her own. The existence of sub-game perfect equilibrium [44] for games with binary objectives was shown in [23]. When we consider games with overlapping objectives, existence of Nash equilibrium for games with Muller conditions was shown by [37].

Note that these solution concepts raise the question of *existence* of optimal strategies: either winning strategies or equilibrium profiles, and perhaps realising these (offline) strategies by automata. Returning to the motivation we presented earlier, a natural question to consider is: given a game arena and a finite presentation of player objectives, how can a player select and compose strategies online, to achieve an outcome? This requires us to look for structure in strategies, which is what we take up next.

The use of finite state automata to study strategies in games of unbounded duration underlies this entire body of work, and [22] is a good source for an introduction to this methodology. However, the literature focusses largely on win/lose games, as the intended applications are for system and design and verification: the system being designed is in a game situation against a hypothetical environment, and behaviour according to intended specification constitutes a win for the system. Extensions to multi-player games typically consider a coordination game: a multi-component system where all components coordinate against the hypothetical environment.

Our departure here is the use of automata to represent the process of strategizing by bounded memory players in games of unbounded duration. In a theory of play, we need to describe how players observe play, record their observations and strategize on-line. Automata provide a representation of this process.

### 3 Strategizing by Players

We now shift our focus to studying the game from the players’ viewpoint, and look for a theory in which players start the game with an initial set of strategies

and compose from them by switching between them depending on their interpretation of course of play. In the process they generate more and more complex strategies. We use tools from logic and automata theory for this study.

### 3.1 Partial Strategies

Not only do resource-bounded players strategise dynamically, their strategies are also partial, consisting of partial plans. Players, in general, cannot conceive of every possible situation right at the beginning of the game and hence plan only partially. A partial strategy specifies a subset of the available moves for every history, or in other words, it restricts some of the available moves. Formally, given a game  $\mathcal{G} = (W, \rightarrow, w_0, \lambda)$  a partial strategy is a function  $\nu : (W \times \Sigma)^* W^i \rightarrow (2^\Sigma \setminus \emptyset)$ . Let  $Pstrat^i$  denote the set of all partial strategies of player  $i$ . A partial strategy  $\nu$  of player  $i$  can also be viewed as a labelled tree  $T_{\mathcal{G}}^\nu = (T_{\mathcal{G}}, m^\nu)$  where  $m^\nu : S^i \rightarrow 2^\Sigma$  such that, for every node  $s \in S^i$ , we have  $m^\nu(s) = \nu(s)$ .

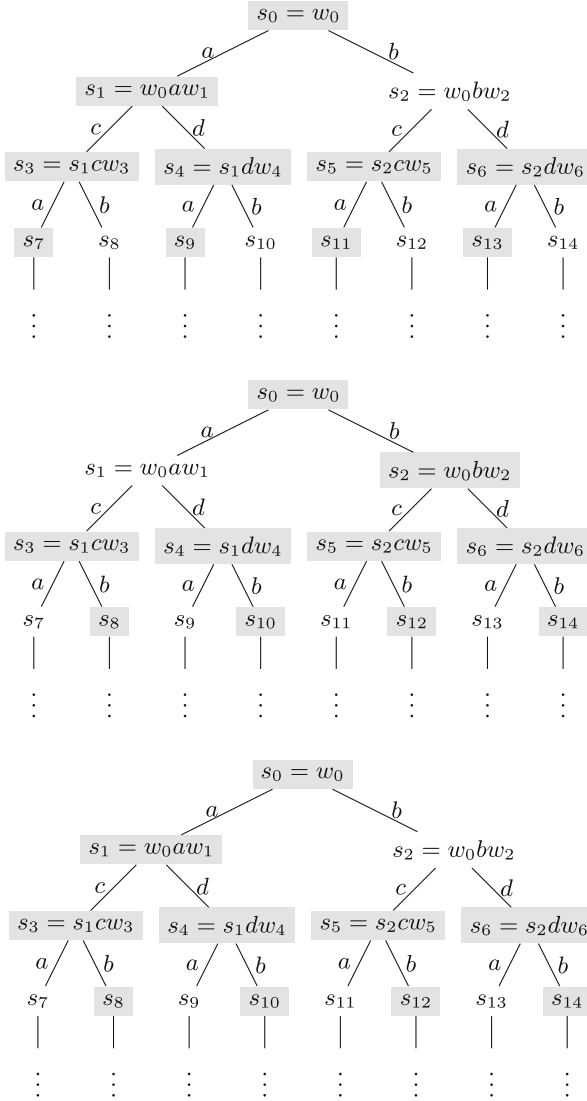
A partial strategy tree  $T_{\mathcal{G}}^\nu = (T_{\mathcal{G}}, m^\nu)$  for player  $i$  may be viewed as a set  $\mathcal{T}_{\mathcal{G}}^\nu$  of total strategy trees where a total strategy tree  $T = (T_{\mathcal{G}}, m) \in \mathcal{T}_{\mathcal{G}}^\nu$  if and only if for every  $s \in S^i$ ,  $m(s) \in m^\nu(s)$ . It is easy to see that a finite memory partial strategy can thus be presented in terms of a transducer  $\mathcal{A}^\nu$  such that  $Lang(\mathcal{A}^\nu) = \mathcal{T}_{\mathcal{G}}^\nu$ .

Given a history  $s$ , it is also convenient to define a partial strategy  $\nu$  “after”  $s$ , denoted  $\nu[s]$  which is formally a map  $\nu[s] : s(W \times \Sigma)^* W^i \rightarrow (2^\Sigma \setminus \emptyset)$ . Let  $Pstrat^i(s)$  denote the set of all partial strategies after  $s$  of player  $i$ . The corresponding partial strategy tree is denoted by  $T_{\mathcal{G}}^{\nu[s]}$ .

### 3.2 Switching from One Strategy to Another

Suppose there are two players 1 and 2 and their set of actions are  $\{a, b\}$  and  $\{c, d\}$  respectively. Consider two strategies of Player 1,  $\mu_a$  and  $\mu_b$ .  $\mu_a$  specifies the action  $a$  at every game node and  $\mu_b$  specifies the action  $b$  at every game node. Now suppose Player 1 plays strategy  $\mu_a$  for the first move and then plays strategy  $\mu_b$  on the subtree from her next move. The resulting prescription  $\mu$  (say) is thus also a strategy for player 1. See Fig. 2.

Suppose  $\mu_1$  and  $\mu_2$  are two strategies of player  $i$  and suppose that she follows  $\mu_1$  for at most  $k$  moves of the game (here, by convention, a move can be either a Player 1 move or a Player 2 move) and then switches to  $\mu_2$ . We denote the resulting set of strategies by  $\mu_1^k \mu_2$  and the resulting set of strategy trees by  $\mathcal{T}_{\mathcal{G}}^{\mu_1^k} \mathcal{T}_{\mathcal{G}}^{\mu_2}$ . Every tree  $T$  in the set  $\mathcal{T}_{\mathcal{G}}^{\mu_1^k} \mathcal{T}_{\mathcal{G}}^{\mu_2}$  can be viewed to have been constructed as follows. Let  $T_{\mathcal{G}}^{\mu_1}(k)$  be a finite subtree of  $T_{\mathcal{G}}^{\mu_1}$  such that every branch of  $T_{\mathcal{G}}^{\mu_1}(k)$  is of depth at most  $k$ . Let  $leaves(T_{\mathcal{G}}^{\mu_1}(k))$  denote the set of leaves of this tree. For every  $s \in leaves(T_{\mathcal{G}}^{\mu_1}(k))$ , we attach  $T_{\mathcal{G}}^{\mu_2}(s)$  to  $s$ . The operation can also be lifted to an arbitrary set of strategy trees,  $\mathcal{T}_1$  and  $\mathcal{T}_2$  (say). We denote the resulting set of trees as  $\mathcal{T}_1^k \mathcal{T}_2$ .



**Fig. 2.** The strategy  $\mu_a$ ; the strategy  $\mu_b$ ; and  $\mu_a^2 \mu_b$ . Note that, by definition, since the second move is by Player 2,  $\mu_a^2 \mu_b$  and  $\mu_a \mu_b$  define the same strategies

### 3.3 Specification of Strategy Composition

To talk about the outcomes of the game, we introduce a countable set of propositions  $\mathcal{P}_i$  for every player  $i$  and a valuation function  $val_i$  which are evaluated on the vertices of the arena. These propositions code up the outcomes of the game in the lines of [10].  $val_i$  is lifted on the nodes of the game tree  $T_G$  in the usual manner.

**Syntax.** By an ‘atomic strategy’ of player  $i$  we mean a strategy which dictates her to play the same action at all positions. We denote atomic strategies by  $\phi_a$ ,  $a \in \Sigma_i$ .

The strategy set  $\Phi_i$  of player  $i$  is obtained by combining her atomic strategies using some operators. Let  $\Phi_i^-$  be defined as:

$$\Phi_i^- := \phi_a, a \in \Sigma_i \mid \phi_1 \cup \phi_2 \mid \phi_1 \hat{\ } \phi_2 \mid \phi_1 + \phi_2$$

The intuitive meaning of the strategy building operators is explained as follows:

- $\phi_a$ ,  $a \in \Sigma_i$  is the atomic strategy where player  $i$  plays the action  $a$  at each move.
- $\phi_1 \cup \phi_2$  means that the player plays according to the strategy  $\phi_1$  or the strategy  $\phi_2$ .
- $\phi_1 \hat{\ } \phi_2$  means that the player plays according to the strategy  $\phi_1$  and then after some history, switches to playing according to  $\phi_2$ . The position at which she makes the switch is not fixed in advance.
- $\phi_1 + \phi_2$  says that at every point, the player can choose to follow either  $\phi_1$  or  $\phi_2$ .

**The Test Operator.** Let  $\Phi_i$  be the strategies built from player  $i$ ’s atomic strategies applying the operators of  $\Phi_i^-$  and also the *test operator*  $\psi? \phi$ . Intuitively,  $\psi? \phi$  says that at every history, the player tests if the property  $\psi$  holds of that history. If it does then she plays according to  $\phi$ .

What is the observable condition  $\psi$  that player  $i$  checks for? We think of these conditions as past time formulas of a simple tense logic over the atomic set of observables  $\mathcal{P}_i$ . More specifically,  $\psi$  belongs to the following syntax:

$$\Psi_i := p \in \mathcal{P}_i \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \langle a \rangle^- \psi \mid \langle a \rangle^+ \psi \mid \psi_1 \mathcal{S} \psi_2 \mid j? \phi, j \neq i, \phi \in \Phi_j^-$$

Intuitively,  $j? \phi$  is the test where player  $i$  checks if player  $j$  is playing according to  $\phi$ . Now as the observables of player  $j$ ,  $\mathcal{P}_j$  and their combinations are private to her, player  $i$  cannot reason based on them. Hence  $\phi$  comes from  $\Phi_j^-$  rather than the entire  $\Phi_j$ .

The usual operators  $\ominus\psi$  (previous),  $\bigcirc\psi$  (next),  $\diamond\psi$  (sometime in the past) and  $\boxminus\psi$  (throughout the past) are defined as  $\ominus\psi \equiv \bigvee_{a \in \Sigma} \langle a \rangle^- \psi$ ,  $\bigcirc\psi \equiv \bigvee_{a \in \Sigma} \langle a \rangle^+ \psi$ ,  $\diamond\psi \equiv \top \mathcal{S} \psi$  and  $\boxminus\psi \equiv \neg \diamond \neg \psi$ .

An observable  $\psi \in \Psi_i$  is interpreted over the nodes of the game tree. Formally, for a node  $s \in T_{\mathcal{G}}$ ,  $s \models \psi$  is defined inductively as:

- $s \models p$  iff  $p \in \text{val}_i(s)$ .
- $s \models \neg\psi$  iff  $s \not\models \psi$ .
- $s \models \psi_1 \vee \psi_2$  iff  $s \models \psi_1$  or  $s \models \psi_2$ .
- $s \models \langle a \rangle^- \psi$  iff  $s = s'a$  and  $s' \models \psi$ .
- $s \models \langle a \rangle^+ \psi$  iff for all  $sa \in T$ ,  $sa \models \psi$ .

- $s \models \psi_1 \mathcal{S} \psi_2$  iff  $\exists s' \sqsubset s$  such that  $s' \models \psi_2$  and  $\forall s'' : s' \sqsubset s'' \sqsubseteq s, s'' \models \psi_1$ .
- $s \models j? \phi$  iff there exists  $T'$  such that  $T' \in \llbracket \phi, \epsilon \rrbracket_{T_G}$  and  $s \in T'$ .

where  $\llbracket \phi, \epsilon \rrbracket_{T_G}$  will be defined shortly.

Note that in the syntax of  $\Psi_i$ , we do not have the corresponding indefinite future time operator  $\mathcal{U}$ . This is to reflect our view of strategizing as memory limited observations of the past, with bounded lookahead, as expressed by the operator  $\langle a \rangle^+ \psi$ .

**Semantics.** We now give the formal semantics of the strategy specifications. Given the game tree  $T_G = (S, \Rightarrow, s_0)$ , the semantics of a strategy specification  $\phi \in \Phi_i$  is a function  $\llbracket \cdot \rrbracket_{T_G} : \Phi_i \times S \rightarrow 2^{Strat_i}$ . That is, each specification at a node  $s$  of the game tree is associated with a set of total strategy trees after history  $s$ .

For any  $s \in S$ ,  $\llbracket \cdot \rrbracket_{T_G}$  is defined inductively as follows:

- $\llbracket \phi_a, s \rrbracket_{T_G} = T^{\phi_a[s]}$  where for all  $s' \in S$  such that  $s \sqsubseteq s'$  and  $\widehat{\lambda}(s') = i$ , we have  $\phi_a[s](s') = a$ .
- $\llbracket \phi_1 \cup \phi_2, s \rrbracket_{T_G} = \llbracket \phi_1, s \rrbracket_{T_G} \cup \llbracket \phi_2, s \rrbracket_{T_G}$ .
- $\llbracket \phi_1 \hat{\cup} \phi_2, s \rrbracket_{T_G} = \llbracket \phi_1, s \rrbracket_{T_G} \cup \bigcup_{l \geq |s|} (\llbracket \phi_1, s \rrbracket_{T_G} \overset{l}{\llbracket \phi_2, \epsilon \rrbracket_{T_G}}$ .
- $\llbracket (\phi_1 + \phi_2), s \rrbracket_{T_G} : T = (T_G, m) \in \llbracket (\phi_1 + \phi_2), s \rrbracket_{T_G}$  if and only if there exists  $T^1 = (T_G, m_1) \in \llbracket \phi_1, s \rrbracket_{T_G}$  and  $T^2 = (T_G, m_2) \in \llbracket \phi_2, s \rrbracket_{T_G}$  such that the following condition is satisfied:
  - for all  $s' \in S$  such that  $s \sqsubseteq s'$  and  $\widehat{\lambda}(s') = i$ ,  $m(s') = m_1(s')$  or  $m(s') = m_2(s')$ .
- $\llbracket \psi? \phi, s \rrbracket_{T_G} : T = (T_G, m) \in \llbracket \psi? \phi, s \rrbracket_{T_G}$  if and only if  $s \in T$  and there exists  $T' = (T_G, m') \in \llbracket \phi, s \rrbracket_{T_G}$  such that the following condition is satisfied:
  - for all  $s' \in S$  such that  $s \sqsubseteq s'$  and  $\widehat{\lambda}(s') = i$ , if  $\psi$  holds at  $s'$  then  $m(s') = m'(s')$  and if  $\psi$  does not hold at  $s'$  then  $m(s') = \vec{w}$  where  $w = last(s')$ .

A crucial difference between the two kinds of switching operators needs emphasis.  $\phi_1 \hat{\cup} \phi_2$  specified a one-time nondeterministic switching from following  $\phi_1$  to following  $\phi_2$  whereas  $\phi_1 + \phi_2$  specifies switching nondeterministically back and forth between the two. The latter can be seen as a qualitative specification of a *mixed strategy* for extensive form games: a player mixes the two strategies nondeterministically during course of play. The crucial difference is the lack of any probability distribution; however, such an extension is easy to conceive of: let  $\phi_1 +_r \phi_2$  specify that at any point of time, if the player is playing  $\phi_i$ , she switches to  $\phi_{3-i}$  with probability  $r$  (and continues playing  $\phi_i$  with probability  $1-r$ ). This suggests that we can build compositional structure in mixed strategies as well.

### 3.4 An Example

Consider the game of tennis. A player, poised to serve, considers for a moment: Should I serve wide, or down the ‘T’? Should I serve deep, or short? When I

last served deep and wide, he hit it onto the net. Should I try it again, or has he learnt now? The opponent, on his part, considers as he takes his stance: if he serves short, should I go for a cross-court shot and reveal my strength off that flank? If I do it too often, he will not serve such balls at all. Should I play it safe?

Let the player's set of atomic strategies be given as  $\Sigma_{player} = \{\sigma_{short}, \sigma_{deep}, \sigma_{wide}, \sigma_T\}$  which corresponds to serving short, deep, wide and down-the-T respectively.

Let  $p_{(short,net)}$  be the observable which says that the outcome of short serve is a return into the net. Then the following specification says that the player keeps serving short and wide till the opponent is able to return (does not hit into the net).

$$- \neg \hat{\diamond} p_{(short,shot)}? (\sigma_{short} \cap \sigma_{wide})$$

The specification  $\sigma_{short} \hat{\sim} \sigma_{wide} \hat{\sim} \sigma_{deep}$  for the player says that he starts by serving short and after some point he switches to serving wide and again switches but this time to serving deep.

The simple specification  $\sigma_{short} + \sigma_{deep}$  is yet most natural and indicative of a mixed strategy, that of occasionally serving short and serving deep but in no fixed pattern.

### 3.5 Logic

Note that we have only presented a syntax for strategy composition and not a logic for reasoning about games and strategies. Our idea is that such strategy specifications can be embedded into modal and temporal logics, and in our contention, without altering their flavour. Consider a modal logic with the modal operator  $\langle i, \sigma \rangle \alpha$ , interpreted at  $s$  to mean that player  $i$  has a strategy  $\sigma$  from that point on to ensure the outcome  $\alpha$  [6]. Here  $\sigma$  need not be a specific functional strategy and can be a compositional specification. A presentation on these lines is carried out in [39].

Similarly a dynamic game logic, in which we have the operator  $\langle i, g, \sigma \rangle \alpha$  which asserts that in game  $g$  player  $i$  has a strategy  $\sigma$  to ensure  $\alpha$  [18]. Again, these can be compositional strategies, and this is discussed in [41].

Consider a temporal logic for games such as ATL with explicit strategies. However, since we are speaking of long term strategic abilities of players and coalitions, we are in the context of ATL\* [14]. We can again consider a modality such as  $\langle\langle C, f \rangle\rangle \alpha$ , where  $f$  specifies a strategy  $\sigma_i$  for each player  $i \in C$ . The intended meaning is that the coalition  $C$ , has a collective strategy given by  $f$  to ensure  $\alpha$ . While such explicit strategies have been studied in alternating temporal logics, the use of structured strategies constraining the paths is the departure advocated here. This can be seen in the spirit of extensions of temporal logic currently used in the industry such as PSL [20] in which the until operator is indexed:  $\alpha \mathcal{U}_\pi \beta$ , where  $\pi$  is a regular expression, and asserts the existence of an instant reachable by  $\pi$  at which  $\beta$  holds, and until then  $\alpha$  holds.

However, such embeddings of structured strategies in modalities would not in themselves lead to reasoning as envisaged by us. For instance, in the case of game logic, strategy composition critically depends on game composition and reasoning about them separately glosses over this. Similarly, in the context of alternating temporal logic, in  $\langle\langle C, f \rangle\rangle\alpha$ , more than the function  $f$ , we are interested in the interdependence between  $f(i)$  and  $f(j)$ , where both  $i$  and  $j$  are in  $C$ . Such analysis opens up interesting avenues for exploration.

## 4 Transducer Lemma

In this section we prove a result that shows the correspondence between the strategy specifications introduced in the previous section with finite state transducers. This helps us to do algorithmic analysis of the games.

**Lemma 1.** *Given a strategy specification  $\phi \in \Phi_i$ , we can construct a finite state transducer  $\mathcal{A}_\phi$  such that for all histories  $s$ , for all  $\nu[s] \in Pstrat_i(s)$  and for all strategy trees  $T_G^{\mu[s]} \in \mathcal{T}_G^{\nu[s]}$ , we have  $T_G^{\mu[s]} \in \llbracket \phi, s \rrbracket_{T_G}$  iff  $T_G^{\mu[s]} \in Lang(\mathcal{A}_\phi, s)$ .*

*Proof Idea.* The proof proceeds in two steps. In the first step we construct the transducer  $\mathcal{A}_\phi$  and in the second step we show that for all  $s$ , for all  $\nu[s] \in Pstrat_i(s)$  and for all strategy trees  $T_G^{\mu[s]} \in \mathcal{T}_G^{\nu[s]}$ ,  $T_G^{\mu[s]} \in \llbracket \phi, s \rrbracket_{T_G}$  iff  $T_G^{\mu[s]} \in Lang(\mathcal{A}_\phi, s)$ .  $\mathcal{A}_\phi$  is constructed inductively.  $\mathcal{A}_{\phi_a}$  is just a one state transducer that outputs  $a$  at every turn of player  $i$ .  $\mathcal{A}_{\phi_1 \cup \phi_2}$  is constructed as the union of  $\mathcal{A}_{\phi_1}$  and  $\mathcal{A}_{\phi_2}$ .  $\mathcal{A}_{\phi_1 \cup \phi_2}$  nondeterministically chooses either  $\mathcal{A}_{\phi_1}$  or  $\mathcal{A}_{\phi_2}$  right at the beginning and then simulates it, mirroring its output.  $\mathcal{A}_{\phi \frown \phi_2}$  and  $\mathcal{A}_{\phi_1 + \phi_2}$  are constructed as products of  $\mathcal{A}_{\phi_1}$  and  $\mathcal{A}_{\phi_2}$ . Both  $\mathcal{A}_{\phi_1}$  and  $\mathcal{A}_{\phi_2}$  are simulated in parallel. In the former case,  $\mathcal{A}_{\phi \frown \phi_2}$  switches from mirroring the output of  $\mathcal{A}_{\phi_1}$  to that of  $\mathcal{A}_{\phi_2}$  nondeterministically at some point. Whereas,  $\mathcal{A}_{\phi_1 + \phi_2}$  switches back and forth between mirroring the outputs of  $\mathcal{A}_{\phi_1}$  and  $\mathcal{A}_{\phi_2}$  nondeterministically.  $\mathcal{A}_{\psi? \phi'}$  has to check, at each step, if  $\psi$  holds at that history. If so then it mirrors the output of  $\mathcal{A}_{\phi'}$  and if not it outputs any move. This is achieved by taking the states of  $\mathcal{A}_{\psi? \phi'}$  to be the product of  $\mathcal{A}_{\phi'}$  with “logical states” (atoms of  $\psi$ ) that tell us whether a subformula of  $\psi$  holds at that history.

Step 2 is proved again by an induction on the structure of  $\phi$ . The idea is to show that a strategy tree  $T$  is in the semantics of  $\phi$  iff  $T$  is in the language defined by  $\mathcal{A}_\phi$ . The base case for  $\phi_a$  follows from the definition. For  $\phi_1 \cup \phi_2$  we show that  $T$  is a strategy tree of  $\phi_1 \cup \phi_2$  iff it is in the language defined by either  $\mathcal{A}_{\phi_1}$  or  $\mathcal{A}_{\phi_2}$ . For  $\phi_1 \frown \phi_2$  we know that  $T$  is a strategy tree iff it can be obtained by pruning a tree  $T_1$  at some finite depth and attaching a trees  $T_m$  to every leaf of the resulting tree where  $T_1$  is a strategy tree of  $\phi_1$  and the  $T_m$ s are strategy trees of  $T_2$ . By the induction hypothesis, this holds iff  $T_1$  is in the language defined by  $\mathcal{A}_{\phi_1}$  and the  $T_m$ s are in the language defined by  $\mathcal{A}_{\phi_2}$ . Finally, for  $\psi? \phi'$ , we know that a tree  $T$  is a strategy tree of  $\psi? \phi'$  iff on every branch of  $T$  if for a finite path,  $\psi$  holds then the move of player  $i$  corresponds to the strategy  $\phi'$ . The check for  $\psi$  is done using the ‘atom graph’ of  $\psi$  and the



move corresponding to  $\phi'$  is verified using the transducer  $\mathcal{A}_{\phi'}$  which exists by the induction hypothesis.  $\square$

We now present the proof in detail.

*Proof. Step 1:* The construction of the transducer  $\mathcal{A}_{\phi}$  is done inductively on the structure of  $\phi$ . Fix the input and output alphabets to be  $\Sigma$ .

$\phi \equiv \phi_a$ : The transducer consists of a single state and outputs the action  $a$  at every turn for player  $i$ . Formally,  $\mathcal{A}_{\phi} = (\{q_0\}, \delta, o, \{q_0\})$  where

- $\delta = \{(q, b, q) \mid b \in \Sigma\}$ .
- $o(q, w) = a$  for all  $w$  such that  $\lambda(w) = i$ .

$\phi \equiv \phi_1 \cup \phi_2$ : The transducer  $\mathcal{A}_{\phi_1 \cup \phi_2}$  should nondeterministically choose either  $\mathcal{A}_{\phi_1}$  or  $\mathcal{A}_{\phi_2}$  right at the beginning and then simulate it, mirroring its output. By the induction hypothesis we have transducers  $\mathcal{A}_{\phi_1} = (Q_1, \delta_1, o_1, I_1)$  and  $\mathcal{A}_{\phi_2} = (Q_2, \delta_2, o_2, I_2)$ . We define  $\mathcal{A}_{\phi_1 \cup \phi_2} = (Q, \delta, o, I)$  where

- $Q = Q_1 \cup Q_2$ ,
- $\delta = \delta_1 \cup \delta_2$ ,
- $I = I_1 \cup I_2$ , and
- $o = o_1 \cup o_2$ .

$\phi \equiv \phi_1 \hat{\ } \phi_2$ : The state space of  $\mathcal{A}_{\phi_1 \hat{\ } \phi_2}$  is the product space of the states of the transducers  $\mathcal{A}_{\phi_1}$  and  $\mathcal{A}_{\phi_2}$ .  $\mathcal{A}_{\phi_1 \hat{\ } \phi_2}$  simulates both these transducers and switches from mirroring the output of  $\mathcal{A}_{\phi_1}$  to that of  $\mathcal{A}_{\phi_2}$  nondeterministically at some point. By the induction hypothesis, we have transducers  $\mathcal{A}_{\phi_1} = (Q_1, \delta_1, o_1, I_1^0)$  and  $\mathcal{A}_{\phi_2} = (Q_2, \delta_2, o_2, I_2^0)$ . We define  $\mathcal{A}_{\phi_1 \hat{\ } \phi_2} = (Q, \delta, o, I^0)$  where

- $Q = Q_1 \times Q_2 \times \{1, 2\}$ ,
- $\delta = \{(q_1, q_2, 1) \xrightarrow{a} (q'_1, q'_2, 1), (q_1, q_2, 2) \xrightarrow{a} (q'_1, q'_2, 2),$   
 $(q_1, q_2, 1) \xrightarrow{a} (q'_1, q'_2, 2) \mid q_1 \xrightarrow{a_1} q'_1, q_2 \xrightarrow{a_2} q'_2\}$ ,
- $I^0 = I_1^0 \times I_2^0 \times \{1\}$  and
- $o : Q \times W \rightarrow \Sigma$  such that  $o((q_1, q_2, 1), w) = o_1(q_1, w)$ ,  $o((q_1, q_2, 2), w) = o_2(q_2, w)$ .

$\phi \equiv \phi_1 + \phi_2$ : The construction of  $\mathcal{A}_{(\phi_1 + \phi_2)}$  is similar to that of  $\mathcal{A}_{\phi_1 \hat{\ } \phi_2}$ . It simulates both  $\mathcal{A}_{\phi_1}$  and  $\mathcal{A}_{\phi_2}$  and keeps switching nondeterministically between the outputs of both. By the induction hypothesis we have transducers  $\mathcal{A}_{\phi_1} = (Q_1, \delta_1, o_1, I_1^0)$  and  $\mathcal{A}_{\phi_2} = (Q_2, \delta_2, o_2, I_2^0)$ . We define  $\mathcal{A}_{(\phi_1 + \phi_2)} = (Q, \delta, o, I^0)$  where

- $Q = Q_1 \times Q_2 \times \{1, 2\}$ ,
- $\delta = \{(q_1, q_2, 1) \xrightarrow{a} (q'_1, q'_2, 1), (q_1, q_2, 2) \xrightarrow{a} (q'_1, q'_2, 2),$   
 $(q_1, q_2, 1) \xrightarrow{a} (q'_1, q'_2, 2), (q_1, q_2, 2) \xrightarrow{a} (q'_1, q'_2, 1) \mid$   
 $q_1 \xrightarrow{a_1} q'_1, q_2 \xrightarrow{a_2} q'_2\}$ ,
- $I^0 = I_1^0 \times I_2^0 \times \{1, 2\}$  and

–  $o : Q \times W \rightarrow \Sigma$  such that  $o((q_1, q_2, 1), w) = o_1(q_1, w)$ ,  $o((q_1, q_2, 2), w) = o_2(q_2, w)$ .

$\phi \equiv \psi? \phi'$ : At each step  $\mathcal{A}_{\psi? \phi'}$  has to check if  $\psi$  holds at that history. We achieve this by taking the states of  $\mathcal{A}_{\psi? \phi'}$  to be the product of  $\mathcal{A}_{\phi'}$  with “logical states” that tell us whether a subformula of  $\psi$  holds at that tree node.

First, some preliminaries. Let  $SF_\psi$  be the least set containing  $\psi$  and closed under subformulas: if  $\alpha \in SF_\psi$  and  $\beta$  is a subformula of  $\alpha$  then  $\beta \in SF_\psi$ . Further we assume that  $SF_\psi$  is closed under negation:  $\beta \in SF_\psi$  iff  $\neg\beta \in SF_\psi$  where we treat  $\neg\neg\alpha$  to be the same as  $\alpha$ ; moreover, if  $\alpha\mathcal{S}\beta \in SF_\psi$ , we have that  $\ominus\alpha\mathcal{S}\beta \in SF_\psi$  as well. Such a set is called the Fischer - Ladner closure and we denote this set by  $CL(\psi)$ .

Now let  $v \subseteq CL(\psi)$ . We call  $v$  an *atom* if it is ‘locally’ consistent and complete: that is, for every  $\psi' \in CL(\psi)$ ,  $\neg\psi' \in v$  iff  $\psi' \notin v$ ; for every  $\psi_1 \vee \psi_2 \in CL(\psi)$ ,  $\psi_1 \vee \psi_2 \in v$  iff  $\psi_1 \in v$  or  $\psi_2 \in v$ ; for every  $\psi_1\mathcal{S}\psi_2 \in CL(\psi)$ , we have: if  $\psi_2 \in v$  then  $\psi_1\mathcal{S}\psi_2 \in v$ ; otherwise  $\{\psi_1, \psi_1\mathcal{S}\psi_2, \ominus\psi_1\mathcal{S}\psi_2\} \subseteq v$ . Let  $V_\psi$  denote the set of  $\psi$ -atoms.  $v \in V_\psi$  is said to be an *initial* atom if it satisfies the conditions:  $\psi_1\mathcal{S}\psi_2 \in v$  iff  $\psi_2 \in v$  and there is no formula of the form  $\langle a \rangle^- \alpha$  in  $v$ .

Define a relation  $\rightarrow_\psi \subseteq (V_\psi \times \Sigma \times V_\psi)$  as follows:  $v \xrightarrow{a} v'$  iff the following conditions hold: for every  $\langle a \rangle^- \alpha$  in  $CL(\psi)$ , if  $\alpha \in v$  then  $\langle a \rangle^- \alpha \in v'$  and for every  $\langle a \rangle^+ \alpha$  in  $CL(\psi)$ , if  $\alpha \in v'$  then  $\langle a \rangle^- \alpha \in v$ . This relation gives us  $G_\psi = (V_\psi, \rightarrow_\psi)$ , the atom graph of  $\psi$ .

We can now define the transducer for this case. Let  $\mathcal{A}_{\phi'} = (Q', \delta', o', I')$  which exists by the induction hypothesis. We define  $\mathcal{A}_{\psi? \phi'} = (Q, \delta, o, I)$  where

- $Q \subseteq Q' \times V_\psi$ ,
- $I \subseteq Q$  such that  $(q, v) \in I$  iff  $q \in I'$  and  $v$  is an initial atom,
- $(q, v) \xrightarrow{a} (q', v')$  iff  $q \xrightarrow{a} q'$  and  $v \xrightarrow{a} v'$ , and
- $o((q, v), w) = o(q, w)$  iff  $\psi \in v$ . Otherwise  $o((q, v), w) = \Sigma$ .

Note that  $o((q, v), w) = \Sigma$  means that the transducer outputs ‘any’ action that is available at  $w$ .

**Step 2:** We now show that for all  $s$ , for all  $\nu[s] \in Pstrat_i(s)$  and for all strategy trees  $T_G^{\mu[s]} \in \mathcal{T}_G^{\nu[s]}$ ,  $T_G^{\mu[s]} \in \llbracket \phi, s \rrbracket_{T_G}$  iff  $T_G^{\mu[s]} \in Lang(\mathcal{A}_\phi, s)$ . The proof is by induction on the structure of  $\phi$ .

$\phi = \phi_a$ : By construction  $\mathcal{A}_\phi$  has just one state  $q$  and all transitions out of  $q$  lead to  $q$  itself. The output function is the constant output function  $o(q, w) = a$  for all  $w$  such that  $\lambda(w) = i$ . Hence by the definition of  $\llbracket \phi_a, s \rrbracket_{T_G}$  we have,  $T_G^{\mu[s]} \in \llbracket \phi_a, s \rrbracket_{T_G}$  if and only if  $T_G^{\mu[s]} \in Lang(\mathcal{A}_{\phi_a}, s)$  for all  $s$ .

$\phi = \phi_1 \cup \phi_2$ : By the semantics,  $T_G^{\mu[s]} \in \llbracket \phi_1 \cup \phi_2, s \rrbracket_{T_G}$  iff  $T_G^{\mu[s]} \in \llbracket \phi_1, s \rrbracket_{T_G} \cup \llbracket \phi_2, s \rrbracket_{T_G}$  iff  $T_G^{\mu[s]} \in \llbracket \phi_1, s \rrbracket_{T_G}$  or  $T_G^{\mu[s]} \in \llbracket \phi_2, s \rrbracket_{T_G}$ . By induction hypothesis,  $T_G^{\mu[s]} \in Lang(\mathcal{A}_{\phi_1}, s)$  or  $T_G^{\mu[s]} \in Lang(\mathcal{A}_{\phi_2}, s)$ . Since  $Lang(\mathcal{A}_{\phi_1 \cup \phi_2}, s) =$

$\text{Lang}(\mathcal{A}_{\phi_1}, s) \cup \text{Lang}(\mathcal{A}_{\phi_2}, s)$  by the construction of  $\mathcal{A}_{\phi_1 \cup \phi_2}$ , we have  $T_G^{\mu[s]} \in \text{Lang}(\mathcal{A}_{\phi_1}, s) \cup \text{Lang}(\mathcal{A}_{\phi_2}, s)$  iff  $T_G^{\mu[s]} \in \text{Lang}(\mathcal{A}_{\phi_1 \cup \phi_2}, s)$ .

$\phi = \phi_1 \hat{\wedge} \phi_2$ : The labelled tree  $T_G^{\mu[s]} \in \llbracket \phi_1 \hat{\wedge} \phi_2, s \rrbracket_{T_G}$  implies by definition of  $\llbracket \phi_1 \hat{\wedge} \phi_2, s \rrbracket_{T_G}$  that there exists a labelled tree  $T_G^{\mu'[s]} \in \llbracket \phi_1, s \rrbracket_{T_G}$  and a finite subtree  $T_G^{\mu'[s]}(k)$  of  $T_G^{\mu[s]}$  such that  $T_G^{\mu'[s]}(k)$  is also a subtree of  $T_G^{\mu'[s]}$  and every branch of  $T_G^{\mu'[s]}(k)$  is of depth at most  $k$ . Also by definition, for every leaf node  $s'$  of  $T_G^{\mu'[s]}(k)$ , there exists  $T_G^{\mu''[s']}$   $\in \llbracket \phi_2, s' \rrbracket_{T_G}$  such that the labelled subtree of  $T_G^{\mu[s]}$  rooted at  $s'$  is the same as  $T_G^{\mu''[s']}$ . By the induction hypothesis,  $T_G^{\mu'[s]} \in \text{Lang}(\mathcal{A}_{\phi_1}, s)$  and  $T_G^{\mu''[s']} \in \text{Lang}(\mathcal{A}_{\phi_2}, s')$ . Thus the run of  $\mathcal{A}_{\phi_1 \hat{\wedge} \phi_2}$  where the transducer makes a switch from the output of  $\mathcal{A}_{\phi_1}$  to that of  $\mathcal{A}_{\phi_2}$  at  $\text{leaves}(T_G^{\mu'[s]}(k))$  is the accepting run for  $T_G^{\mu[s]}$ .

Conversely, suppose  $T_G^{\mu[s]} \in \text{Lang}(\mathcal{A}_{\phi_1 \hat{\wedge} \phi_2})$ . Let  $r$  be an accepting run of  $\mathcal{A}_{\phi_1 \hat{\wedge} \phi_2}$ . Then there exists a  $k \geq 0$  such that  $\mathcal{A}_{\phi_1 \hat{\wedge} \phi_2}$  switches from mirroring the output of  $\mathcal{A}_{\phi_1}$  to that of  $\mathcal{A}_{\phi_2}$  in at most  $k$  steps on all branches of  $r$ . Let  $T_G(k)$  be the finite tree on which  $r$  is a valid run till it makes the switch. We have that there exists a labelled tree  $T_G^{\mu'[s]} \in \text{Lang}(\mathcal{A}_{\phi_1}, s)$  such that  $T_G(k)$  is a subtree of  $T_G^{\mu'[s]}$ . Also, for all leaf nodes  $s' \in \text{leaves}(T_G(k))$ , there exist  $T_G^{\mu''[s']}$  where  $\in \text{Lang}(\mathcal{A}_{\phi_2}, s')$  such that  $\mu[s](s'') = \mu'[s](s'')$  for all  $s''$  with  $|s''| \leq |s'|$  and  $\mu[s](s'') = \mu''[s'](s''')$  where  $ss'' = s's'''$  for all  $s''$  with  $|s''| > |s'|$ . By induction hypothesis,  $T_G^{\mu'[s]} \in \llbracket \phi_1, s \rrbracket_{T_G}$  and  $T_G^{\mu''[s']}$   $\in \llbracket \phi_2, s' \rrbracket_{T_G}$ . Hence the labelled tree  $T_G^{\mu[s]}$  obtained by pasting  $T_G^{\mu''[s']}$  at all leaf nodes  $s'$  of  $T_G^{\mu'[s]}$  belongs to  $\llbracket \phi_1 \hat{\wedge} \phi_2, s \rrbracket_{T_G}$  by the definition of  $\llbracket \phi_1 \hat{\wedge} \phi_2, s \rrbracket_{T_G}$ .

$\phi = (\phi_1 + \phi_2)$ : Suppose  $T_G^{\mu[s]} = (T, m) \in \llbracket \phi_1 + \phi_2, s \rrbracket_{T_G}$ . By semantics,  $s \in T_G^{\mu[s]}$  and there exists  $(T_1, m_1) \in \llbracket \phi_1, s \rrbracket_{T_G}$  and  $(T_2, m_2) \in \llbracket \phi_2, s \rrbracket_{T_G}$  such that for all  $s' \in T_G^{\mu[s]}$ ,  $m(s') = m_1(s')$  or  $m(s') = m_2(s')$ . By induction hypothesis  $(T_1, m_1) \in \text{Lang}(\mathcal{A}_{\phi_1}, s)$  and  $(T_2, m_2) \in \text{Lang}(\mathcal{A}_{\phi_2}, s)$ . By construction, at every node  $s'$ , the transducer  $\mathcal{A}_{\phi_1 + \phi_2}$  mirrors the output of either  $\mathcal{A}_{\phi_1}$  or  $\mathcal{A}_{\phi_2}$ . Therefore we have that  $T_G^{\mu[s]} \in \text{Lang}(\mathcal{A}_{\phi_1 + \phi_2}, s)$ .

Conversely, suppose  $T_G^{\mu[s]} = (T, m) \in \text{Lang}(\mathcal{A}_{\phi_1 + \phi_2}, s)$ . For all  $s \in T_G^{\mu[s]}$ ,  $\mathcal{A}_{\phi_1 + \phi_2}$  mirrors the the output of either  $\mathcal{A}_{\phi_1}$  or  $\mathcal{A}_{\phi_2}$ . Therefore, by construction of  $\mathcal{A}_{\phi_1 + \phi_2}$ , there exists  $(T_1, m_1) \in \text{Lang}(\mathcal{A}_{\phi_1}, s)$  and  $(T_2, m_2) \in \text{Lang}(\mathcal{A}_{\phi_2}, s)$  such that for all  $s'$  we have  $m(s') = m_1(s')$  or  $m(s') = m_2(s')$ . By induction hypothesis,  $(T_1, m_1) \in \llbracket \phi_1, s \rrbracket_{T_G}$  and  $(T_2, m_2) \in \llbracket \phi_2, s \rrbracket_{T_G}$ . By semantics,  $T_G^{\mu[s]} \in \llbracket \phi_1 + \phi_2, s \rrbracket_{T_G}$ .

$\phi = \psi? \phi'$ : Let  $\mathcal{A}_\phi = (Q, \delta, o, I)$ , let  $r$  be a run of  $\mathcal{A}_\phi$  on  $T_G^{\mu[s]}$ .

*Claim 1.* For all  $s' \in T_G^{\mu[s]}$  and for all  $\alpha \in \text{CL}(\psi)$ ,  $\alpha \in r(s')$  iff  $s' \models \alpha$ . where  $\text{CL}(\psi)$  is the subformula closure of  $\psi$ . Assume Claim 1 and suppose that  $T_G^{\mu[s]} = (T, m) \in \llbracket \phi, s \rrbracket_{T_G}$ . By semantics, there exists  $(T', m') \in$

$\llbracket \phi', s \rrbracket_{T_G}$  such that for all  $s' \in T_G^{\nu[s]}$ , if  $s' \models \psi$  then  $m(s') = m'(s')$ . And if  $s' \not\models \psi$  then  $m(s') \in \Sigma$ . By induction hypothesis,  $T_G^{\mu'[s]} \in \text{Lang}(\mathcal{A}_{\phi'}, s)$  and by Claim 1 we have  $s' \models \psi$  implies  $\psi \in r(s')$ . By construction of  $\mathcal{A}_\phi$ , we then have that  $T_G^{\mu[s]} \in \text{Lang}(\mathcal{A}_\phi, s)$ .

Conversely, assume Claim 1 and suppose that  $T_G^{\mu[s]} = (T, m) \in \text{Lang}(\mathcal{A}_\phi, s)$ . For a node  $s' \in T_G^{\mu[s]}$ , let  $r(s') = (q', v')$ . If  $\psi \in v'$  then  $\mathcal{A}_\phi$  mirrors the output of  $\mathcal{A}_{\phi'}$  and if  $\psi \notin v'$  then the transducer outputs an arbitrary action available in  $\Sigma$ . Therefore, by construction of  $\mathcal{A}_\phi$ , there exists  $(T', m') \in \text{Lang}(\mathcal{A}_{\phi'}, s)$  such that for all  $s'$ , if  $\psi \in v'$  then  $m(s) = m'(s')$ . By induction hypothesis,  $T_G^{\mu'[s]} \in \llbracket \phi', s \rrbracket_{T_G}$  and by Claim 1 if  $\psi \in v'$  then  $s' \models \psi$ . By the semantics we then have that  $T_G^{\mu[s]} \in \llbracket \phi, s \rrbracket_{T_G}$ .

It now remains to prove Claim 1. We do so by a second induction on the structure of  $\alpha$ .

$\alpha = p \in \mathcal{P}_i$ : Follows from definition since in the construction we ensured that  $(q, v) \in Q$  iff  $q \cap \mathcal{P}_i = v \cap \mathcal{P}_i$ .

$\alpha = \neg\beta$ :  $s' \models \neg\beta$  iff  $s' \not\models \beta$  iff  $\beta \notin r(s') = (q, v)$  iff  $\neg\beta \in (q, v)$  (since  $v$  is an atom).

$\alpha = \alpha_1 \vee \alpha_2$ :  $s' \models \alpha$  iff  $s' \models \alpha_1$  or  $s' \models \alpha_2$  iff  $\alpha_1 \in r(s')$  or  $\alpha_2 \in r(s')$  where  $r(s') = (q, v)$  iff  $\alpha_1 \vee \alpha_2 \in r(s')$  (since  $v$  is an atom).

$\alpha = \langle a \rangle^{-}\beta$ :  $s' \models \langle a \rangle^{-}\beta$  iff  $s' = s''aw$  and  $s'' \models \beta$  iff  $\beta \in r(s'') = (q', v')$  iff  $\langle a \rangle^{-}\beta \in r(s) = (q, v)$  since  $v$  is an atom and  $(q', v') \xrightarrow{a} (q, v)$  iff  $q' \xrightarrow{a} q$  and  $v' \xrightarrow{a} v$  by construction.

$\alpha = \langle a \rangle^{+}\beta$ : Similar to the case for  $\alpha = \langle a \rangle^{-}\beta$ .

$\alpha = \alpha_1 \mathcal{S} \alpha_2$ :  $s' \models \alpha_1 \mathcal{S} \alpha_2$  iff there exists  $s_1, s_1 \sqsubset s'$  such that  $s_1 \models \alpha_2$  and for all  $s_2, s_1 \sqsubset s_2 \sqsubseteq s'$  and  $s_2 \models \alpha_1$ . We do an induction on  $|s'| - |s_1|$ . When  $|s'| - |s_1| = 0$ ,  $s' \models \alpha_2$  iff  $\alpha \in r(s')$  by induction hypothesis where  $r(s') = (q, v)$  iff  $\alpha_1 \mathcal{S} \alpha_2 \in r(s')$  (since  $v$  is an atom). When  $|s'| - |s_1| = k + 1$ ,  $s_2 \models \alpha_1 \mathcal{S} \alpha_2$  and either  $s'$  in which case we are done (by definition of initial atom), or  $s' \models \ominus(\alpha_1 \mathcal{S} \alpha_2)$  where  $s' = s_2aw$  iff  $\alpha_1 \mathcal{S} \alpha_2 \in r(s_2)$  iff  $\alpha_1 \mathcal{S} \alpha_2 \in r(s') = (q, v)$  (since  $v$  is an atom).

$j^?\phi$ :  $s' \models j^?\phi$  iff there exists  $T^\phi$  such that  $T^\phi \in \llbracket \phi, \epsilon \rrbracket_{T_G}$  and  $s' \in T^\phi$  iff  $T^\phi \in \text{Lang}(\mathcal{A}_\phi, \epsilon)$  by the main induction hypothesis, iff  $j^?\phi \in r(s') = (q, v)$  since by construction,  $j^?\phi \in v$  and  $q \in \delta'(q_0, \text{last}(s'))$  where  $q_0$  is an initial state of  $\mathcal{A}_\phi$ .

## 5 Applications

The Transducer Lemma provides us with a tool to talk about eventual outcomes in games and eventual behaviour of players given that they play according to strategy specifications given in our syntax. We formalise this below.

Given a game arena, and strategy specifications for players, we are interested in studying long-range outcomes. The strategy specifications are akin to player

types; based on observations during play, they constrain player actions at game positions. We would then like to know whether a given outcome is achieved. Since these are infinite games, the natural such notion is that of **stable** outcomes.

## 5.1 Stable Outcome

Let  $\mathcal{G} = (W, \rightarrow, w_0, \lambda)$  be a game arena and let there be  $n$  players. Suppose the strategies of the players are given as  $\phi_1, \phi_2, \dots, \phi_n$  respectively where each  $\phi_i$  is from the syntax  $\Phi_i$  as described in Sect. 3. Suppose  $\alpha$  is a property from the following syntax:

$$\alpha := p \in \mathcal{P} \mid \neg\alpha \mid \alpha_1 \vee \alpha_2 \mid \langle a \rangle^+ \alpha$$

where  $\mathcal{P}$  is a set of ‘global’ propositions and  $val : W \rightarrow 2^{\mathcal{P}}$  gives their valuation at the nodes of the arena. We want to check if it is the case that eventually the property  $\alpha$  always holds given that the players play according to their strategy specifications. In other words, we wish to check if  $\alpha$  becomes ‘stable’ in the game, where we define stability as: Given a subarena  $\mathcal{G}'$  of  $\mathcal{G}$ ,  $\alpha$  is said to be **stable** in  $\mathcal{G}'$  if  $s \models \alpha$  for every  $s \in T_{\mathcal{G}'}$ .

We do this as follows. We first construct transducers  $\mathcal{A}_{\phi_i}$ , as described in Sect. 4, for the strategy specification of each player  $i$ . We then take the ‘restriction’ of the arena  $\mathcal{G}$  with respect to each of these transducers which results in a new arena  $\mathcal{G}'$ . The restriction of  $\mathcal{G}$  with respect to a transducer  $\mathcal{A}_{\phi_i} = (Q, \delta, o, q_0)$  is denoted as  $\mathcal{G} \upharpoonright \mathcal{A}_{\phi_i}$  and is defined as  $\mathcal{G} \upharpoonright \mathcal{A}_{\phi_i} = (W', \rightarrow', w'_0, \lambda')$  where

- $W' \subseteq W \times Q$  such that  $w' = (w, q) \in W'$  iff  $p \in val(w) \Leftrightarrow p \in val_i(q)$  for all  $p \in \mathcal{P} \cap \mathcal{P}_i$ .
- For every  $w'_1, w'_2 \in W'$  where  $w'_1 = (w_1, q_1)$  and  $w'_2 = (w_2, q_2)$   $w'_1 \xrightarrow{a'} w'_2$  iff  $w_1 \xrightarrow{a} w_2$  and  $q_1 \xrightarrow{a} q_2$ .
- $w'_0 = (w_0, q_0)$ .
- $\lambda'(w', q') = \lambda(w)$ .

Thus the final restricted arena is  $\mathcal{G}' = (((\mathcal{G} \upharpoonright \mathcal{A}_{\phi_1}) \upharpoonright \dots) \upharpoonright \mathcal{A}_{\phi_n})$ . Note that the Transducer Lemma ensures that in the restricted arena  $\mathcal{G}'$ , for every node  $w' \in \mathcal{G}'$  the outgoing edges are exactly the moves prescribed by the strategy specification  $\phi_i$  for player  $i$  where  $\lambda'(w') = i$ . Finally, to check whether  $\alpha$  becomes eventually stable, we check the stability of  $\alpha$  in  $\mathcal{G}'$ . This can be done by a simple marking procedure on the nodes of  $\mathcal{G}'$  with the subformulae of  $\alpha$ . We thus have proved the following theorem:

**Theorem 1.** *Given a game arena  $\mathcal{G} = (W, \rightarrow, w_0, \lambda)$  with  $n$  players and given that the players play according to strategy specifications  $\phi_1, \phi_2, \dots, \phi_n$  respectively where each  $\phi_i \in \Phi_i$ , we can effectively decide if a property  $\alpha$  becomes eventually stable in the game.*

## 5.2 Solution Concepts

Apart from achieving logically specified outcomes as above, there are other applications of the transducer construction. In the study of (offline) strategies in game theory, the standard questions relate to best response, equilibria etc. In the case of structured strategies, similar questions can be answered using automata.

However, since a strategy specification denotes a set of strategies satisfying certain properties, notions like strategy comparison and best response with respect to strategy specifications need to be redefined. When we have functional strategies  $\mu_1$  and  $\mu_2$  for a player  $i$ , we can consider  $\mu_1$  to dominate  $\mu_2$  if for every opponent strategy profile  $\mu^{-i}$ , the play  $(\mu_1, \mu^{-i})$  is preferred by  $i$  over the play  $(\mu_2, \mu^{-i})$ . But when does a set of strategies dominate another set? This is not clear.

One notion we can define says that  $\sigma$  is better than  $\sigma'$  if for any outcome  $\alpha$  if there is a strategy conforming to the specification  $\sigma'$  which ensures  $\alpha$ , then there also exists a strategy conforming to  $\sigma$  which ensures  $\alpha$  as well.

But then it's equally reasonable to define comparison somewhat differently. According to this,  $\sigma$  is better than  $\sigma'$  if for any outcome  $\alpha$  whenever there is a strategy conforming to  $\sigma$  which cannot guarantee  $\alpha$ , there also exists a strategy conforming to  $\sigma'$  which cannot guarantee  $\alpha$  either. This can be thought of as a soundness condition. A risk averse player might prefer this way of comparison.

We do not want to fix any comparison notion here, but merely point out that there are different and interesting notions of comparison. But once we fix such a notion, we can consider algorithmic questions relating to players' best response.

- Does player  $i$  have a strategy conforming to  $\sigma$  to ensure outcome  $\alpha$  as long as the other players play according to the specifications  $\tau^{-i}$ ?
- Do all strategies for player  $i$  conforming to  $\sigma$  constitute a best response against  $\tau^{-i}$  for  $\alpha$ ?
- Given specifications  $\tau^{-i}$  for the other players, synthesize a best response for player  $i$  as an automaton.

Now, all these questions can be answered algorithmically using the transducer construction for strategy specifications, for appropriately defined outcomes. [40] presents some results in that direction. In general, we can study questions like whether a given tuple of strategy specifications constitutes a Nash equilibrium (with respect to fixed strategy comparison notions). In some sense, the main idea is that all the difficulties related to online strategy switching are modularised into the transducer construction.

## 5.3 Game Logics

As indicated earlier, we expect strategy specifications to be embedded in logics with assertions of the form “playing a strategy conforming to  $\sigma$  ensures outcome  $\alpha$  for player  $i$ ”. The decision procedures for model checking such logics are naturally constructed by tree automata that run over game trees, with transducers

running in parallel for strategy specifications in subformulas. Thus the construction offers a generic way of algorithmically constructing strategies on the fly for logically specified outcomes. [42] presents some results in that direction.

A more general remark would be that in any logics for reasoning about extensive form games, if the embedded specifications admit a transducer construction such as the one outlined here, the model checking algorithm for the logic constitutes strategy construction in the sense envisaged in our motivation. [41] constitutes one such example, but this style of automata based reasoning about strategies needs to be expanded much further, especially for Alternating Temporal Logics. An interesting challenge would be to introduce *belief structure* into automata so that the considerable body of logical work on epistemic reasoning about strategies ([8]) can be addressed. As long as epistemic assertions relate only to the past, automaton construction seems extendable. When beliefs about future need to be considered, the issues become indeed complex.

## 6 Discussion

We have presented a syntax of strategy specifications involving strategy switching and showed their realization as automata. This be seen as making a *prima facie* case for considering online strategizing and the use of automata for studying memory structure in such strategies.

### Related Work

As remarked in the introduction, the motivation for considering online strategizing at all comes from the felt need for a ‘theory of play’ on the lines of [6, 8, 9]. The motivation for considering games as infinite trees for such a study is akin to the discussion on games with memory in [43]. The model itself and the representation of strategies as finite state transducers, is based on the study of regular infinite games, as in [22].

An important line of work in game theory about strategy switching is that of Lipman and Wang in [28, 29]. The framework is that of finite and infinite repeated games, and players switch strategies during periods of the repeated game. They look at how equilibria change when players incur a cost on every strategy switch they make. The critical difference for our work is that switching is not based on previous game outcomes but observations by players in the course of play.

In general, dynamic learning has been extensively studied in game theory: for instance, Young ([47, 48]) considers a model in which each player chooses an optimal strategy based on a sample of information about what other players have done in the past. Similar analyses have been carried out in the context of cooperative game theory as well: here players decide dynamically which coalition to join. One asks how coalition structures change over time, and which coalition players will eventually arrive at ([2]). Evolutionary game theory ([46]) studies how players observe payoffs of other players in their neighbourhood and

accordingly change strategies to maximise fitness. However, these studies again use offline strategies and deliberation between periods of repeated games rather than online strategizing in games which is the focus here.

We have used a syntax for strategy specifications that is logical, using formulas of a simple modal logic for observations of players but otherwise using operators that can be seen as algebraic, their semantics given by operations on trees. Note the absence of negation in strategy specifications; thus, they are closer to programs in propositional dynamic logic (PDL) rather than logical assertions in themselves. Indeed, like in the case of PDL, where programs are embedded in an assertion language that speaks of postconditions, strategy specifications would be embedded in a logical language that speaks of preferences and outcomes, so that one can speak of  $\sigma$  ensuring the “best” outcome  $\alpha$  for a player. A programme of this sort is carried out for a more restricted class of strategy specifications in [42] and a complete axiomatization is given. In general, we do not see the language of specifications as a logical means rather than as an end, and our purpose here has been to show their realization as automata.

In this context, we note that there exist a variety of logics in which strategy specifications may be embedded in the sense above. In general, any logic that speaks of a player (or a set of players) having a strategy to ensure an outcome would be appropriate. Notable among these is the work on alternating temporal logic (ATL) [1], a logic on trees with just these kind of assertions. Various extensions of ATL ([26, 27]) have been proposed to incorporate knowledge of players and strategies explicitly into the logic. In particular, the logic ATL\* ([14]) is powerful enough to include changing strategy contexts and constructive concepts of strategy as well. However, in our presentation, we focus not on reasoning about games as in ATL but only on strategy switching and composition.

In [4, 5, 8] van Benthem uses dynamic logic to describe games as well as strategies. In fact, PDL can be seen as a language for strategy specifications as well. However, it is unclear whether the switching operator, being a kind of ‘interval’ operator, or the mixing operator  $+$ , are definable in PDL at all. Note that these operators are much closer to those of *process logic* ([24]). A form of  $\wedge$  operator, called ‘chop’ is extensively used in interval logics ([32]) but these are over linear orders and similar operations on trees seem to be difficult to define.

Ghosh [18] presents a complete axiomatisation of a logic describing both games and strategies in a dynamic logic framework where assertions are made about atomic strategies. Our earlier work [39] studies a logic in which not only are games structured, but so also are strategies. [19] enriches strategy specifications with an interleaving based parallel composition operator. All these logics are closely related and automata constructions can be given for the strategy specifications in these logics. Typically, logics like PDL correspond to automata on sequences whereas strategy logics involve automata that accept trees or act as tree to word transducers, and hence the constructions tend to be more complex. Strategy switching makes essential use of transductions which has been the emphasis here.



## Further work

We have advocated a programme of working with ‘online constructible’ strategy spaces in large extensive form games, but what we have provided is a mere illustration of possibilities that such constructions can be carried out within the framework of simple modal logics and automata. However, the programme requires a precise delineation of such a strategy space within that of the classical offline strategy space, or even within the computable strategy space. The work presented here does not provide any basis for such a characterization.

The notion of strategies as constraining relations rather than functions is in itself worthy of deeper study, and characterizing the class of relations that can be realised as programs seems to be an interesting question. Partial strategies are akin to heuristics that can apply to a variety of game situations. The study here suggests that we can consider access to a library of such partial strategies and online composition, much like software.

While we have focused on one particular aspect of online strategizing, namely that of composition and switching, a number of aspects need incorporation for building a theory of play. Critically, the belief structure of players about other players’ strategies and their mutual beliefs and expectations critically affects strategizing ([8]). While strategy specifications presented here can be viewed in themselves as player types, the latter crucially incorporate beliefs as well, and this needs further exploration.

The expressiveness of the specification language for strategies that we have presented is unclear. Since the automata considered recognize regular tree languages, and hence (presumably) are equivalent to some monadic second order logic on trees, one expects that there must be strategies implementable as automata but not definable in this limited specification language. However, note that switching involves taking an ‘initial fragment’ of one subtree and a ‘final fragment’ of another subtree and gluing them together. The logical status of such tree operations is unclear. Is this first-order definable, and if yes, with what vocabulary? Note that characterizing first order definability on (unranked) trees is difficult in general. What is an expressively complete set of strategy composition operators with respect to finite state transducers? This seems to be an interesting question to answer.

Another important dimension is the algebraic structure of strategy composition ([21]). What is a natural notion of equivalence on strategies, and reductions between them? Switching imposes an interval-like substructure on trees, and the interaction of such structure with other operators seems worthy of further study.

**Acknowledgments.** The authors would like to thank Dietmar Berwanger and Sujata Ghosh for many interesting discussions on this theme and the editors of the volume for their patience and encouragement.

## References

1. Alur, R., Henzinger, T.A., Kupferman, O.: Alternating-time temporal logic. *J. ACM* **49**(5), 672–713 (2002)
2. Arnold, T., Schwalbe, U.: Dynamic coalition formation and the core. *J. Econ. Behav. Organ.* **49**, 363–380 (2002)
3. Aumann, R.J., Dreze, J.H.: When all is said and done, how should you play and what should you expect? (2005). [http://www.ma.huji.ac.il/raumann/pdf/dp\\_387.pdf](http://www.ma.huji.ac.il/raumann/pdf/dp_387.pdf)
4. van Benthem, J.: Games in dynamic epistemic logic. *Bull. Econ. Res.* **53**(4), 219–248 (2001)
5. van Benthem, J.: Extensive games as process models. *J. Logic Lang. Inform.* **11**, 289–313 (2002)
6. van Benthem, J.: In praise of strategies. In: van Eijck, J., Verbrugge, R. (eds.) *Foundations of Social Software, Studies in Logic*, pp. 283–317. College Publications (2007)
7. van Benthem, J.: Exploring a theory of play. In: Apt, K.R. (ed.) *Proceedings of the 13th Conference on Theoretical Aspects of Rationality and Knowledge*, pp. 12–16. ACM (2011)
8. van Benthem, J.: *Logic in Games*. MIT Press, Cambridge (2014)
9. van Benthem, J., Pacuit, E., Roy, O.: Toward a theory of play: a logical perspective on games and interaction. *Games* **2**, 52–86 (2011)
10. Bonanno, G.: Modal logic and game theory: two alternative approaches. *Risk Decis. Policy* **7**, 309–324 (2002)
11. Borel, E.: La théorie du jeu et les équations intégrales à noyau symétrique gauche. *C. R. Acad. Sci.* **173**, 1304–1308 (1921)
12. Büchi, J.R.: On a decision method in restricted second-order arithmetic. In: *Proceedings of the 1960 International Congress for Logic, Methodology and Philosophy of Science*, pp. 1–11. Stanford University Press (1962)
13. Büchi, J.R., Landweber, L.H.: Solving sequential conditions by finite-state strategies. *Trans. Americal Math. Soc.* **138**, 295–311 (1969)
14. Bulling, N., Goranko, V., Jamroga, W.: Logics for reasoning about strategic abilities in multi-player games. In: van Benthem, J., Ghosh, S., Verbrugge, R. (eds.) *Models of Strategic Reasoning*. LNCS, vol. 8972, pp. 93–136. Springer, Heidelberg (2015)
15. Majumdar, R., Jurdziński, M., Chatterjee, K.: On nash equilibria in stochastic games. In: Tarlecki, A., Marcinkowski, J. (eds.) *CSL 2004*. LNCS, vol. 3210, pp. 26–40. Springer, Heidelberg (2004)
16. Emerson, A., Jutla, C.: Tree automata, mu-calculus and determinacy. In: *Proceedings of the 32nd IEEE Symposium on Foundations of Computer Science*, pp. 368–377. IEEE Computer Society Press (1991)
17. Gale, D., Stewart, F.: Infinite games with perfect information. *Ann. Math. Stud.* **28**, 245–266 (1953)
18. Ghosh, S.: Strategies made explicit in dynamic game logic. In: *Proceedings of the Workshop on Logic and Intelligent Interaction, ESSLLI 2008*, pp. 74–81 (2008)
19. Ghosh, S., Simon, S., Ramanujam, R.: Playing extensive form games in parallel. In: Dix, J., Governatori, G., Leite, J., Jamroga, W. (eds.) *CLIMA XI*. LNCS, vol. 6245, pp. 153–170. Springer, Heidelberg (2010)

20. Glazberg, Z., Moulin, M., Orni, A., Ruah, S., Zarpas, E.: PSL: Beyond hardware verification. In: Ramesh, S., Sampath, P. (eds.) *Next Generation Design and Verification Methodologies for Distributed Embedded Control Systems*, pp. 245–260. Springer, Netherlands (2007)
21. Goranko, V.: The basic algebra of game equivalences. *Stud. Logica* **75**(2), 221–238 (2003)
22. Farwer, B.:  $\omega$ -automata. In: Grädel, E., Thomas, W., Wilke, T. (eds.) *Automata, Logics, and Infinite Games*. LNCS, vol. 2500, pp. 3–21. Springer, Heidelberg (2002)
23. Grädel, E., Ummels, M.: Solution concepts and algorithms for infinite multiplayer games. In: *New Perspectives on Games and Interaction*. Texts in Logic and Games, vol. 4, pp. 151–178. Amsterdam University Press (2008)
24. Harel, D., Kozen, D., Parikh, R.: Process logic: expressiveness, decidability, completeness. *J. Comput. Syst. Sci.* **25**(2), 144–170 (1982)
25. van den Herik, J., Schaeffer, J.: Games, computers, and artificial intelligence. *Artif. Intell.* **134**, 1–7 (2001)
26. van der Hoek, W., Jamroga, W., Wooldridge, M.: A logic for strategic reasoning. In: *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pp. 157–164 (2005)
27. van der Hoek, W., Wooldridge, M.: Cooperation, knowledge, and time: alternating-time temporal epistemic logic and its applications. *Stud. Logica* **75**(1), 125–157 (2003)
28. Lipman, B.L., Wang, R.: Switching costs in frequently repeated games. Working Papers 955, Queen’s University, Department of Economics, September 1997
29. Lipman, B.L., Wang, R.: Switching costs in infinitely repeated games. *Games Econ. Behav.* **66**(1), 292–314 (2009)
30. Martin, D.A.: Borel determinacy. *Ann. Math.* **102**, 363–371 (1975)
31. Mostowski, A.W.: Games with forbidden positions. Technical Report 78, Instytut Matematyki, Uniwersytet Gdański, Poland (1991)
32. Moszkowski, B.C., Manna, Z.: Reasoning in interval temporal logic. In: Clarke, E.M., Kozen, D. (eds.) *Logic of Programs*. Lecture Notes in Computer Science, vol. 164, pp. 371–382. Springer, Heidelberg (1983)
33. Muller, D.E.: Infinite sequences and finite machines. In: *Proceedings of the 4th IEEE Symposium on Switching Circuit Theory and Logical Design*, pp. 3–16 (1963)
34. Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V.V.: *Algorithmic Game Theory*. Cambridge University Press, New York (2007)
35. Osborne, M.J., Rubinstein, A.: *A Course in Game Theory*. MIT Press, Cambridge (1994)
36. Paul, S., Ramanujam, R., Simon, S.: Stability under strategy switching. In: Löwe, B., Merkle, W., Ambos-Spies, K. (eds.) *CiE 2009*. LNCS, vol. 5635, pp. 389–398. Springer, Heidelberg (2009)
37. Paul, S., Simon, S.: Nash equilibrium in generalised Muller games. In: *Proceedings of the Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS, Leibniz International Proceedings in Informatics (LIPIcs)*, vol. 4, pp. 335–346. Schloss Dagstuhl-Leibniz-Zentrum für Informatik (2009)
38. Perea, A.: *Epistemic Game Theory: Reasoning and Choice*. Cambridge University Press, Cambridge (2012)
39. Ramanujam, R., Simon, S.: Axioms for composite strategies. In: *Proceedings of the 7th Conference on Logic and the Foundations of Game and Decision Theory (LOFT 2006)* pp. 189–198 (2006)

40. Ramanujam, R., Simon, S.: Structured strategies in games on graphs. In: Flum, J., Grädel, E., Wilke, T. (eds.) *Logic and Automata: History and Perspectives*. Texts in Logic and Games, vol. 2, pp. 567–587. Amsterdam University Press (2007)
41. Ramanujam, R., Simon, S.: Dynamic logic on games with structured strategies. In: *Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR-2008)*, pp. 49–58. AAAI Press (2008)
42. Ramanujam, R., Simon, S.: A logical structure for strategies. In: *Logic and the Foundations of Game and Decision Theory (LOFT 7)*, Texts in Logic and Games, vol. 3, pp. 183–208. Amsterdam University Press (2008)
43. Rubinstein, A.: Comments on the interpretation of game theory. *Econometrica* **59**, 909–924 (1991)
44. Selten, R.: Spieltheoretische Behandlung eines Oligopolmodells mit Nachfragerträgeit. *Zeitschrift für die Gesamte Staatswissenschaft* **121**, 301–324 (1965)
45. von Neumann, J.: Zur Theorie der Gesellschaftsspiele. *Math. Ann.* **100**, 295–328 (1928)
46. Weibull, J.W.: *Evolutionary Game Theory*. MIT Press, Cambridge, MA (1997)
47. Young, H.P.: The evolution of conventions. *Econometrica* **61**, 57–84 (1993). Blackwell Publishing
48. Young, H.P.: The diffusion of innovations in social networks. Economics Working Paper Archive 437, The Johns Hopkins University, Department of Economics (2000)