Playing extensive form games in parallel

Sujata Ghosh[1] *, R. Ramanujam[2] and Sunil Simon[3] **

[1] Department of Artificial Intelligence
University of Groningen.
sujata@ai.rug.nl
[2] The Institute of Mathematical Sciences
C.I.T. Campus, Chennai 600 113, India.
jam@imsc.res.in
[3] Institute for Logic, Language and Computation
University of Amsterdam.
s.e.simon@uva.nl

Abstract. Consider a player playing against different opponents in two extensive form games simultaneously. Can she then have a strategy in one game using information from the other? The famous example of playing chess against two grandmasters simultaneously illustrates such reasoning. We consider a simple dynamic logic of extensive form games with sequential and parallel composition in which such situations can be expressed. We present a complete axiomatization and show that the satisfiability problem for the logic is decidable.

1 Motivation

How can any one of us [4] expect to win a game of chess against a Grandmaster (GM)? The strategy is simple: play simultaneously against two Grandmasters! If we play black against GM 1 playing white, and in the parallel game play white against GM 2 playing black, we can do this simply. Watch what GM 1 plays, play that move in the second game, get GM 2's response, play that *same* move as our response in game 1, and repeat this process. If one of the two GMs wins, we are assured of a win in the other game. In the worst case, both games will end in a draw.

Note that the strategy construction in this example critically depends on several features:

- Both games need to be played in *lock-step synchrony*; if they are slightly out of step with each other, or are sequentialized in some way, the strategy is not applicable. So concurrency is critically exploited.

* Supported by the Netherlands Organisation of Scientific Research grant 600.065.120.08N201

** Supported by the European Commission grant EMECW15

[4] By "us" we mean poor mortals who know how to play the game but lack expertise.

– The strategy cannot be constructed *a priori*, as we do not know what moves would be played by either of the GMs. Such reasoning is intrinsically different from the discussion of the existence of winning strategies in determined games. In particular, strategic reasoning as in normal form games is not applicable.
– The common player in the two games acts as a conduit for transfer of information from one game to the other; thus *game composition* is essential for such reasoning. The example illustrates that playing several instances of the same game may mean something very different from repeated games.
– The common player can be a resource bounded agent who cannot analyse the entire game structure and compute the winning strategy (even if it exists). The player thus mimics the moves of an "expert" in order to win one of the constituent games.

In general, when extensive form games are played in parallel, with one player participating in several games simultaneously, such an information transfer from one game to the other is possible. In general, since strategies are structured in extensive form games, they can make use of such information in a non-trivial manner.

In the context of agent-based systems, agents are supposed to play several interactive roles at the same time. Hence when interaction is modelled by games (as in the case of negotiations, auctions, social dilemma games, market games, etc.) such parallel games can assume a great deal of importance. Indeed, a prominent feature of an agent in such a system is the ability to *learn* and transferring strategic moves from one game to the other can be of importance as one form of learning.

Indeed, sequential composition of games can already lead to interesting situations. Consider player A playing a game against B, and after the game is over, playing another instance of the *same* game against player C. Now each of the leaf nodes of the first game carries important historical information about play in the game, and A can strategize differently from each of these nodes in the second game, thus reflecting learning again. Negotiation games carry many such instances of history-based strategizing.

What is needed is an algebra of game composition in which the addition of a parallel operator can be studied in terms of how it interacts with the other operators like choice and sequential composition. This is reminiscent of process calculi, where equivalence of terms in such algebras is studied in depth.

In this paper, we follow the seminal work of Parikh ([12]) on **propositional game logic**. We use dynamic logic for game expressions but extended with parallel composition; since we wish to take into account game structure, we work with extensive form games embedded in Kripke structures rather than with effectivity functions. In this framework, we present a complete axiomatization of the logic and show that the satisfiability problem for the logic is decidable.

The interleaving operator has been looked at in the context of program analysis in terms of dynamic logic [1]. The main technical difficulty addressed in the paper is that parallel composition is not that of sequences (as typically done in process calculi) but that of trees. The main modality of the logic is an assertion of the form $\langle g, i\rangle\alpha$ which asserts, at a state s, that a tree t in the "tree language" associated with g is enabled at s, and that player i has a strategy (subtree) in it to ensure α. Parallel composition is not compositional in the standard logical sense: the semantics of $g_1 \| g_2$ is not given in terms of the semantics of g_1 and g_2 considered as wholes, but by going into their structure. Therefore, defining the enabled-ness of a strategy as above is complicated. Note that the branching structure we consider is quite different from the intersection operator in dynamic logic [8, 6, 11] and is closer to the paradigm of concurrent dynamic logic [14].

For ease of presentation, we first present the logic with only sequential and parallel composition and discuss technicalities before considering iteration, which adds a great deal of complication. Note that the dual operator, which is important in Parikh's game logic is not relevant here, since we wish to consider games between several players played in parallel.

Related work

Games have been extensively studied in temporal and dynamic logics. For concurrent games, this effort was pioneered by work on Alternating time temporal logic (ATL) [3], which considers selective quantification over paths. Various extension of ATL was subsequently proposed, these include ones in which strategies can be named and explicitly referred to in the formulas of the logic [18, 2, 19]. Parikh's work on propositional game logics [12] initiated the study of game structures in terms of algebraic properties. Pauly [13] has built on this to reason about abilities of coalitions of players. Goranko draws parallels between Pauly's coalition logic and ATL [7]. Van Benthem uses dynamic logic to describe games and strategies [16]. Strategic reasoning in terms of a detailed notion of agency has been studied in the *stit* framework [10, 4, 5].

Somewhat closer in spirit is the work of [17] where van Benthem and co-authors develop a logic to reason about simultaneous games in terms of a parallel operator. The reasoning is based on powers of players in terms of the outcome states that can be ensured. Our point of departure is in considering extensive form game trees explicitly and looking at interleavings of moves of players in the tree structure.

2 Preliminaries

2.1 Extensive form games

Let $N = \{1, \ldots, n\}$ denote the set of players, we use i to range over this set. For $i \in N$, we often use the notation $\bar{\imath}$ to denote the set $N \setminus \{i\}$. Let Σ be a finite set of action symbols representing moves of players, we let a, b range over Σ. For a set X and a finite sequence $\rho = x_1 x_2 \ldots x_m \in X^*$, let $last(\rho) = x_m$ denote the last element in this sequence.

Game trees: Let $\mathbb{T} = (S, \Rightarrow, s_0)$ be a tree rooted at s_0 on the set of vertices S and $\Rightarrow : (S \times \Sigma) \rightarrow S$ is a *partial* function specifying the edges of the tree. The tree $\mathbb{T}$ is said to be finite if S is a finite set. For a node $s \in S$, let $\overrightarrow{s} = \{s' \in S \mid s \overset{a}{\Rightarrow} s' \text{ for some } a \in \Sigma\}$, $moves(s) = \{a \in \Sigma \mid \exists s' \in S \text{ with } s \overset{a}{\Rightarrow} s'\}$ and $E_T(s) = \{(s, a, s') \mid s \overset{a}{\Rightarrow} s'\}$. By $E_T(s) \times x$ we denote the set $\{((s,x), a, (s', x)) \mid (s, a, s') \in E_T(s)\}$. The set $x \times E_T(s)$ is defined similarly. A node s is called a leaf node (or terminal node) if $\overrightarrow{s} = \emptyset$. The depth of a tree is the length of the longest path in the tree.

An extensive form game tree is a pair $T = (\mathbb{T}, \widehat{\lambda})$ where $\mathbb{T} = (S, \Rightarrow, s_0)$ is a tree. The set S denotes the set of game positions with s_0 being the initial game position. The edge function $\Rightarrow$ specifies the moves enabled at a game position and the turn function $\widehat{\lambda} : S \rightarrow N$ associates each game position with a player. Technically, we need player labelling only at the non-leaf nodes. However, for the sake of uniform presentation, we do not distinguish between leaf nodes and non-leaf nodes as far as player labelling is concerned. An extensive form game tree $T = (\mathbb{T}, \widehat{\lambda})$ is said to be finite if $\mathbb{T}$ is finite. For $i \in N$, let $S^i = \{s \mid \widehat{\lambda}(s) = i\}$ and let $frontier(T)$ denote the set of all leaf nodes of T. Let $S^L_T = frontier(T)$ and $S^{NL}_T = S \setminus S^L_T$. For a tree $T = (S, \Rightarrow, s_0, \widehat{\lambda})$ we use $head(T)$ denote the depth one tree generated by taking all the outgoing edges of s_0.

A play in the game T starts by placing a token on s_0 and proceeds as follows: at any stage if the token is at a position s and $\widehat{\lambda}(s) = i$ then player i picks an action which is enabled for her at s, and the token is moved to s' where $s \overset{a}{\Rightarrow} s'$. Formally a play in T is simply a path $\rho : s_0 a_1 s_1 \cdots$ in $\mathbb{T}$ such that for all $j > 0$, $s_{j-1} \overset{a_j}{\Rightarrow} s_j$. Let $Plays(T)$ denote the set of all plays in the game tree T.

2.2 Strategies

A strategy for player $i \in N$ is a function μ^i which specifies a move at every game position of the player, i.e. $\mu^i : S^i \rightarrow \Sigma$. A strategy μ^i can also be viewed as a subtree of T where for each player i node, there is a unique outgoing edge and for nodes belonging to players in $\bar{\imath}$, every enabled move is included. Formally we define the strategy tree as follows: For $i \in N$ and a player i strategy $\mu^i : S^i \rightarrow \Sigma$ the strategy tree $T_{\mu^i} = (S_{\mu^i}, \Rightarrow_{\mu^i}, s_0, \widehat{\lambda}_{\mu^i})$ associated with μ is the least subtree of T satisfying the following property: $s_0 \in S_{\mu^i}$,

- For any node $s \in S_{\mu^i}$,
  - if $\widehat{\lambda}(s) = i$ then there exists a unique $s' \in S_{\mu^i}$ and action a such that $s \overset{a}{\Rightarrow}_{\mu^i} s'$.
  - if $\widehat{\lambda}(s) \neq i$ then for all s' such that $s \overset{a}{\Rightarrow} s'$, we have $s \overset{a}{\Rightarrow}_{\mu^i} s'$.

Let $\Omega^i(T)$ denote the set of all strategies for player i in the extensive form game tree T. A play $\rho : s_0 a_0 s_1 \cdots$ is said to be consistent with μ^i if for all $j \geq 0$ we have $s_j \in S^i$ implies $\mu^i(s_j) = a_j$.

2.3 Composing game trees

We consider sequential and parallel composition of game trees. In the case of sequences, composing them amounts to concatenation and interleaving. Concatenating trees is less straightforward, since each leaf node of the first is now a root of the second tree. Interleaving trees is not the same as a tree obtained by interleaving paths from the two trees, since we wish to preserve choices made by players.

Sequential composition: Suppose we are given two finite extensive form game trees $T_1 = (S_1, \Rightarrow_1, s_1^0, \widehat{\lambda}_1)$ and $T_2 = (S_2, \Rightarrow_2, s_2^0, \widehat{\lambda}_2)$. The sequential composition of T_1 and T_2 (denoted $T_1; T_2$) gives rise to a game tree $T = (S, \Rightarrow, s_0, \widehat{\lambda})$, defined as follows: $S = S_1^{NL} \cup S_2$, $s_0 = s_1^0$,

- $\widehat{\lambda}(s) = \widehat{\lambda}_1(s)$ if $s \in S_1^{NL}$ and $\widehat{\lambda}(s) = \widehat{\lambda}_2(s)$ if $s \in S_2$.
- $s \overset{a}{\Rightarrow} s'$ iff:
  - $s, s' \in S_1^{NL}$ and $s \overset{a}{\Rightarrow}_1 s'$, or
  - $s, s' \in S_2$ and $s \overset{a}{\Rightarrow}_2 s'$, or
  - $s \in S_1^{NL}$, $s' = s_2^0$ and there exists $s'' \in S_1^L$ such that $s \overset{a}{\Rightarrow}_1 s''$.

In other words, the game tree $T_1; T_2$ is generated by pasting the tree T_2 at all the leaf nodes of T_1. The definition of sequential composition can be extended to a set of trees $\mathcal{T}_2$ (denoted $T_1; \mathcal{T}_2$) with the interpretation that at each leaf node of T_1, a tree $T_2 \in \mathcal{T}_2$ is attached.

Parallel composition: The parallel composition of T_1 and T_2 (denoted $T_1 \| T_2$) yields a set of trees. A tree $t = (S, \Rightarrow, s_0, \widehat{\lambda})$ in the set of trees $T_1 \| T_2$ provided: $S \subseteq S_1 \times S_2$, $s_0 = (s_1^0, s_2^0)$,

- For all $(s, s') \in S$:
  - $E_T((s, s')) = E_{t_1}(s) \times s'$ and $\widehat{\lambda}(s, s') = \widehat{\lambda}_1(s)$, or
  - $E_T((s, s')) = s \times E_{t_2}(s')$ and $\widehat{\lambda}(s, s') = \widehat{\lambda}_2(s')$.
- For every edge $s_1 \overset{a}{\Rightarrow}_1 s_1'$ in t_1, there exists $s_2 \in S_2$ such that $(s_1, s_2) \overset{a}{\Rightarrow} (s_1', s_2)$ in t.
- For every edge $s_2 \overset{a}{\Rightarrow}_2 s_2'$ in t_2, there exists $s_1 \in S_1$ such that $(s_1, s_2) \overset{a}{\Rightarrow} (s_1, s_2')$ in t.

3 Examples

Consider the trees T_1 and T_2 given in Figure 1. The sequential composition of T_1 and T_2 (denoted $T_1; T_2$) is shown in Figure 2. This is obtained by pasting the tree T_2 at all the leaf nodes of T_1.

Now consider two finite extensive form game trees T_4 and T_5 given in figure 3. Each game is played between two players, player 2 is common in both games.

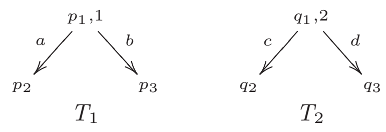


Fig. 1. atomic games

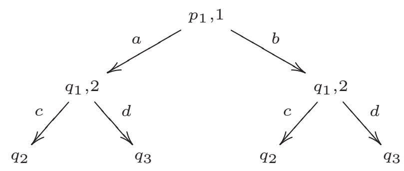


Fig. 2. T_1; T_2

Note that we are talking about different instances of the same game (as evident from the similar game trees) played between different pairs of players with a player in common. Consider the interleaving of T_4 and T_5 where player 1 moves first in T_4, followed by 2 and 3 in T_5, and then again coming back to the game T_4, with the player 2-moves. This game constitutes a valid tree in the set of trees defined by $T_4 \| T_5$ and is shown in Figure 4.

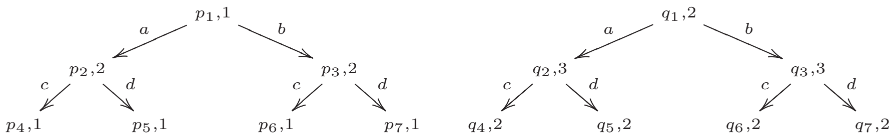
Due to space constraints, we have not provided the names for each of the states in the parallel game tree, but they are quite clear from the context. The game starts with player 1 moving from p_1 in T_4 to p_2 or p_3. Then the play moves to the game T_5, where player 2 moves to q_2 or q_3, followed by the moves of player 3. After that, the play comes back to T_4, where player 2 moves once again.

These games clearly represent toy versions of "playing against two Grandmasters simultaneously". Players 1 and 3 can be considered as the Grandmasters, and 2 as the poor mortal. Let us now describe the copycat strategy that can be used by player 2, when the two games are played in parallel. The simultaneous game (figure 4), starts with player 1 making the first move a, say in the game tree T_4 (from (p_1, q_1)) to move to (p_2, q_1). Player 2 then copies this move in game T_5, to move to (p_2, q_2). The game continues in T_5, with player 3 moving to (p_2, q_4), say. Player 2 then copies this move in T_4 (playing action c) to move to (p_4, q_4). This constitutes a play of the game, where player 2 copies the moves of players 1 and 3, respectively.

Evidently, if player 1 has a strategy in T_4 to achieve a certain objective, whatever be the moves of player 2, following the same strategy, player 2 can attain the same objective in T_5.

Parallel composition can also be performed with respect to games structures which are not the same. Consider the game trees T_6 and T_7 given in Figure 5.

An interleaved game where each game is played alternatively starting from the game T_6 can be represented by the game tree in Figure 6.



(a) T_4 (b) T_5

Fig. 3. Atomic games

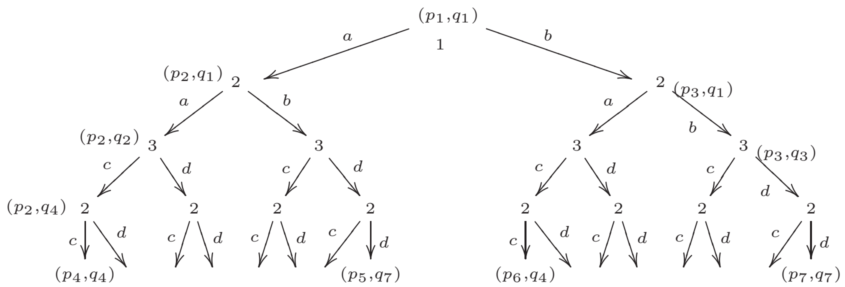


Fig. 4. Game tree T

4 The logic

For a finite set of action symbols Σ, let $\mathcal{T}(\Sigma)$ be a countable set of finite extensive form game trees over the action set Σ which is closed under subtree inclusion. That is, if $T \in \mathcal{T}(\Sigma)$ and T' is a subtree of T then $T' \in \mathcal{T}(\Sigma)$. We also assume that for each $a \in \Sigma$, the tree consisting of the single edge labelled with a is in $\mathcal{T}(\Sigma)$. Let $\mathbb{H}$ be a countable set and h, h' range over this set. Elements of $\mathbb{H}$ are referred to in the formulas of the logic and the idea is to use them as names for extensive form game trees in $\mathcal{T}(\Sigma)$. Formally we have a map $\nu : \mathbb{H} \to \mathcal{T}(\Sigma)$ which given any name $h \in \mathbb{H}$ associates a tree $\nu(h) \in \mathcal{T}(\Sigma)$. We often abuse notation and use h to also denote $\nu(h)$ where the meaning is clear from the context.

4.1 Syntax

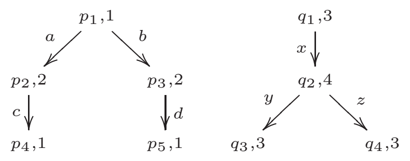
Let P be a countable set of propositions, the syntax of the logic is given by:

$$\Gamma := h \mid g_1; g_2 \mid g_1 \cup g_2 \mid g_1 \| g_2$$

$$\Phi := p \in P \mid \neg\alpha \mid \alpha_1 \vee \alpha_2 \mid \langle g, i \rangle \alpha$$

where $h \in \mathbb{H}$ and $g \in \Gamma$.

In Γ, the atomic construct h specifies a finite extensive form game tree. Composite games are then constructed using the standard dynamic logic operators



(a) T_6 (b) T_7

Fig. 5. Atomic games

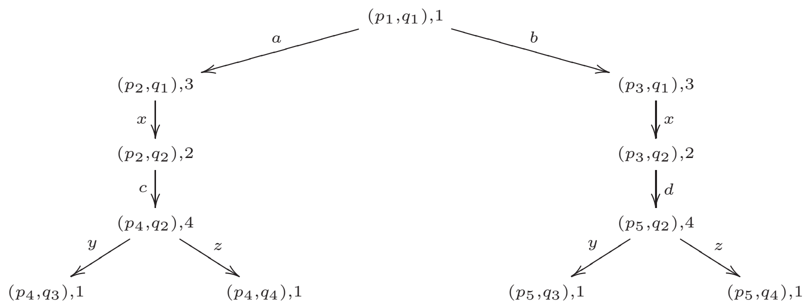


Fig. 6. A game tree in $T_6 \| T_7$

along with the parallel operator. $g_1 \cup g_2$ denotes playing g_1 or g_2. Sequential composition is denoted by $g_1; g_2$ and $g_1 \| g_2$ denotes the parallel composition of games.

The main connective $\langle g, i \rangle \alpha$ asserts at state s that a tree in g is enabled at s and that player i has a strategy subtree in it at whose leaves α holds.

4.2 Semantics

A model $M = (W, \rightarrow, \widehat{\lambda}, V)$ where W is the set of states (or game positions), $\rightarrow \subseteq W \times \Sigma \times W$ is the move relation, $V : W \rightarrow 2^P$ is a valuation function and $\widehat{\lambda} : W \rightarrow N$ is a player labelling function. These can be thought of as standard Kripke structures whose states correspond to game positions along with an additional player labelling function. An extensive form game tree can be thought of as *enabled* at a certain state, say s of a Kripke structure, if we can embed the tree structure in the tree unfolding of the Kripke structure rooted at s. We make this notion more precise below.

Enabling of trees: For a game position $u \in W$, let T_u denote the tree unfolding of M rooted at u. We say the game h is enabled at a state u if the structure $\nu(h)$ can be embedded in T_u with respect to the enabled actions and player labelling. Formally this can be defined as follows:

Given a state u and $h \in \mathbb{H}$, let $T_u = (S^s_M, \Rightarrow_M, \widehat{\lambda}_M, s)$ and $\nu(h) = T_h = (S_h, \Rightarrow_h, \widehat{\lambda}_h, s_{h,0})$. The restriction of T_u with respect to the game tree h (denoted $T_u \upharpoonright h$) is the subtree of T_s which is generated by the structure specified by T_h. The restriction is defined inductively as follows: $T_u \upharpoonright h = (S, \Rightarrow, \widehat{\lambda}, s_0, f)$ where $f : S \to S_h$. Initially $S = \{s\}$, $\widehat{\lambda}(s) = \widehat{\lambda}_M(s)$, $s_0 = s$ and $f(s_0) = s_{h,0}$.

For any $s \in S$, let $f(s) = t \in S_h$. Let $\{a_1, \ldots, a_k\}$ be the outgoing edges of t, i.e. for all $j : 1 \leq j \leq k$, $t \overset{a_j}{\Rightarrow}_h t_j$. For each a_j, let $\{s^1_j, \ldots, s^m_j\}$ be the nodes in S^s_M such that $s \overset{a_j}{\Rightarrow}_M s^l_j$ for all $l : 1 \leq l \leq m$. Add nodes $s^1_j, \ldots, s^m_j$ to S and the edges $s \overset{a_j}{\Rightarrow} s^l_j$ for all $l : 1 \leq l \leq m$. Also set $\widehat{\lambda}(s^l_j) = \widehat{\lambda}_M(s^l_j)$ and $f(s^l_j) = t_j$.

We say that a game h is enabled at u (denoted $\mathit{enabled}(h, u)$) if the tree $T_u \upharpoonright h = (S, \Rightarrow, \widehat{\lambda}, s_0, f)$ satisfies the following properties: for all $s \in S$,

- $\mathit{moves}(s) = \mathit{moves}(f(s))$,
- if $\mathit{moves}(s) \neq \emptyset$ then $\widehat{\lambda}(s) = \widehat{\lambda}_h(f(s))$.

Interpretation of atomic games: To formally define the semantics of the logic, we need to first fix the interpretation of the compositional games constructs. In the dynamic logic approach, for each game construct g and player i we would associate a relation $R^i_g \subseteq (W \times 2^W)$ which specifies the outcome of a winning strategy for player i. However due to the ability of being able to interleave game positions, in this setting we need to keep track of the actual tree structure rather just the "input-output" relations, which is closer in spirit to what is done in process logics [9] . Thus for a game g and player i we define the relation $R^i_g \subseteq 2^{(W \times W)^*}$. For a pair $\mathbf{x} = (u, w) \in W \times W$ and a set of sequences $Y \in 2^{(W \times W)^*}$ we define $(u, w) \cdot Y = \{(u, w) \cdot \rho \mid \rho \in Y\}$. For $j \in \{1, 2\}$ we use $\mathbf{x}[j]$ to denote the j-th component of $\mathbf{x}$.

For each atomic game h and each state $u \in W$, we define $R^i_h(u)$ in a bottom-up manner in such a way that whenever h is enabled at u, $R^i_h(u)$ encodes the set of all available strategies (cf. Section 2.2) for player i in the game h enabled at u. The collection of all such strategies that a player i can have, whenever the game h is enabled at some state $u \in W$ is given by R^i_h.

Let $h = (S, \Rightarrow, s_0, \widehat{\lambda})$ be a depth 1 tree with $\mathit{moves}(s_0) = \{a_1, \ldots, a_k\}$ and for all $s \neq s_0$, $\mathit{moves}(s) = \emptyset$. For $i \in N$ and a state $u \in W$, we define $R^i_h(u) \subseteq 2^{(W \times W)^*}$ as follows:

- If $\widehat{\lambda}(s_0) = i$ then $R^i_h(u) = \{X_j \mid \mathit{enabled}(h, u)$ and $X_j = \{(u, w_j)\}$ where $u \overset{a_j}{\rightarrow} w_j\}$.
- if $\widehat{\lambda}(s_0) \in \bar{\imath}$ then $R^i_h(u) = \{\{(u, w_j) \mid \mathit{enabled}(h, u)$ and $\exists a_j \in \mathit{moves}(s_0)$ with $u \overset{a_j}{\rightarrow} w_j\}\}$.

For $g \in \Gamma$, let $R^i_g = \bigcup_{u \in W} R^i_g(u)$.

For a tree $h = (S, \Rightarrow, s_0, \widehat{\lambda})$ such that $\mathit{depth}(h) > 1$, we define $R^i_h(u)$ as,

- if $\widehat{\lambda}(s_0) = i$ then $R^i_h(u) = \{\{(u, w) \cdot Y\} \mid \exists X \in R^i_{\mathit{head}(h)}$ with $(u, w) \in X, u \overset{a_j}{\rightarrow} w$ and $Y \in R^i_{h_{a_j}}\}$

– if $\widehat{\lambda}(s_0) \in \bar{\imath}$ then $R^i_h(u) = \{\{(u,w)\cdot Y \mid \exists X \in R^i_{head(h)}$ with $(u,w) \in X, u \xrightarrow{a_j} w$ and $Y \in R^i_{h_{a_j}}\}\}$.

Remark: Note that a set $X \in R^i_h$ can contain sequences such as $(u,w)(v,x)$ where $w \neq v$. Thus in general sequence of pairs of states in X need not represent a subtree of T_u for some $u \in W$. We however need to include such sequences since if h is interleaved with another game tree h', a move enabled in h' could make the transition from w to v. A sequence $\varrho \in X$ is said to be legal if whenever $(u,w)(v,x)$ is a subsequence of ϱ then $w = v$. A set $X \subseteq 2^{(W\times W)^*}$ is a valid tree if for all sequence $\varrho \in X$, ϱ is legal and X is prefix closed. For X which is a valid tree we have the property that for all $\varrho, \varrho' \in X$, $\mathit{first}(\varrho)[1] = \mathit{first}(\varrho')[1]$. We denote this state by $root(X)$. We also use $\mathit{frontier}(X)$ to denote the frontier nodes, i.e. $\mathit{frontier}(X) = \{last(\varrho)[2] \mid \varrho \in X\}$.

For a game tree h, although every set $X \in R^i_h$ need not be a valid tree, we can associate a tree structure with X (denoted $\mathfrak{T}(X)$) where the edges are labelled with pairs of the form (u,w) which appears in X. Conversely given $W \times W$ edge labelled finite game tree $\mathfrak{T}$, we can construct a set $X \subseteq 2^{(W\times W)^*}$ by simply enumerating the paths and extracting the labels of each edge in the path. We denote this translation by $\mathfrak{f}(\mathfrak{T})$. We use these two translations in what follows:

Interpretation of composite games: For $g \in \Gamma$ and $i \in N$, we define $R^i_g \subseteq 2^{(W\times W)^*}$ as follows:

– $R^i_{g_1 \cup g_2} = R^i_{g_1} \cup R^i_{g_2}$.
– $R^i_{g_1;g_2} = \{\mathfrak{f}(\mathfrak{T}(X);\mathcal{T}) \mid X \in R^i_{g_1}$ and $\mathcal{T} = \{\mathfrak{T}(X_1),\ldots,\mathfrak{T}(X_k)\}$ where $\{X_1,\ldots,X_k\} \subseteq R^i_{g_2}\}$.
– $R^i_{g_1 \| g_2} = \{\mathfrak{f}(\mathfrak{T}(X_1) \| \mathfrak{T}(X_2)) \mid X_1 \in R^i_{g_1}$ and $X_2 \in R^i_{g_2}\}$.

The truth of a formula $\alpha \in \Phi$ in a model M and a position u (denoted $M, u \models \alpha$) is defined as follows:

– $M, u \models p$ iff $p \in V(u)$.
– $M, u \models \neg\alpha$ iff $M, u \not\models \alpha$.
– $M, u \models \alpha_1 \vee \alpha_2$ iff $M, u \models \alpha_1$ or $M, u \models \alpha_2$.
– $M, u \models \langle g, i\rangle\alpha$ iff $\exists X \in R^i_g$ such that X constitutes a valid tree, $root(X) = u$ and for all $w \in \mathit{frontier}(X)$, $M, w \models \alpha$.

A formula α is satisfiable if there exists a model M and a state u such that $M, u \models \alpha$.

Let h_1 and h_2 be the game trees T_4 and T_5 given in Figure 3. The tree in which the moves of players are interleaved in lock-step synchrony is one of the trees in the semantics of $h_1 \| h_2$. This essentially means that at every other stage if a depth one tree is enabled then after that the same tree structure is enabled again, except for the player labelling. Given the (finite) atomic trees, we can write a formula α_{LS} which specifies this condition. If the tree h is a minimal one, i.e. of depth one given by $(S, \Rightarrow, s_0, \widehat{\lambda})$, α_{LS_h} can be defined as, $\bigwedge_{a_j \in moves(s_0)}(\langle a_j\rangle\top \wedge [a_j](\wedge_{a_j \in moves(s_0)}\langle a_j\rangle\top)$.

If player 1 has a strategy (playing a, say) to achieve certain objective ϕ in the game h_1, player 2 can play (copy) the same strategy in h_2 to ensure ϕ. This phenomenon can be adequately captured in the interleaved game structure, where player 2 has a strategy (viz. playing a) to end in those states of the game $h_1 \| h_2$, where player 1 can end in h_1. So we have that, whenever h_1 and $h_1 \| h_2$ are enabled and players can move in lock-step synchrony with respect to the game h_1 (or, h_2), $\langle h_1, 1\rangle\phi \rightarrow \langle h_1 \| h_2, 2\rangle\phi$ holds.

5 Axiom system

The main technical contribution of this paper is a sound and complete axiom system. Firstly, note that the logic extends standard PDL. For $a \in \Sigma$ and $i \in N$, let T_a^i be the tree defined as: $T_a^i = (S, \Rightarrow, s_0, \widehat{\lambda})$ where $S = \{s_0, s_1\}$, $s_0 \overset{a}{\Rightarrow} s_1$, $\widehat{\lambda}(s_0) = i$ and $\widehat{\lambda}(s_1) \in N$. Let t_a^i be the name denoting this tree, i.e. $\nu(t_a^i) = T_a^i$. For each $a \in \Sigma$ we define,

- $\langle a\rangle\alpha = \bigwedge_{i \in N}(\mathbf{turn}_i \supset \langle t_a^i, i\rangle\alpha)$.

From the semantics it is easy to see that we get the standard interpretation for $\langle a\rangle\alpha$, i.e. $\langle a\rangle\alpha$ holds at a state u iff there is a state w such that $u \overset{a}{\rightarrow} w$ and α holds at w.

Enabling of trees: The crucial observation is that the property of whether a game is enabled can be described by a formula of the logic. Formally, for $h \in \mathbb{H}$ such that $\nu(h) = (S, \Rightarrow, s_0, \widehat{\lambda})$ and $moves(s_0) \neq \emptyset$ and an action $a \in moves(s_0)$, let h_a be the subtree of T rooted at a node s' with $s_0 \overset{a}{\Rightarrow} s'$. The formula $h^\checkmark$ (defined below) is used to express the fact that the tree structure $\nu(h)$ is enabled and $head_h^\checkmark$ to express that $head(\nu(h))$ is enabled. This is defined as,

- If $\nu(h)$ is atomic then $h^\checkmark = \top$ and $head_h^\checkmark = \top$.
- If $\nu(h)$ is not atomic and $\widehat{\lambda}(s_0) = i$ then
  - $h^\checkmark = \mathbf{turn}_i \wedge (\bigwedge_{a_j \in moves(s_0)}(\langle a_j\rangle\top \wedge [a_j]h_{a_j}^\checkmark))$.
  - $head_h^\checkmark = \mathbf{turn}_i \wedge (\bigwedge_{a_j \in moves(s_0)} \langle a_j\rangle\top)$.

Due to the ability to interleave choices of players, we also need to define for a composite game expression g, the initial (atomic) game of g and the game expression generated after playing the initial atomic game (or in other words the residue). We make this notion precise below:

Definition of init

- $init(h) = \{h\}$ for $h \in \mathrm{G}$
- $init(g_1; g_2) = init(g_1)$ if $g_1 \neq \epsilon$ else $init(g_2)$.
- $init(g_1 \cup g_2) = init(g_1) \cup init(g_2)$.
- $init(g_1 \| g_2) = init(g_1) \cup init(g_2)$.

Definition of residue

- $h \backslash h = \epsilon$ and $\epsilon \backslash h = \epsilon$.
- $(g_1; g_2)\backslash h = \begin{cases} (g_1\backslash h); g_2 & \text{if } g_1 \neq \epsilon. \\ (g_2\backslash h) & \text{otherwise.} \end{cases}$
- $(g_1 \cup g_2)\backslash h = \begin{cases} (g_1\backslash h) \cup (g_2\backslash h) & \text{if } h \in \mathit{init}(g_1) \text{ and } h \in \mathit{init}(g_2). \\ g_1\backslash h & \text{if } h \in \mathit{init}(g_1) \text{ and } h \notin \mathit{init}(g_2). \\ g_2\backslash h & \text{if } h \in \mathit{init}(g_2) \text{ and } h \notin \mathit{init}(g_1). \end{cases}$
- $(g_1 \| g_2)\backslash h = \begin{cases} (g_1\backslash h \| g_2) \cup (g_1 \| g_2\backslash h) & \text{if } h \in \mathit{init}(g_1) \text{ and } h \in \mathit{init}(g_2). \\ (g_1\backslash h \| g_2) & \text{if } h \in \mathit{init}(g_1) \text{ and } h \notin \mathit{init}(g_2). \\ (g_1 \| g_2\backslash h) & \text{if } h \in \mathit{init}(g_2) \text{ and } h \notin \mathit{init}(g_1). \end{cases}$

The translation used to express the property of enabling of trees in terms of standard PDL formulas also suggest that the techniques developed for proving completeness of PDL can be applied in the current setting. We base our axiomatization of the logic on the "reduction axioms" methodology of dynamic logic. The most interesting reduction axiom in our setting would naturally involve the parallel composition operator. Intuitively, for game expressions g_1, g_2, a formula α and a player $i \in N$ the reduction axiom for $\langle g_1 \| g_2, i\rangle\alpha$ need to express the following properties:

- There exists an atomic tree $h \in \mathit{init}(g_1 \| g_2)$ such that $\mathit{head}(\nu(h))$ is enabled.
- Player i has a strategy in $\mathit{head}(\nu(h))$ which when composed with a strategy in the residue ensures α. We use $\mathit{comp}^i(h, g_1, g_2, \alpha)$ to denote this property and formally define it inductively as follows:

Suppose $h = (S, \Rightarrow, s_0, \widehat{\lambda})$ where $A = \mathit{moves}(s_0) = \{a_1, \ldots, a_k\}$.

- If $h \in \mathit{init}(g_1)$, $h \in \mathit{init}(g_2)$ and
  - $\widehat{\lambda}(s_0) = i$ then $\mathit{comp}^i(h, g_1, g_2, \alpha) = \bigvee_{a_j \in A}(\langle a_j\rangle\langle(h_{a_j}; (g_1\backslash h)) \| g_2\rangle\alpha \vee \langle a_j\rangle\langle g_1 \| (h_{a_j}; (g_2\backslash h))\rangle\alpha)$.
  - $\widehat{\lambda}(s_0) \in \bar{\imath}$ then $\mathit{comp}^i(h, g_1, g_2, \alpha) = \bigwedge_{a_j \in A}([a_j]\langle(h_{a_j}; (g_1\backslash h)) \| g_2\rangle\alpha \vee [a_j]\langle g_1 \| (h_{a_j}; (g_2\backslash h))\rangle\alpha)$.
- If $h \in \mathit{init}(g_1)$, $h \notin \mathit{init}(g_2)$ and
  - $\widehat{\lambda}(s_0) = i$ then $\mathit{comp}^i(h, g_1, g_2, \alpha) = \bigvee_{a_j \in A}(\langle a_j\rangle\langle(h_{a_j}; (g_1\backslash h)) \| g_2\rangle\alpha)$.
  - $\widehat{\lambda}(s_0) \in \bar{\imath}$ then $\mathit{comp}^i(h, g_1, g_2, \alpha) = \bigwedge_{a_j \in A}([a_j]\langle(h_{a_j}; (g_1\backslash h)) \| g_2\rangle\alpha)$.
- if $h \in \mathit{init}(g_2)$, $h \notin \mathit{init}(g_1)$ and
  - $\widehat{\lambda}(s_0) = i$ then $\mathit{comp}^i(h, g_1, g_2, \alpha) = \bigvee_{a_j \in A}(\langle a_j\rangle\langle g_1 \| (h_{a_j}; (g_2\backslash h))\rangle\alpha)$.
  - $\widehat{\lambda}(s_0) \in \bar{\imath}$ then $\mathit{comp}^i(h, g_1, g_2, \alpha) = \bigwedge_{a_j \in A}([a_j]\langle g_1 \| (h_{a_j}; (g_2\backslash h))\rangle\alpha)$.

Note that the semantics for parallel composition allows us to interleave subtrees of g_2 within g_1 (and vice versa). Therefore in the definition of comp^i at each stage after an action a_j, it is important to perform the sequential composition of the subtree h_{a_j} with the residue of the game expression.

The axiom schemes

(A1) Propositional axioms:
  (a) All the substitutional instances of tautologies of PC.
  (b) $\mathbf{turn}_i \equiv \bigwedge_{j \in \bar{\imath}} \neg \mathbf{turn}_j$.

(A2) Axiom for single edge games:
  (a) $\langle a \rangle(\alpha_1 \vee \alpha_2) \equiv \langle a \rangle \alpha_1 \vee \langle a \rangle \alpha_2$.
  (b) $\langle a \rangle \mathbf{turn}_i \supset [a]\mathbf{turn}_i$.

(A3) Dynamic logic axioms:
  (a) $\langle g_1 \cup g_2, i \rangle \alpha \equiv \langle g_1, i \rangle \alpha \vee \langle g_2, i \rangle \alpha$.
  (b) $\langle g_1 ; g_2, i \rangle \alpha \equiv \langle g_1, i \rangle \langle g_2, i \rangle \alpha$.
  (c) $\langle g_1 \| g_2, i \rangle \alpha \equiv \bigvee_{h \in init(g_1 \| g_2)} head_h^{\checkmark} \wedge comp^i(h, g_1, g_2, \alpha)$.

(A4) $\langle h, i \rangle \alpha \equiv h^{\checkmark} \wedge \downarrow_{(h,i,\alpha)}$.

For $h \in \mathbb{H}$ with $\nu(h) = T = (S, \Rightarrow, s_0, \widehat{\lambda})$ we define $\downarrow_{(h,i,\alpha)}$ as follow:

– $$\downarrow_{(h,i,\alpha)} = \begin{cases} \alpha & \text{if } \ moves(s_0) = \emptyset. \\ \bigvee_{a \in \Sigma} \langle a \rangle \langle h_a, i \rangle \alpha & \text{if } \ moves(s_0) \neq \emptyset \text{ and } \widehat{\lambda}(s_0) = i. \\ \bigwedge_{a \in \Sigma} [a] \langle h_a, i \rangle \alpha & \text{if } \ moves(s_0) \neq \emptyset \text{ and } \widehat{\lambda}(s_0) \in \bar{\imath}. \end{cases}$$

Inference rules

$$(MP)\ \frac{\alpha, \quad \alpha \supset \beta}{\beta} \qquad (NG)\ \frac{\alpha}{[a]\alpha}$$

Axioms (A1) and (A2) are self explanatory. Axiom (A3) constitutes the reduction axioms for the compositional operators. Note that unlike in PDL sequential composition in our setting corresponds to composition over trees. The following proposition shows that the usual reduction axiom for sequential composition remains valid.

Proposition 5.1. *The formula* $\langle g_1 ; g_2, i \rangle \alpha \equiv \langle g_1, i \rangle \langle g_2, i \rangle \alpha$ *is valid.*

Proof. Suppose $\langle g_1 ; g_2, i \rangle \alpha \supset \langle g_1, i \rangle \langle g_2, i \rangle \alpha$ is not valid. This means there exists a model M and a state u such that $M, u \models \langle g_1 ; g_2, i \rangle \alpha$ and $M, u \not\models \langle g_1, i \rangle \langle g_2, i \rangle \alpha$. From semantics we get $\exists X \in R^i_{g_1;g_2}$ such that X is a valid tree, $root(X) = u$ and for all $w \in frontier(X)$ we have $M, u \models \alpha$. By definition, X is of the form $\mathfrak{f}(\mathfrak{T}(Y); \mathcal{T})$ where $Y \in R^i_{g_1}$ and $\mathcal{T} = \{\mathfrak{T}(X_1), \ldots, \mathfrak{T}(X_k)\}$ with $\{X_1, \ldots, X_k\} \subseteq R^i_{g_2}\}$. Since X is a valid tree we have $Y, X_1, \ldots, X_k$ are valid trees. Thus we get that for all $j : 1 \leq j \leq k$, $M, root(X_j) \models \langle \xi_2, i \rangle \alpha$ and from semantics we have $M, u \models \langle g_1, i \rangle \langle g_2, i \rangle \alpha$ which gives the required contradiction.

A similar argument which makes use of the definition of R^i_g and the semantics shows that $\langle g_1, i \rangle \langle g_2, i \rangle \alpha \supset \langle g_1 ; g_2, i \rangle \alpha$ is valid.

5.1 Completeness

To show completeness, we prove that every consistent formula is satisfiable. Let α_0 be a consistent formula, and $CL(\alpha_0)$ denote the subformula closure of α_0. In addition to the usual subformula closure we also require the following: if $\langle h, i\rangle\alpha \in CL(\alpha_0)$ then $g^{\checkmark}, \downarrow_{(h,i,\alpha)} \in CL(\alpha_0)$ and if $\langle g_1 \| g_2, i\rangle\alpha \in CL(\alpha_0)$ then $\bigwedge_{h\in init(g_1\|g_2)} head_h^{\checkmark}$, $comp^i(h, g_1, g_2, \alpha) \in CL(\alpha_0)$.

Let $AT(\alpha_0)$ be the set of all maximal consistent subsets of $CL(\alpha_0)$, referred to as atoms. We use u, w to range over the set of atoms. Each $u \in AT(\alpha_0)$ is a finite set of formulas, we denote the conjunction of all formulas in u by $\widehat{u}$. For a nonempty subset $X \subseteq AT(\alpha_0)$, we denote by $\widetilde{X}$ the disjunction of all $\widehat{u}, u \in X$. Define a transition relation on $AT(\alpha_0)$ as follows: $u \xrightarrow{a} w$ iff $\widehat{u} \wedge \langle a\rangle\widehat{w}$ is consistent. Let the model $M = (W, \longrightarrow, V)$ where $W = AT(\alpha_0)$ and the valuation function V is defined as $V(w) = \{p \in P \mid p \in w\}$. Once the model is defined, the semantics (given earlier) specifies relation R_g^i. The following lemma asserts the consistency condition on elements of R_g^i.

Lemma 5.1. *For all $i \in N$, for all $h \in \mathbb{H}$, for all $X \subseteq (W \times W)^*$ with $\mathcal{X} = frontier(X)$, for all $u \in W$ the following holds:*

1. *if X is a valid tree with $root(X) = u$ and $X \in R_h^i$ then $\widehat{u} \wedge \langle h, i\rangle\widetilde{\mathcal{X}}$ is consistent.*
2. *if $\widehat{u} \wedge \langle h, i\rangle\widetilde{\mathcal{X}}$ is consistent then there exists a X' which is a valid tree with $frontier(X') \subseteq \mathcal{X}$ and $root(X') = u$ such that $X' \in R_h^i$.*

Proof. A detailed proof is given in the appendix. It essentially involves showing that the game h is enabled at the state u and that there is a strategy for player i in $T_u \upharpoonright h$ represented by the tree X whose frontier nodes are $\mathcal{X}$. The strategy tree X is constructed in stages starting at u. For any path of the partially constructed strategy tree if the paths ends in a position of player i then the path is extended by guessing a unique outgoing edge. If the position belongs to a player in $\bar{\imath}$ then all edges are taken into account.

Lemma 5.2. *For all $i \in N$, for all $g \in \Gamma$, for all $X \subseteq (W \times W)^*$ with $\mathcal{X} = frontier(X)$ and $u \in W$, if $\widehat{u} \wedge \langle h, i\rangle\widetilde{\mathcal{X}}$ is consistent then there exists X' which is a valid tree with $frontier(X') \subseteq \mathcal{X}$ and $root(X') = u$ such that $X' \in R_h^i$.*

Proof is given in the appendix.

Lemma 5.3. *For all $\langle g, i\rangle\alpha \in CL(\alpha_0)$, for all $u \in W$, $\widehat{u} \wedge \langle g, i\rangle\alpha$ is consistent iff there exists $X \in R_g^i$ which is a valid tree with $root(X) = u$ such that $\forall w \in frontier(X)$, $\alpha \in w$.*

Proof. ($\Rightarrow$) Follows from lemma 5.2.
($\Leftarrow$) Suppose there exists $X \in R_g^i$ which is a valid tree with $root(X) = u$ such that $\forall w \in frontier(X)$, $\alpha \in w$. We need to show that $\widehat{u} \wedge \langle g, i\rangle\alpha$ is consistent, this is done by induction on the structure of g.

- The case when $g = h$ follows from lemma 5.1. For $g = g_1 \cup g_2$ the result follows from axiom (A3a).
- $g = g_1; g_2$: Since $X \in R^i_{g_1;g_2}$, $\exists Y$ with $\mathit{root}(Y) = u$ and $\mathit{frontier}(Y) = \{v_2, \dots, v_k\}$, there exist sets $X_1, \dots, X_k$ where for all $j : 1 \leq j \leq k$, $\mathit{root}(X_j) = v_j$, $\bigcup_{j=1,\dots,k} \mathit{frontier}(X_j) = \mathit{frontier}(X)$, $X_j \in R^i_{g_2}$ and $Y \in R^i_{g_1}$. By induction hypothesis, for all j, $\widehat{v_j} \wedge \langle g_2 \rangle \alpha$ is consistent. Since v_j is an atom and $\langle g_2, i \rangle \alpha \in CL(\alpha_0)$, we get $\langle g_2, i \rangle \alpha \in v_j$. Again by induction hypothesis we have $\widehat{u} \wedge \langle g_1, i \rangle \langle g_2, i \rangle \alpha$ is consistent. Hence from (A3b) we have $\widehat{u} \wedge \langle g_1; g_2, i \rangle \alpha$ is consistent.
- $g = g_1 \| g_2$: Let $h \in \mathit{init}(g_1 \| g_2)$, and $h = (S, \Rightarrow, s_0, \widehat{\lambda})$. We have three cases depending on whether h is the initial constituent game in g_1 and g_2. We look at the case when $h \in \mathit{init}(g_1)$ and $h \notin \mathit{init}(g_2)$, the arguments for the remaining cases are similar. Let $A = \mathit{moves}(s_0) = \{a_1, \dots, a_k\}$. By semantics, since $\mathit{enabled}(h, u)$ holds we have $\mathit{moves}(u) = A$. We also get there exists $Y_j \in R^i_{t_{a_j};(g_1 \setminus h) \| g_2}$ where $\bigcup_{j=1,\dots,k} \mathit{frontier}(Y_j) = \mathit{frontier}(X)$. Suppose $\widehat{\lambda}(s_0) = \overline{\imath}$, by performing a second induction on the depth of X we can argue that $\widehat{u} \wedge (\bigwedge_{a_j \in A} ([a_j] \langle (t_{a_j}; (g_1 \setminus h)) \| g_2 \rangle \alpha)$ is consistent. Therefore from axiom (A3c) we have $\widehat{u} \wedge \langle g_1 \| g_2 \rangle \alpha$ is consistent.

This leads us to the following theorem from which we can deduce the completeness of the axiom system.

Theorem 5.1. *For all formulas α_0, if α_0 is consistent then α_0 is satisfiable.*

Dedidability: Given a formula α_0, let $\mathfrak{H}(\alpha_0)$ be the set of all atomic game terms appearing in α_0. Let $\mathfrak{T}(\alpha_0) = \{\nu(h) \mid h \in \mathfrak{H}(\alpha_0)\}$ and $\mathfrak{m} = \max_{T \in \mathfrak{T}(\alpha_0)} |T|$. For any finite tree T, we define $|T|$ to be the number of vertices and edges in T. It can be verified that $|CL(\alpha_0)|$ is linear in $|\alpha_0|$ and therefore we have $|AT(\alpha_0)| = \mathcal{O}(2^{|\alpha_0|})$. The states of the model M constitutes atoms of α_0 and therefore we get that if α_0 is satisfiable then there is a model whose size is at most exponential in $|\alpha_0|$. The relation R^i_g can be explicitly constructed in time $\mathcal{O}(2^{|M|^{\mathfrak{m}}})$. Thus we get the following corollary.

Corollary 5.1. *The satisfiability problem for the logic is decidable.*

6 Discussion

Iteration

An obvious extension of the logic is to add an operator for (unbounded) iteration of sequential composition. The semantics is slightly more complicated since we are dealing with trees. One needs to define it in terms of a least fixed point operator (as seen in [12]). Under this interpretation, the standard dynamic logic axiom for iteration remains valid: $\langle g^*, i \rangle \alpha \equiv \alpha \vee \langle g, i \rangle \langle g^*, i \rangle \alpha$.

We also have the familiar induction rule for dynamic logic which asserts that when α is invariant under g so it is with the iteration of g.

$(IND)\ \dfrac{\langle g, i\rangle\alpha \supset \alpha}{\langle g^*, i\rangle\alpha \supset \alpha}$

Note that the completeness proof (in the presence of interleaving) gets considerably more complicated now. Firstly, the complexity of $g\backslash h$ is no longer less than that of g so we cannot apply induction directly for parallel composition. In general when we consider $g_1^* \| g_2^*$, the interleaving critically depends on how many iterations are chosen in each of the components. The technique is to consider a graph for every g as follows: add an edge labelled h from g to $g\backslash h$. This is a finite graph, and we can show that the enabling of g at a state s corresponds to the existence of an embedding of this graph at s. In effect, the unfolding of the parallel composition axiom asserts the existence of this subgraph, and the rest of the proof uses the induction rule as in the completeness proof for dynamic logic. We omit the detailed proof here since it is technical and lengthy.

Strategy specifications

Throughout the paper we have been talking of existence of strategies in compositional games. It would be more interesting to specify strategies explicitly in terms of their properties as done in [15]. In the presence of parallel composition, this adds more value to the analysis since apart from specifying structural conditions which ensures the ability for players to copy moves, we can also specify the exact sequence of moves which are copied across games. The basic techniques used here can be extended to deal with strategy specification. However, it would be more interesting to come up with compositional operators for strategy specifications which can naturally exploit the interleaving semantics.

Acknowledgements. We thank the anonymous referees for their valuable comments and suggestions. The second author thanks the Netherlands Institute for Advanced Study in the Humanities and Social Sciences for its support.

7 Appendix

Lemma 5.1. For all $i \in N$, for all $h \in \mathbb{H}$, for all $X \subseteq (W \times W)^*$ with $\mathcal{X} = \mathit{frontier}(X)$, for all $u \in W$ the following holds:

1. if X is a valid tree with $\mathit{root}(X) = u$ and $X \in R_h^i$ then $\widehat{u} \wedge \langle h, i\rangle\widetilde{\mathcal{X}}$ is consistent.
2. if $\widehat{u} \wedge \langle h, i\rangle\widetilde{\mathcal{X}}$ is consistent then there exists a X' which is a valid tree with $\mathit{frontier}(X') \subseteq \mathcal{X}$ and $\mathit{root}(X') = u$ such that $X' \in R_h^i$.

Proof. Let $h = (S, \Rightarrow, s_0, \widehat{\lambda})$. If $\mathit{moves}(s_0) = \emptyset$ then from axiom (A4) we get $\langle h, i\rangle\alpha \equiv \beta \wedge \alpha$ and the lemma holds. Let $\mathit{moves}(s_0) = \{a_1, \ldots, a_k\}$ and $\widehat{\lambda}(s_0) = i$.

Suppose $X \in R_h^i$, since X is a valid tree and $\mathit{enabled}(\mathit{head}(h), u)$ holds, there exist sets $Y_1, \ldots, Y_k$ such that for all $j : 1 \leq j \leq k$, $w_j = \mathit{root}(Y_j)$ and $u \xrightarrow{a_j} w_j$.

Since u is an i node we have that the strategy should choose a w_j such that $u \xrightarrow{a_j} w_j$ and $X' \in R^i_{h_{a_j}}$ where $X = (u, w_j) \cdot X'$. By induction hypothesis we have $\widehat{w}_j \wedge \langle h_{a_j}, i\rangle\widetilde{\mathcal{X}}$ is consistent. Hence from axiom (A4) we conclude $\widehat{u} \wedge \langle h, i\rangle\widetilde{\mathcal{X}}$ is consistent.

Suppose $\widehat{u} \wedge \langle h, i\rangle\widetilde{\mathcal{X}}$ is consistent. From axiom (A4) it follows that there exists $w_1, \ldots, w_k$ such that for all $j : 1 \leq j \leq k$, we have $u \xrightarrow{a_j} w_j$ and hence $enabled(h, u)$ holds. Let $\mathcal{X} = \{v_1, \ldots, v_m\}$, from axiom (A4) we have $\widehat{u} \wedge (\bigvee_{a \in \Sigma} \langle a\rangle\langle h_a, i\rangle\widetilde{\mathcal{X}})$ is consistent. Hence we get that there exists w_j such that $u \xrightarrow{a_j} w_j$ and $\widehat{w}_j \wedge \langle h_a, i\rangle\widetilde{\mathcal{X}}$ is consistent. By induction hypothesis there exists X' which is a valid tree with $frontier(X') \subseteq \mathcal{X}$, $root(X') = w_j$ and $X' \in R^i_{h_a}$. By definition of R^i we get $(u, w_j) \cdot X' \in R^i_h$.

Let $\widehat{\lambda}(s_0) = \bar{\imath}$ and suppose $X \in R^i_h$. Since $enabled(head(h), u)$ holds and X is a valid tree, there exist sets $Y_1, \ldots, Y_k$ such that for all $j : 1 \leq j \leq k$, $w_j = root(Y_j)$ and $u \xrightarrow{a_j} w_j$. Since u is an $\bar{\imath}$ node, any strategy of i need to have all the branches at u (by definition of strategy). Thus we get: for all w_j with $u \xrightarrow{a_j} w_j$, there exists X_j with $root(X_j) = w_j$ such that $X_j \in R^i_h$ and $X = \bigcup_{j=1,\ldots,k} (u, w_j) \cdot X_j$. By induction hypothesis and the fact that $\mathcal{X}_j = frontier(X_j) \subseteq \mathcal{X}$, we have $\widehat{w}_j \wedge \langle h, i\rangle\widetilde{\mathcal{X}}$ is consistent. Hence from axiom (A4) we get $\widehat{u} \wedge \langle h, i\rangle\widetilde{\mathcal{X}}$ is consistent.

Likewise, using axiom (A4) we can show that if $\widehat{u} \wedge \langle h, i\rangle\widetilde{\mathcal{X}}$ is consistent then there exists a X' which is a valid tree with $frontier(X') \subseteq \mathcal{X}$ and $root(X') = u$ such that $X' \in R^i_h$.

Lemma 5.2. For all $i \in N$, for all $g \in \Gamma$, for all $X \subseteq (W \times W)^*$ with $\mathcal{X} = frontier(X)$ and $u \in W$, if $\widehat{u} \wedge \langle h, i\rangle\widetilde{\mathcal{X}}$ is consistent then there exists X' which is a valid tree with $frontier(X') \subseteq \mathcal{X}$ and $root(X') = u$ such that $X' \in R^i_h$.

Proof. By induction on the structure of g.

- $g = h$: The claim follows from Lemma 5.1 item 2.
- $g = g_1 \cup g_2$: By axiom (A3a) we get $\widehat{u} \wedge \langle g_1, i\rangle\widetilde{\mathcal{X}}$ is consistent or $\widehat{u} \wedge \langle g_2, i\rangle\widetilde{\mathcal{X}}$ is consistent. By induction hypothesis there exists X_1 which is a valid tree with $frontier(X_1) \subseteq \mathcal{X}$ and $root(X_1) = u$ such that $(u, X_1) \in R^i_h$ or there exists X_2 which is a valid tree with $frontier(X_2) \subseteq \mathcal{X}$ and $root(X_2) = u$ such that $X_2 \in R^i_h$. Hence we have $X_1 \in R^i_{g_1 \cup g_2}$ or $X_2 \in R^i_{g_1 \cup g_2}$.
- $g = g_1; g_2$: By axiom (A3b), $\widehat{u} \wedge \langle g_1, i\rangle\langle g_2, i\rangle\widetilde{\mathcal{X}}$ is consistent. Hence $\widehat{u} \wedge \langle g_1, i\rangle(\bigvee(\widehat{w} \wedge \langle g_2, i\rangle\widetilde{\mathcal{X}}))$ is consistent, where the join is taken over all $w \in \mathcal{Y} = \{w \mid w \wedge \langle g_2, i\rangle\widetilde{\mathcal{X}}$ is consistent $\}$. So $\widehat{u} \wedge \langle g_1, i\rangle\widetilde{\mathcal{Y}}$ is consistent. By induction hypothesis, there exists Y' which is a valid tree with $\mathcal{Y}' = frontier(Y') \subseteq \mathcal{Y}$ and $root(Y') = u$ such that $(u, Y') \in R^i_{g_1}$. We also have that for all $w \in \mathcal{Y}$, $\widehat{w} \wedge \langle g_2, i\rangle\widetilde{\mathcal{X}}$ is consistent. Therefore we get for all $w_j \in \mathcal{Y}' = \{w_1, \ldots, w_k\}$, $\widehat{w}_j \wedge \langle g_2, i\rangle\widetilde{\mathcal{X}}$ is consistent. By induction hypothesis, there exists X_j which is a valid tree with $\mathcal{X}_j = frontier(X_j) \subseteq \mathcal{X}$ and $root(\mathcal{X}_j) = w_j$ such that

$X_j \in R^i_{g_2}$. Let X' be the tree in $Y'; \{X_j \mid j = 1, \ldots, k\}$ obtained by pasting X_j to the leaf node w_j in Y'. We get $X' \in R^i_{g_1;g_2}$.

– $g = g_1 \| g_2$: Note that for all $g \in \Gamma$ and $h \in \mathit{head}(g)$, the complexity of $g \backslash h$ is less than that of g. Therefore by making use of axiom (A3c) we can show that there exists X' with $\mathit{frontier}(X') \subseteq \mathcal{X}'$ and $\mathit{root}(X') = u$ such that $X' \in R^i_h$.

References

1. K. Abrahamson. *Decidability and expressiveness of logics of processes*. PhD thesis, Dept. of Computer Science, Univ. of Washington, 1980.
2. T. Ågotnes. Action and knowledge in alternating time temporal logic. *Synthese*, 149(2):377–409, 2006.
3. R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49:672–713, 2002.
4. J. Broersen. CTL.STIT: Enhancing ATL to express important multi-agent system verification properties. In *Proceedings of AAMAS-2010*. ACM Press, 2010.
5. J. Broersen, A. Herzig, and N. Troquard. Embedding Alternating-time Temporal Logic in strategic STIT logic of agency. *Journal of Logic and Computation*, 16(5):559–578, 2006.
6. R. Danecki. Nondeterministic propositional dynamic logic with intersection is decidable. In *Proc. 5th Symposium in Computation Theory*, Lecture Notes in Computer Science, pages 34–53. Springer, 1984.
7. V. Goranko. Coalition games and alternating temporal logics. In *Proceedings of TARK-2001*, pages 259–272, 2001.
8. D. Harel. Dynamic logic. *Handbook of Philosophical Logic*, 2:496–604, 1984.
9. D. Harel, D. Kozen, and R. Parikh. Process logic: Expressiveness, decidability, completeness. *Journal of Computer and System Sciences*, 25(2):144–170, 1982.
10. J. Horty. *Agency and Deontic Logic*. Oxford University Press, 2001.
11. M. Lange and C. Lutz. 2-EXPTIME lower bounds for propositional dynamic logics with intersection. *Journal of Symbolic Logic*, 70(4):1072–1086, 2005.
12. R. Parikh. The logic of games and its applications. *Annals of Discrete Mathematics*, 24:111–140, 1985.
13. M. Pauly. *Logic for Social Software*. PhD thesis, Univ. of Amsterdam, 2001.
14. D. Peleg. Concurrent dynamic logic. *Journal of the ACM*, 34(2):450–479, 1987.
15. R. Ramanujam and S. Simon. Dynamic logic on games with structured strategies. In *Proceedings of KR-08*, pages 49–58. AAAI Press, 2008.
16. J. van Benthem. Extensive games as process models. *Journal of Logic Language and Information*, 11:289–313, 2002.
17. J. van Benthem, S. Ghosh, and F. Liu. Modelling simultaneous games with dynamic logic. *Synthese (Knowledge, Rationality and Action)*, 165:247–268, 2008.
18. W. van der Hoek, W. Jamroga, and M. Wooldridge. A logic for strategic reasoning. *Proceedings of AAMAS-2005*, pages 157–164, 2005.
19. D. Walther, W. van der Hoek, and M. Wooldridge. Alternating-time temporal logic with explicit strategies. In *Proceedings of TARK-2007*, pages 269–278, 2007.