A logical structure for strategies

R. Ramanujam and Sunil Simon

[1] The Institute of Mathematical Sciences
C.I.T. Campus, Chennai 600 113, India.
{jam,sunils}@imsc.res.in

Abstract

We consider a logic for reasoning about composite strategies in games, where players' strategies are like programs, composed structurally. These depend not only on conditions that hold at game positions but also on properties of other players' strategies. We present an axiomatization for the logic and prove its completeness.

1 Summary

Extensive form turn-based games are trees whose nodes are game positions and branches represent moves of players. With each node is associated a player whose turn it is to move at that game position. A player's strategy is then simply a subtree which contains a unique successor for every node where it is this player's turn to make a move, and contains all successors (from the game tree) for nodes where other players make moves. Thus a strategy is an advice function that tells a player what move to play when the game reaches any specific position. In two-player win/loss games, analysis of the game amounts to seeing if either player has a winning strategy from any starting position, and if possible, synthesize such a winning strategy.

In multi-player games where the outcomes are not merely winning and losing, the situation is less clear. Every player has a preference for certain outcomes and hence cooperation as well as conflict become strategically relevant. Moreover, each player has some expectations (and assumptions) about strategies adopted by other players, and fashions her response appropriately. In such situations, game theory tries to explain what *rational* players would do.

In so-called small (normal form) games, where the game consists of a small fixed number of moves (often one move chosen independently by each player), strategies have little structure, and prediction of stable behaviour (equilibrium strategy profiles) is possible. However, this not only becomes difficult in games with richer structure and long sequences of moves, it is also less clear how to postulate behaviour of rational players. Moreover, if we look to game theory not only for existence of equilibria but also *advice* to players on how to play, the structure of strategies followed by players becomes relevant.

Even in games of perfect information, if the game structure is sufficiently rich, we need to re-examine the notion of strategy as a function that determines a player's move in every game position. Typically, the game position is itself only partially known, in terms of properties that the player can test for. Viewed in this light, strategies are like programs, built up systematically from atomic decisions like *if* b *then* a where b is a condition checked by the player to hold (at some game position) and a is a move available to the player at that position.

There is another dimension to strategies, namely that of responses to other players' moves. The notion of each player independently deciding on a strategy needs to be re-examined as well. A player's chosen strategy depends on the player's perception of apparent strategies followed by other players. Even when opponents' moves are visible, an opponent's strategy is not known completely as a function. Therefore the player's strategy is necessarily partial as well.

The central idea of this paper is to suggest that it helps to study strategies given by their properties. Hence, assumptions about strategies can be partial, and these assumptions can in turn be structurally built into the specification of other strategies. This leads us to proposing a logical structure for strategies, where we can reason with assertions of the form "(partial) strategy σ ensures the (intermediate) condition α".

This allows us to look for *induction principles* which can be articulated in the logic. For instance, we can look at what conditions must be maintained locally (by one move) to influence an outcome eventually. Moreover, we can compare strategies in terms of what conditions they can enforce.

The main contributions of this paper are:

- We consider non-zero-sum games over finite graphs, and consider best response strategies (rather than winning strategies).
- The reasoning carried out works explicitly with the structure of strategies rather than existence of strategies.
- We present a logic with structured strategy specifications and formulas describe how strategies ensure outcomes.
- We present an axiom system for the logic and prove that it is complete.

1.1 Other work

Games are quite popularly used to solve certain decision problems in logic. Probably the best example of a logical game is the Ehrenfeucht-Fraïssé game which is played on two structures to check whether a formula of a certain logic can distinguish between these structures ([Ehr61]). Games are also used as tools to solve the satisfiability and model checking questions for various modal and temporal logics ([Lan02]). Here, an existential and a universal player play on a formula to decide if the formula is satisfiable. The satisfiability problem is then characterised by the question of whether the existential player has a winning strategy in the game. These kinds of games designed specifically for semantic evaluation are generally called *logic games*.

Recently, the advent of computational tasks on the world-wide web and related security requirements have thrown up many game theoretic situations. For example, signing contracts on the web requires interaction between principals who do not know each other and typically distrust each other. Protocols of this kind which involve *selfish agents* can be easily viewed as strategic games of imperfect information. These are complex interactive processes which critically involve players reasoning about each others' strategies to decide on how to act. In this approach, instead of designing games to solve specific logical tasks, one can use logical systems to study structure of games and to reason about them.

Game logics are situated in this context, employing modal logics (in the style of logics of programs) to study logical structure present in games. Parikh's work on propositional game logic ([Par85]) initiated the study of game structure using algebraic properties. Pauly ([Pau01]) has built on this to provide interesting relationships between programs and games, and to describe coalitions to achieve desired goals. Bonnano ([Bon91]) suggested obtaining game theoretic solution concepts as characteristic formulas in modal logic. van Benthem ([vB01]) uses dynamic logic to describe games as well as strategies. van Ditmarsch ([vD00]) uses a dynamic epistemic language to study complex information change caused by actions in games. The relationship between games defined by game logics and that of logic games, is studied by van Benthem in ([vB03]).

On the other hand, the work on Alternating Temporal Logic ([AHK02]) considers selective quantification over paths that are possible outcomes of games in which players and an environment alternate moves. Here, we talk of the existence of a strategy for a coalition of players to force an outcome. [Gor01] draws parallels between these two lines of work, that of Pauly's coalition logics and alternating temporal logic. It is to be noted that in these logics, the reasoning is about *existence* of strategies, and the strategies themselves do not figure in formulas.

In the work of [HvdHMW03] and [vdHJW05], van der Hoek and co-authors develop logics for strategic reasoning and equilibrium concepts and this line of work is closest to ours in spirit. Our point of departure is in bringing logical structure into strategies rather than treating strategies as atomic. In particular, the strategy specifications we use are partial (in the sense that a player may assume that an opponent plays a whenever p holds, without knowing under what conditions the opponent strategy picks another move b), allowing for more generality in reasoning. In the context of programs, logics like propositional dynamic logic [HKT00] explicitly analyse the structure of programs. This approach has been very useful in program verification.

2 Game Arenas

We begin with a description of game models on which formulas of the logic will be interpreted. We use the graphical model for extensive form turn-based multiplayer games, where at most one player gets to move at each game position.

Game Arena

Let $N = \{1, 2, \ldots, n\}$ be a non-empty finite set of players and $\Sigma = \{a_1, a_2, \ldots, a_m\}$ be a finite set of action symbols, which represent *moves* of players. A **game arena** is a finite graph $\mathcal{G} = (W, \longrightarrow, w_0, \chi)$ where W is the set of nodes which represents the *game positions*, $\longrightarrow: (W \times \Sigma) \rightarrow W$ is a function also called the move function, w_0 is the initial node of the game.

Let the set of successors of $w \in W$ be defined as $\overrightarrow{w} = \{w' \in W \mid w \xrightarrow{a} w'$ for some $a \in \Sigma\}$. A node w is said to be *terminal* if $\overrightarrow{w} = \varnothing$. $\chi : W \rightarrow N$ assigns to each node w in W the player who "owns" w: that is, if $\chi(w) = k$ and w is not terminal then player k has to pick a move at w.

In an arena defined as above, the play of a game can be viewed as placing a token on w_0. If player k owns the game position w_0 i.e $\chi(w_0) = k$ and she picks an action 'a' which is enabled for her at w_0, then the new game position moves the token to w' where $w_0 \xrightarrow{a} w'$. A play in the arena is simply a sequence of such moves. Formally, a play in $\mathcal{G}$ is a finite path $\rho = w_0 \xrightarrow{a_1} w_1 \xrightarrow{a_2} \ldots \xrightarrow{a_k} w_k$ where w_k is terminal, or it is an infinite path $\rho = w_0 \xrightarrow{a_1} w_1 \xrightarrow{a_2} \ldots$ where $\forall i : w_i \xrightarrow{a_i} w_{i+1}$ holds. Let *Plays* denote the set of all plays in the arena.

With a game arena $\mathcal{G} = (W, \longrightarrow, w_0, \chi)$, we can associate its tree unfolding also referred to as the extensive form game tree $\mathcal{T} = (S, \Rightarrow, s_0, \lambda)$ where $(S, \Rightarrow)$ is a countably infinite tree rooted at s_0 with edges labelled by Σ and $\lambda : S \rightarrow W$ such that:

- $\lambda(s_0) = w_0$.
- For all $s, s' \in S$, if $s \stackrel{a}{\Longrightarrow} s'$ then $\lambda(s) \xrightarrow{a} \lambda(s')$.
- If $\lambda(s) = w$ and $w \xrightarrow{a} w'$ then there exists $s' \in S$ such that $s \stackrel{a}{\Longrightarrow} s'$ and $\lambda(s') = w'$.

Given the tree unfolding of a game arena $\mathcal{T}$, a node s in it, we can define the *restriction* of $\mathcal{T}$ to s, denoted $\mathcal{T}_s$ to be the subtree obtained by retaining only the unique path from root s_0 to s and the subtree rooted at s.

Games and Winning Conditions

Let $\mathcal{G}$ be an arena as defined above. The arena merely defines the rules about how the game progresses and terminates. More interesting are winning conditions, which specify the game outcomes. We assume that each player has a preference relation over the set of plays. Let $\preceq^i \subseteq (Plays \times Plays)$ be a complete, reflexive, transitive binary relation denoting the preference relation of player i. Then the game G is given as, $G = (\mathcal{G}, \{\preceq^i\}_{i \in N})$.

Then a game is defined as the pair $G = (\mathcal{G}, (\preceq^i)_{i \in N})$.

Strategies

For simplicity we will restrict ourselves to two player games, i.e. $N = \{1, 2\}$. It is easy to extend the notions introduced here to the general case where we have n players.

Let the game graph be represented by $\mathcal{G} = (W^1, W^2, \longrightarrow, s_0)$ where W^1 is the set of positions of player 1, W^2 that of player 2. Let $W = W^1 \cup W^2$.

Let $\mathcal{T}$ be the tree unfolding of the arena and s_1 a node in it. A strategy for player 1 at node s_1 is given by: $\mu = (S^1_\mu, S^2_\mu, \Rightarrow_\mu, s_1)$ is a subtree of $\mathcal{T}_{s_1}$ which contains the unique path from root s_0 to s_1 in $\mathcal{T}$ and is the least subtree satisfying the following properties:

- $s_1 \in S^1_\mu$, where $\chi(\lambda(s_1)) = 1$.
- For every s in the subtree of $\mathcal{T}_G$ rooted at s_1,
  - if $s \in S^1_\mu$ then for some $a \in \Sigma$, for each s' such that $s \stackrel{a}{\Longrightarrow} s'$, we have $s \stackrel{a}{\Longrightarrow}_\mu s'$.
  - if $s \in S^2_\mu$, then for every $b \in \Sigma$, for each s' such that $s \stackrel{b}{\Longrightarrow} s'$, we have $s \stackrel{b}{\Longrightarrow}_\mu s'$.

Let Ω_i denote the set of all strategies of Player i in G, for $i = 1, 2$. A strategy profile $\langle \mu, \tau \rangle$ defines a unique play ρ^τ_μ in the game $\mathcal{G}$.

3 The logic

We now present a logic for reasoning about composite strategies. The syntax of the logic is presented in two layers, that of strategy specification and game formulas.

Atomic strategy formulas specify, for a player, what conditions she tests for before making a move. Since these are intended to be bounded memory strategies, the conditions are stated as past time formulas of a simple tense logic. Composite strategy specifications are built from atomic ones using connectives (without negation). We crucially use an implication of the form: "if the opponent's play conforms to a strategy π then play σ".

Game formulas describe the game arena in a standard modal logic, and in addition specify the result of a player following a particular strategy at a game position, to choose a specific move a, to *ensure* an intermediate outcome α . Using these formulas one can specify how a strategy helps to eventually *win* an outcome α.

Before we describe the logic and give its semantics, some prelimiaries will be useful. Below, for any countable set X, let $Past(X)$ be a set of formulas given by the following syntax:

$$\psi \in Past(X) := x \in X \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \diamondminus\psi.$$

Such past formulas can be given meaning over finite sequences. Given any sequence $\xi = t_0 t_1 \cdots t_m$, $V : \{t_0, \cdots, t_m\} \to 2^X$, and k such that $0 \leq k \leq m$, the truth of a past formula $\psi \in Past(X)$ at k, denoted $\xi, k \models \psi$ can be defined as follows:

- $\xi, k \models p$ iff $p \in V(t_k)$.
- $\xi, k \models \neg\psi$ iff $\xi, k \not\models \psi$.
- $\xi, k \models \psi_1 \vee \psi_2$ iff $\xi, k \models \psi_1$ or $\xi, k \models \psi_2$.
- $\xi, k \models \diamondminus\psi$ iff there exists a $j : 0 \leq j \leq k$ such that $\xi, j \models \psi$.

Strategy specifications

For simplicity of presentation, we stick with two player games, where the players are Player 1 and Player 2. Let $\bar{\imath} = 2$ when $i = 1$ and $\bar{\imath} = 1$ when $i = 2$.

Let $P^i = \{p_0^i, p_1^i, \ldots\}$ be a countable set of proposition symbols where $\tau_i \in P_i$, for $i \in \{1, 2\}$. Let $P = P^1 \cup P^2 \cup \{leaf\}$. τ_1 and τ_2 are intended to specify, at a game position, which player's turn it is to move. *leaf* specifies whether the position is a terminal node.

Further, the logic is parametrized by the finite alphabet set $\Sigma = \{a_1, a_2, \ldots, a_m\}$ of players' moves and we only consider game arenas over Σ.

Let $Strat^i(P^i)$, for $i = 1, 2$ be the set of strategy specifications given by the following syntax:

$$Strat^i(P^i) := [\psi \mapsto a_k]^i \mid \sigma_1 + \sigma_2 \mid \sigma_1 \cdot \sigma_2 \mid \pi \Rightarrow \sigma$$

where $\pi \in Strat^{\bar{\imath}}(P^1 \cap P^2)$, $\psi \in Past(P^i)$ and $a_k \in \Sigma$.

The idea is to use the above constructs to specify properties of strategies. For instance the interpretation of a player i specification $[p \mapsto a]^i$ will be to choose move "a" for every i node where p holds. $\pi \Rightarrow \sigma$ would say, at any node player i sticks to the specification given by σ if on the history of the play, all moves made by $\bar{i}$ conforms to π. In strategies, this captures the aspect of players actions being responses to the opponents moves. As the opponents complete strategy is not available, the player makes a choice taking into account the apparent behaviour of the opponent on the history of play.

For a game tree $\mathcal{T}$, a node s and a strategy specification $\sigma \in \mathit{Strat}^i(P^i)$, we define $\mathcal{T}_s \upharpoonright \sigma = (S_\sigma, \Longrightarrow_\sigma, s_0)$ to be the least subtree of $\mathcal{T}_s$ which contains $\rho_{s_0}^s$ (the unique path from s_0 to s) and closed under the following condition.

- For every s' in S_σ such that $s \Longrightarrow_\sigma^* s'$,
  - s' is an i node: $s' \xrightarrow{a} s''$ and $a \in \sigma(s') \Leftrightarrow s' \xrightarrow{a}_\sigma s''$.
  - s' is an $\bar{i}$ node: $s' \xrightarrow{a} s'' \Leftrightarrow s' \xrightarrow{a}_\sigma s''$.

Given a game tree $\mathcal{T}$ and a node s in it, let $\rho_{s_0}^s : s_0 \xrightarrow{a_1} s_1 \cdots \xrightarrow{a_m} s_m = s$ denote the unique path from s_0 to s. For a strategy specification $\sigma \in \mathit{Strat}^i(P^i)$ and a node s we define $\sigma(s)$ as follows:

- $[\psi \mapsto a]^i(s) = \begin{cases} \{a\} & \text{if } \; s \in W^i \text{ and } \rho_{s_0}^s, m \models \psi \\ \Sigma & \text{otherwise} \end{cases}$
- $(\sigma_1 + \sigma_2)(s) = \sigma_1(s) \cup \sigma_2(s)$.
- $(\sigma_1 \cdot \sigma_2)(s) = \sigma_1(s) \cap \sigma_2(s)$.
- $(\pi \Rightarrow \sigma)(s) = \begin{cases} \sigma(s) & \text{if } \; \forall j : 0 \leq j < m,\, a_j \in \pi(s_j) \\ \Sigma & \text{otherwise} \end{cases}$

We say that a path $\rho_s^{s'} : s = s_1 \xrightarrow{a_1} s_2 \cdots \xrightarrow{a_{m-1}} s_m = s'$ in $\mathcal{T}$ conforms to σ if $\forall j : 1 \leq j < m$, $a_j \in \sigma(s_j)$. When the path constitutes a proper play, i.e. when $s = s_0$, we say that the play conforms to σ.

Syntax

The syntax of the logic is given by:

$$\Pi := p \in P \mid \neg\alpha \mid \alpha_1 \vee \alpha_2 \mid \langle a \rangle \alpha \mid \langle \overline{a} \rangle \alpha \mid \diamondminus \alpha \mid (\sigma)_i : c \mid \sigma \leadsto_i \beta$$

where $c \in \Sigma$, $\sigma \in \mathit{Strat}^i(P^i)$, $\beta \in \mathit{Past}(P^i)$. The derived connectives $\wedge$ $\supset$ and $[a]\alpha$ are defined as usual. Let $\diamondminus\alpha = \neg\boxminus\neg\alpha$, $\langle N \rangle \alpha = \bigvee_{a \in \Sigma} \langle a \rangle \alpha$, $[N]\alpha = \neg\langle N \rangle \neg\alpha$, $\langle P \rangle \alpha = \bigvee_{a \in \Sigma} \langle \overline{a} \rangle \alpha$ and $[P] = \neg\langle P \rangle \neg\alpha$.

The formula $(\sigma)_i : c$ asserts, at any game position, that the strategy specification σ for player i suggests that the move c can be played at that position. The formula $\sigma \leadsto_i \beta$ says that from this position, there is a way of following the strategy σ for player i so as to ensure the outcome β. These two modalities constitute the main constructs of our logic.

Semantics

The models for the logic are extensive form game trees along with a valuation function. A model $M = (\mathcal{T}, V)$ where $\mathcal{T} = (S^1, S^2, \longrightarrow, s_0)$ is a game tree as defined in section 2, and $V : S \to 2^P$ is the valuation function, such that:

- For $i \in \{1, 2\}$, $\tau_i \in V(s)$ iff $s \in S^i$.
- $leaf \in V(s)$ iff $moves(s) = \varnothing$.

where for any node s, $moves(s) = \{a | s \stackrel{a}{\Longrightarrow} s'\}$.

The truth of a formula $\alpha \in \Pi$ in a model M and position s (denoted $M, s \models \alpha$) is defined by induction on the structure of α, as usual. Let $\rho_{s_0}^{s}$ be $s_0 \stackrel{a_0}{\Longrightarrow} s_1 \cdots \stackrel{a_{m-1}}{\Longrightarrow} s_m = s$.

- $M, s \models p$ iff $p \in V(s)$.
- $M, s \models \neg\alpha$ iff $M, s \not\models \alpha$.
- $M, s \models \alpha_1 \vee \alpha_2$ iff $M, s \models \alpha_1$ or $M, s \models \alpha_2$.
- $M, s \models \langle a \rangle \alpha$ iff there exists $s' \in W$ such that $s \stackrel{a}{\to} s'$ and $M, s' \models \alpha$.
- $M, s \models \langle \overline{a} \rangle \alpha$ iff $m > 0$, $a = a_{m-1}$ and $M, s_{m-1} \models \alpha$.
- $M, s \models \diamondminus\alpha$ iff there exists $j : 0 \leq j \leq m$ such that $M, s_j \models \alpha$.
- $M, s \models (\sigma)_i : c$ iff $c \in \sigma(s)$.
- $M, s \models \sigma \leadsto_i \beta$ iff
  - for all s' in $\mathcal{T}_s \upharpoonright \sigma$, such that $s \Longrightarrow^* s'$, we have $M, s' \models \beta \wedge (\tau_i \supset enabled_\sigma)$.

where $enabled_\sigma \equiv \bigvee_{a \in \Sigma} (\langle a \rangle True \wedge (\sigma)_i : a)$.

The notions of satisfiablility and validity can be defined in the standard way. A formula α is satisfiable iff there exists a model M, there exists s such that $M, s \models \alpha$. A formula α is said to be valid iff for all models M, for all s, we have $M, s \models \alpha$.

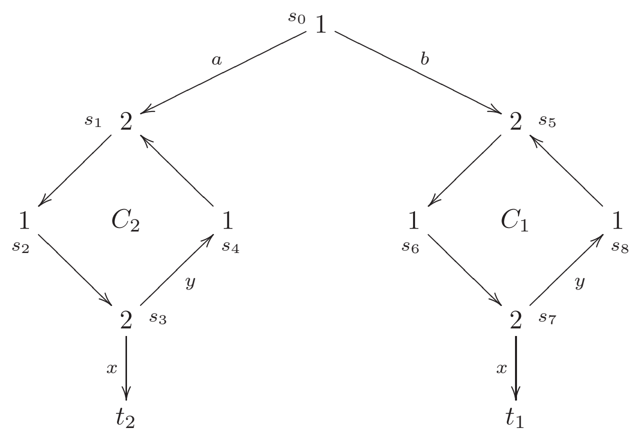

FIGURE 1.

4 Example

Probably the best way to illustrate the notion of strategy specification is to look at heuristics used in large games like chess, go, checkers, etc. A heuristic strategy is basically a partial specification, since it involves checking local properties like patterns on the board and specifying actions when certain conditions hold. For instance, a typical strategy specification for chess would be of the form:

- If a pawn double attack is possible then play the action resulting in the fork.

Note that the above specification is in contrast with a specific advice of the form:

- If a pawn is on f2 and the opponent rook and knight are on e5 and g5 respectively then move f2-f4.

A strategy would prescribe such specific advice rather than a generic one based on abstract game position properties. Heuristics are usually employed when the game graph being analysed is too huge for a functional strategy to be specified. However, we refrain from analysing chess here due to the difficulty in formally presenting the game arena and the fact that it fails to give much insight into the working of our logic. Below we look at a few simple examples which illustrates the logic.

Example 4.1. Consider the game shown in Figure 1. Players alternate moves with 1 starting at s_0. There are two cycles $C_1 : s_5 \rightarrow s_6 \rightarrow s_7 \rightarrow$

$s_8 \to s_5$, $C_2 : s_1 \to s_2 \to s_3 \to s_4 \to s_1$ and two terminal nodes t_1 and t_2. Let the preference ordering of player 1 be $t_1 \preceq^1 t_2 \preceq^1 C_2 \preceq^1 C_1$. As far as player 2 is concerned $t_1 \preceq^2 C_1$ and he is indifferent between C_2 and t_2. However, he prefers C_2 or t_2 over $\{C_1, t_1\}$. Equilibrium reasoning will advise player 1 to choose the action "b" at s_0 since at position s_7 it is irrational for 2 to move x as it will result in 2's worst outcome. However the utility difference between C_1 and t_1 for 2 might be negligible compared to the incentive of staying in the "left" path. Therefore 2 might decide to punish 1 for moving b when 1 knew that $\{C_2, t_2\}$ was equally preferred by 2. Even though t_1 is the worst outcome, at s_7 player 2 can play x to implement the punishment. Let $V(p_j) = \{s_3, s_7\}, V(p_{init}) = \{s_0\}, V(p_{good}) = \{s_0, s_1, s_2, s_3, s_4\}$ and $V(p_{punish}) = \{s_0, s_5, s_6, s_7, t_1\}$. The local objective of 2 will be to remain on the good path or to implement the punishment. Player 2 strategy specification can be written as

$$\pi \equiv ([p_{init} \mapsto b]^1 \Rightarrow [p_j \mapsto x]^2) \cdot ([p_{init} \mapsto a]^1 \Rightarrow [p_j \mapsto y]^2).$$

We get that $\pi \rightsquigarrow_2 (p_{good} \vee p_{punish})$. Player 1 if he knows 2's strategy might be tempted to play "a" at s_0 by which the play will end up in C_2. Let the proposition p_{worst} hold at t_1 which is the worst outcome for player 1. Then we have $[p_{init} \mapsto a]^1 \rightsquigarrow_1 \neg p_{worst}$. This says that if player 1 chooses a at the initial position then he can ensure that the worst outcome is avoided.

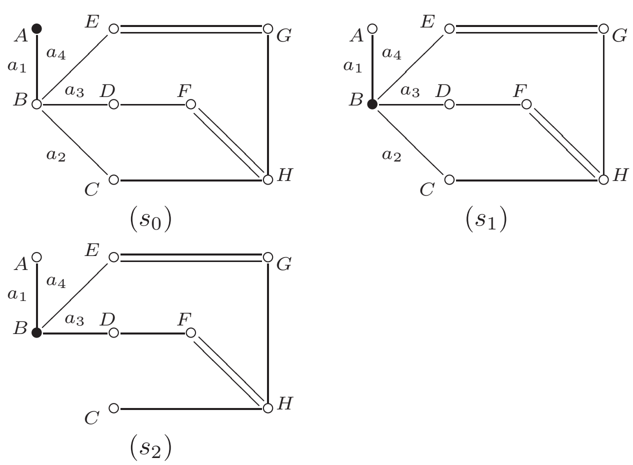

Figure 2. Sabotage Game

Example 4.2. The sabotage game [Ben02] is a two player zero sum game where one player moves along the edges of a labelled graph and the other player removes an edge in each round. Formally let a Σ labelled graph R for some alphabet set Σ is $R = (V, e)$ where V is the set of vertices and $e : V \times \Sigma \rightarrow V$ is the edge function. The sabotage game is played as follows: initially we consider the graph $R_0 = (V_0, e_0, v_0)$. There are two players, *Runner* and *Blocker* who move alternatingly where the *Runner* starts the run from vertex v_0. In round n the *Runner* moves one step further along an existing edge of the graph. i.e., he chooses a vertex $v_{n+1} \in V$ such that there exists some $a \in \Sigma$ with $e_n(v_n, a) = v_{n+1}$. Afterwards the *Blocker* removes one edge of the graph. i.e., he chooses two vertices u and v such that for some $a \in \Sigma$, $e_n(u, a) = v$ and defines the edge function e_{n+1} to be same as that of e_n except that $e_{n+1}(u, a)$ will not be defined. The graph $R_{n+1} = (V, e_{n+1}, v_{n+1})$. We can have a reachability condition as the winning condition. i.e., the *Runner* wins iff he can reach a given vertex called the goal. The game ends, if either the *Runner* gets stuck or if the winning condition is satisfied.

It is easy to build a conventional game arena for the sabotage game where player positions alternate. The game arena will have as its local states subgraphs of R with the current position of *Runner* indicated. i.e., $W = \mathit{Edges} \times V$ where *Edges* is the set of all partial edge functions $e : V \times \Sigma \rightarrow V$. Let W^1 and W^2 be the set of game positions for *Runner* and *Blocker* respectively. The initial vertex $s_0 = (e_0, v_0)$ and $s_0 \in W^1$. Let $s = (e, v)$ and $s' = (e', v')$ be any two nodes in the arena. The trasition is defined as follows.

- if $s \in W^1$ and $e(v, a) = v'$ then $s \xrightarrow{a} s'$, $e = e'$ and $s' \in W^2$
- if $s \in W^2$, for some $u, u' \in W$ we have $e(u, a) = u'$ and e' is is same as e except that $e'(u, a)$ is not defined, then $s \xrightarrow{(u,a,u')} s'$, $v = v'$ and $s' \in W^1$.

Figure 2 shows the first three game positions in a possible run of the sabotage game. The game starts with *Runner* moving from node A to node B. The blocker then removes the edge a_2 adjacent to node B and the game continues. In the formulas given below, we will refer to *Runner* as player 1 and *Blocker* as player 2.

Since the *Runner*'s objective is to not get stuck, he might reason as follows. If it is the case that the *Blocker* always removes an edge adjacent to the node that *Runner* has currently selected then try to move to a node which has multiple outgoing edges. We use the following propositions:

- $\mathit{present}_v$: denotes that the current node of runner is v
- adj_m: denotes that the adjacent node has multiple edges

Let r_v denote the action which removes an adjacent edge of v and $move_{adj}$ denote the action which moves to the adjacent node with multiple edges. The *Runner*'s specification can be given as:

- $[present_v \mapsto r_v]^2 \Rightarrow [adj_m \mapsto move_{adj}]^1$

Consider the situation where all the nodes in the graph has a single outgoing edge and the goal state is a single state. It is quite easy to show that in such a game, the *Runner* wins iff the start node is the goal or if there is an edge connecting the start node with the goal. This property can be captured by the following proposition:

- g^B_{nice}: denotes that in the graph the start node is not the goal and there is no single edge between start and goal nodes. In other words the graph is "nice" for *Blocker*.

- adj^R_g: denotes that the *Runner*'s current node is one adjacent to the goal node.

Let r^g_{adj} denote the action which removes the edge connecting the current node of *Runner* with the goal. Consider the following formula:

- $[(g^B_{nice} \wedge adj^R_g) \mapsto r^g_{adj}]^2 \leadsto_2 (leaf \supset win)$

This says that if the graph is "nice" for *Blocker* and if the current selected node of *Runner* is one adjacent to the goal then remove the only edge connecting it with the goal. In all the other cases the *Blocker* can remove any random edge, and so this need not be mentioned in the strategy specification. This specification ensures that when the terminal node is reached then it is winning for *Blocker*.

5 Axiom system

We now present our axiomatization of the valid formulas of the logic. Before we present the axiomatization, we will find some abbreviations useful:

- $root = \neg\langle P \rangle True$ defines the root node to be one that has no predecessors.

- $\delta^\sigma_i(a) = \tau_i \wedge (\sigma)_i : a$ denotes that move "a" is enabled by σ at an i node.

- $inv^\sigma_i(a, \beta) = (\tau_i \wedge (\sigma)_i : a) \supset [a](\sigma \leadsto_i \beta)$ denotes the fact that after an "a" move by player i which conforms to σ, $\sigma \leadsto_i \beta$ continues to hold.

- $inv^\sigma_{\bar{i}}(\beta) = \tau_{\bar{i}} \supset [N](\sigma \leadsto_i \beta)$ says that after any move of $\bar{i}$, $\sigma \leadsto_i \beta$ continues to hold.

- $conf_\pi = \boxminus(\langle\overline{a}\rangle\tau_{\overline{i}} \supset \langle\overline{a}\rangle(\pi)_{\overline{i}} : a)$ denotes that all opponent moves in the past conform to π.

The axiom schemes

(A0) All the substitutional instances of the tautologies of propositional calculus.

(A1) (a) $[a](\alpha_1 \supset \alpha_2) \supset ([a]\alpha_1 \supset [a]\alpha_2)$
(b) $[\overline{a}](\alpha_1 \supset \alpha_2) \supset ([\overline{a}]\alpha_1 \supset [\overline{a}]\alpha_2)$

(A2) (a) $\langle a\rangle\alpha \supset [a]\alpha$
(b) $\langle\overline{a}\rangle\alpha \supset [\overline{a}]\alpha$
(c) $\langle\overline{a}\rangle \mathit{True} \supset \neg\langle\overline{b}\rangle \mathit{True}$ for all $b \neq a$

(A3) (a) $\alpha \supset [a]\langle\overline{a}\rangle\alpha$
(b) $\alpha \supset [\overline{a}]\langle a\rangle\alpha$

(A4) (a) $\diamondminus\, \mathit{root}$
(b) $\boxminus\alpha \equiv (\alpha \wedge [P]\boxminus\alpha)$

(A5) (a) $([\psi \mapsto a]^i)_i : a$ for all $a \in \Sigma$
(b) $\tau_i \wedge ([\psi \mapsto a]^i)_i : c \equiv \neg\psi$ for all $a \neq c$

(A6) (a) $(\sigma_1 + \sigma_2)_i : c \equiv \sigma_1 : c \vee \sigma_2 : c$
(b) $(\sigma_1 \cdot \sigma_2)_i : c \equiv \sigma_1 : c \wedge \sigma_2 : c$
(c) $(\pi \Rightarrow \sigma)_i : c \equiv conf_\pi \supset (\sigma)_i : c$

(A7) $\sigma \leadsto_i \beta \supset (\beta \wedge inv_i^\sigma(a,\beta) \wedge inv_{\overline{i}}^\sigma(\beta) \wedge (\neg leaf \supset enabled_\sigma))$

Inference rules

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

$$(Ind\text{-}past)\ \frac{\alpha \supset [P]\alpha}{\alpha \supset \boxminus\alpha}$$

$$(Ind \leadsto)\ \frac{\alpha \wedge \delta_i^\sigma(a) \supset [a]\alpha, \quad \alpha \wedge \tau_{\overline{i}} \supset [N]\alpha, \quad \alpha \wedge \neg leaf \supset enabled_\sigma, \quad \alpha \supset \beta}{\alpha \supset \sigma \leadsto_i \beta}$$

The axioms are mostly standard. After the Kripke axioms for the $\langle a\rangle$ modalities, we have axioms that ensure determinacy of both $\langle a\rangle$ and $\langle\overline{a}\rangle$ modalities, and an axiom to assert the uniqueness of the latter. We then have axioms that relate the previous and next modalities with each other, as well as to assert that the past modality steps through the $\langle \overline{a} \rangle$ modality. An axiom asserts the existence of the root in the past. The rest of the axioms describe the semantics of strategy specifications.

The rule *Ind-past* is standard, while $Ind \leadsto$ illustrates the new kind of reasoning in the logic. It says that to infer that the formula $\sigma \leadsto_i \beta$ holds in all reachable states, β must hold at the asserted state and

- for a player i node after every move which conforms to σ, β continues to hold.

- for a player $\overline{i}$ node after every enabled move, β continues to hold.

- player i does not get stuck by playing σ.

To see the soundness of (A7), suppose it is not valid. Then there exists a node s such that $M, s \models \sigma \leadsto_i \beta$ and one of the following holds:

- $M, s \not\models \beta$: In this case, from semantics we get that $M, s \not\models \sigma \leadsto_i \beta$ which is a contradiction.

- $M, s \not\models inv_i^\sigma(a, \beta)$: In this case, we have $s \in W^i$, $M, s \models (\sigma)_i : a$ and $M, s' \not\models \sigma \leadsto_i \beta$ where $s \stackrel{a}{\Longrightarrow} s'$. This implies that there is a path $\rho_{s'}^{s_k}$ which conforms to σ and either $M, s_k \not\models \beta$ or $moves(s_k) \cap \sigma(s_k) = \varnothing$. But since $s \stackrel{a}{\Longrightarrow} s'$, we have $\rho_s^{s_k}$ conforms to σ as well. From which it follows that $M, s \not\models \sigma \leadsto_i \beta$ which is a contradiction.

- $M, s \not\models inv_{\overline{i}}^\sigma(\beta)$: We have a similar argument as above.

- $M, s \not\models \neg leaf \supset enabled_\sigma$: This means that $M, s \models \neg leaf$ and $M, s \not\models enabled_\sigma$. Therefore $moves(s) \cap \sigma(s) = \varnothing$ and by semantics we have $M, s \not\models \sigma \leadsto_i \beta$ which is a contradiction.

To show that the induction rule preserves validity, suppose that the premise is valid and the conclusion is not. Then for some node s we have $M, s \models \alpha$ and $M, s \not\models \sigma \leadsto_i \beta$. i.e. there is a path $\rho_s^{s_k}$ which conforms to σ such that $M, s_k \not\models \beta$ or s_k is a non-leaf node and $\sigma(s_k) \cap moves(s_k) = \varnothing$. Let $\rho_s^{s_k}$ be the shortest of such paths.

Suppose $M, s_k \not\models \beta$, then we have the following two cases to consider.

- $s_{k-1} \in W^i$: By assumption on the path $\rho_s^{s_k}$, we have $M, s_{k-1} \models \alpha \wedge \delta_i^\sigma(a_{k-1})$. From validity of $\alpha \supset \beta$(the premise), we have $M, s_k \not\models \alpha$, which implies $M, s_{k-1} \not\models [a_{k-1}]\alpha$. Therefore we get $M, s_{k-1} \not\models (\alpha \wedge \delta_i^\sigma(a_{k-1})) \supset [a_{k-1}]\alpha$, which gives us a contradiction to the validity of a premise.

- $s_{k-1} \in W^{\bar{i}}$: By assumption on the path $\rho_s^{s_k}$, we have $M, s_{k-1} \models \alpha \wedge \tau_{\bar{i}}$. Using an argument similar to the previous case we also get $M, s_{k-1} \not\models [a_{k-1}]\alpha$. Therefore we have $M, s_{k-1} \not\models (\alpha \wedge \tau_{\bar{i}}) \supset [N]\alpha$, giving us a contradiction to the validity of a premise.

If s_k is a non-leaf node and $\sigma(s_k) \cap moves(s_k) = \varnothing$ then we have $M, s_k \models \alpha \wedge \neg leaf$ and $M, s_k \not\models enabled_\sigma$. Therefore $M, s_k \not\models (\alpha \wedge \neg leaf) \supset enabled_\sigma$, which is the required contradiction.

6 Completeness

To show completeness, we prove that every consistent formula is satisfiable. Let α_0 be a consistent formula, and let W denote the set of all maximal consistent sets (MCS). We use w, w' to range over MCS's. Since α_0 is consistent, there exists an MCS w_0 such that $\alpha_0 \in w_0$.

Define a transition relation on MCS's as follows: $w \xrightarrow{a} w'$ iff $\{\langle a\rangle\alpha | \alpha \in w'\} \subseteq w$. We will find it useful to work not only with MCS's, but also with sets of subformulas of α_0. For a formula α let $CL(\alpha)$ denote the subformula closure of α. In addition to the usual downward closure, we also require that $\diamondminus root$, $leaf \in CL(\alpha)$ and $\sigma \leadsto_i \beta \in CL(\alpha)$ implies that $\beta, inv_i^\sigma(a, \beta), inv_{\bar{i}}^\sigma(\beta), enabled_\sigma \in CL(\alpha)$. Let $\mathcal{AT}$ denote the set of all maximal consistent subsets of $CL(\alpha_0)$, reffered to as atoms. Each $t \in \mathcal{AT}$ is a finite set of formulas, we denote the conjunction of all formulas in t by $\widehat{t}$. For a nonempty subset $X \subseteq \mathcal{AT}$, we denote by $\widetilde{X}$ the disjunction of all $\widehat{t}, t \in X$. Define a transition relation on $\mathcal{AT}$ as follows: $t \xrightarrow{a} t'$ iff $\widehat{t} \wedge \langle a\rangle\widehat{t'}$ is consistent. Call an atom t a *root atom* if there does not exist any atom t' such that $t' \xrightarrow{a} t$ for some a. Note that $t_0 = w_0 \cap CL(\alpha_0) \in \mathcal{AT}$.

Proposition 6.1. There exist $t_1, \ldots, t_k \in \mathcal{AT}$ and $a_1, \ldots a_k \in \Sigma$ $(k \geq 0)$ such that $t_k \xrightarrow{a_k} t_{k-1} \ldots \xrightarrow{a_1} t_0$, where t_k is a root atom.

Proof. Consider the least set R containing t_0 and closed under the following condition: if $t_1 \in R$ and for some $a \in \Sigma$ there exists t_2 such that $t_2 \xrightarrow{a} t_1$, then $t_2 \in R$. Now, if there exists an atom $t' \in R$ such that t' is a root then we are done. Suppose not, then we have $\vdash \widetilde{R} \supset \neg root$. But then we can show that $\vdash \widetilde{R} \supset [P]\widetilde{R}$. By rule *Ind-past* and above we get $\vdash \widetilde{R} \supset \boxminus\neg root$. But then $t_0 \in R$ and hence $\vdash \widehat{t_0} \supset \widetilde{R}$ and therefore we get $\vdash \widehat{t_0} \supset \boxminus\neg root$, contradicting axiom (A4a). Q.E.D.

Above, we have additional properties: for any formula $\diamondminus\alpha \in t_k$, we also have $\alpha \in t_k$. Further, for all $j \in \{0, \cdots, k\}$, if $\diamondminus\alpha \in t_j$, then there exists i such that $k \geq i \geq j$ and $\alpha \in t_i$. Both these properties are ensured by axiom (A4b). A detailed proof can be found in appendix, lemma 8.2.

Hence it is easy to see that there exist MCS's $w_1, \ldots, w_k \in W$ and $a_1, \ldots a_k \in \Sigma$ $(k \geq 0)$ such that $w_k \xrightarrow{a_k} w_{k-1} \ldots \xrightarrow{a_1} w_0$, where $w_j \cap$

$CL(\alpha_0) = t_j$. Now this path defines a (finite) tree $T_0 = (S_0, \Longrightarrow_0, s_0)$ rooted at s_0, where $S_0 = \{s_0, s_1, \ldots, s_k\}$, and for all $j \in \{0, \cdots, k\}$, s_j is labelled by the MCS w_{k-j}. The relation $\Longrightarrow_0$ is defined in the obvious manner. From now we will simply say $\alpha \in s$ where s is the tree node, to mean that $\alpha \in w$ where w is the MCS associated with node s.

Inductively assume that we have a tree $T_k = (S_k, \Longrightarrow_k, s_0)$ such that the past formulas at every node have "witnesses" as above. Pick a node $s \in S_k$ such that $\langle a \rangle \mathit{True} \in s$ but there is no $s' \in S_k$ such that $s \overset{a}{\Longrightarrow} s'$. Now, if w is the MCS associated with node s, there exists an MCS w' such that $w \overset{a}{\longrightarrow} w'$. Pick a new node $s' \notin S_k$ and define $T_{k+1} = S_k \cup \{s'\}$ and $\Longrightarrow_{k+1} = \Longrightarrow_k \cup \{(s, a, s')\}$, where w' is the MCS associated with s'. It is easy to see that every node in T_{k+1} has witnesses for past formulas as well.

Now consider $T = (S, \Longrightarrow, s_0)$ defined by: $S = \bigcup\limits_{k \geq 0} S_k$ and $\Longrightarrow = \bigcup\limits_{k \geq 0} \Longrightarrow_k$. Define the model $M = (T, V)$ where $V(s) = w \cap P$, where w is the MCS associated with s.

Lemma 6.2. For any $s \in S$, we have the following properties.

1. if $[a]\alpha \in s$ and $s \overset{a}{\Longrightarrow} s'$ then $\alpha \in s'$.
2. if $\langle a \rangle \alpha \in s$ then there exists s' such that $s \overset{a}{\Longrightarrow} s'$ and $\alpha \in s'$.
3. if $[\overline{a}]\alpha \in s$ and $s' \overset{a}{\Longrightarrow} s$ then $\alpha \in s'$.
4. if $\langle \overline{a} \rangle \alpha \in s$ then there exists s' such that $s' \overset{a}{\Longrightarrow} s$ and $\alpha \in s'$.
5. if $\boxminus \alpha \in s$ and $s' \Longrightarrow^* s$ then $\alpha \in s'$.
6. if $\diamondminus \alpha \in s$ then there exists s' such that $s' \Longrightarrow^* s$ and $\alpha \in s'$.

Proof. Cases (1) to (5) can be shown using standard modal logic techniques. (6) follows from the existense of a root atom (proposition 6.1) and axiom (A4b). Q.E.D.

Lemma 6.3. For all $\psi \in \mathit{Past}(P)$, for all $s \in S$, $\psi \in s$ iff $\rho_s, s \models \psi$.

Proof. This follows from lemma 6.2 using an inductive argument. Q.E.D.

Lemma 6.4. For all i, for all $\sigma \in \mathit{Strat}^i(P^i)$, for all $c \in \Sigma$, for all $s \in S$, $(\sigma)_i : c \in s$ iff $c \in \sigma(s)$.

Proof. The proof is by induction on the structure of σ. The nontrivial cases are as follows:
$\sigma \equiv [\psi \mapsto a]$:
($\Rightarrow$) Suppose $([\psi \mapsto a]^i)_i : c \in s$. If $c = a$ then the claim holds trivially.

If $c \neq a$ then from (A5a) we get that $\neg\psi \in s$, from lemma 6.3 $\rho_s, s \not\models \psi$. Therefore by definition we have $[\psi \mapsto a]^i(s) = \Sigma$ and $c \in \sigma(w)$.
($\Leftarrow$) Conversly, suppose $([\psi \mapsto a]^i)_i : c \notin s$. From (A5a) we have $a \neq c$. From (A5b) we get $\psi \in s$. By lemma 6.3 $\rho_s, s \models \psi$. Therefore $c \notin \sigma(s)$ by definition.
$\sigma \equiv \pi \Rightarrow \sigma'$: Let $\rho^s_{s_0} : s_0 \overset{a_0}{\Longrightarrow} \cdots \overset{a_{k-1}}{\Longrightarrow} s_k = s$ be the unique path from the root to s.
($\Rightarrow$) Suppose $(\pi \Rightarrow \sigma')_i : c \in s$. To show $c \in (\pi \Rightarrow \sigma')(s)$. Suffices to show that $\rho^s_{s_0}$ conforms to π implies $c \in \sigma'(s)$. From (A6c) we have $\mathit{conf}_\pi \supset (\sigma')_i : c \in s$. Rewriting this we get $\diamondminus(\langle\overline{a}\rangle\tau_{\overline{i}} \wedge [\overline{a}](\neg(\pi)_{\overline{i}} : a)) \vee (\sigma')_i : c \in s$. We have two cases,

- if $(\sigma')_i : c \in s$ then by induction hypothesis we get $c \in \sigma'(s)$. Therefore by definition $c \in (\pi \Rightarrow \sigma)_i(s)$.

- otherwise we have $\diamondminus(\langle\overline{a}\rangle\tau_{\overline{i}} \wedge [\overline{a}](\neg(\pi)_{\overline{i}} : a)) \in s$. From lemma 6.2(6), there exits $s_l \in \rho_s$ such that $\langle\overline{a}\rangle\tau_{\overline{i}} \wedge [\overline{a}](\neg(\pi)_{\overline{i}} : a) \in s_l$. By lemma 6.2(4) there exists $s_{l-1} \in \rho_s \cap W^{\overline{i}}$ such that $s_{l-1} \overset{a}{\Longrightarrow} s_l$. From lemma 6.2(3), $\neg(\pi)_{\overline{i}} : a \in s_{l-1}$. Since s_{l-1} is an MCS, we have $(\pi)_{\overline{i}} : a \notin s_{l-1}$. By induction hypothesis, $a \notin \pi(s_{l-1})$, therefore we have $\rho^s_{s_0}$ does not conform to π.

($\Leftarrow$) Conversly, using (A6c) and a similar argument it can be shown that if $(\pi \Rightarrow \sigma')_i : c \notin s$ then $c \notin (\pi \Rightarrow \sigma')(s)$. Q.E.D.

Theorem 6.5. For all $\alpha \in \Pi$, for all $s \in S$, $\alpha \in s$ iff $M, s \models \alpha$.

Proof. The proof is by induction on the structure of α.
$\alpha \equiv (\sigma)_i : c$.
From lemma 6.4 we have $(\sigma)_i : c \in s$ iff $c \in \sigma(s)$ iff by semantics $M, s \models (\sigma)_i : c$.
$\alpha \equiv \sigma \rightsquigarrow_i \beta$.
($\Rightarrow$) We show the following:

1. If $\sigma \rightsquigarrow_i \beta \in s$ and there exits a transition $s \overset{a}{\Longrightarrow} s'$ such that $a \in \sigma(s)$, then $\{\beta, \sigma \rightsquigarrow_i \beta\} \subseteq s'$.
   Suppose $\sigma \rightsquigarrow_i \beta \in s$, from (A7) we have $\beta \in s$. We have two cases to consider.

   - $s \in W^i$: We have $\tau_i \in s$. Since $a \in \sigma(s)$, by lemma 6.4 we have $(\sigma)_i : a \in s$. From (A7) we get $[a](\sigma \rightsquigarrow_i \beta) \in s$. By lemma 6.2(1) we have $\sigma \rightsquigarrow_i \beta \in s'$.
   - $s \in W^{\overline{i}}$: We have $\tau_{\overline{i}} \in s$. From (A7) we get $[N](\sigma \rightsquigarrow_i \beta) \in s$, since s is an MCS we have for every $a \in \Sigma$, $[a](\sigma \rightsquigarrow_i \beta) \in s$. By lemma 6.2(1) we have $\sigma \rightsquigarrow_i \beta \in s'$.

By applying (A7) at s' we get $\beta \in s'$.

2. If $\sigma \rightsquigarrow_i \beta \in s$ and s is a non-leaf node, then $\exists s'$ such that $s \stackrel{a}{\Longrightarrow} s'$ and $a \in \sigma(s)$.
   Suppose s is a non-leaf node. From (A7), $\bigvee_{a \in \Sigma} (\langle a \rangle \mathit{True} \wedge (\sigma)_i : a) \in s$.
   Since s is an MCS, there exists an a such that $\langle a \rangle \mathit{True} \wedge (\sigma)_i : a \in s$. By lemma 6.2(2), there exists an s' such that $s \stackrel{a}{\Longrightarrow} s'$ and by lemma 6.4 $a \in \sigma(s)$.

(1) ensures that whenever $\sigma \rightsquigarrow_i \beta \in s$ and there exists a path $\rho_s^{s_k}$ which conforms to σ, then we have $\{\beta, \sigma \rightsquigarrow_i \beta\} \subseteq s_k$. Since $\beta \in \mathit{Past}(P)$, by lemma 6.3 we have $M, s_k \models \beta$. (2) ensures that for all paths $\rho_s^{s_k}$ which conforms to σ, if s_k is a non-leaf node, then $\mathit{moves}(s_k) \cap \sigma(s_k) \neq \varnothing$. Therefore we get $M, s \models \sigma \rightsquigarrow_i \beta$.
($\Leftarrow$) Conversly suppose $\sigma \rightsquigarrow_i \beta \notin s$, to show $M, s \not\models \sigma \rightsquigarrow_i \beta$. Suffices to show that there exists a path $\rho_s^{s_k}$ that conforms to σ such that $M, s_k \not\models \beta$ or s_k is a non-leaf node and $\mathit{moves}(s_k) \cap \sigma(s_k) = \varnothing$.

Lemma 6.6. For all $t \in \mathcal{AT}$, $\sigma \rightsquigarrow_i \beta \notin t$ implies there exists a path $\rho_t^{t_k} : t = t_1 \xrightarrow{a_1}_{\mathcal{AT}} t_2 \ldots \xrightarrow{a_{k-1}}_{\mathcal{AT}} t_k$ which conforms to σ such that one of the following conditions hold.

- $\beta \notin t_k$.
- t_k is a non-leaf node and $\mathit{moves}(t_k) \cap \sigma(t_k) = \varnothing$.

We have $t = s \cap CL(\sigma \rightsquigarrow_i \beta)$ is an atom. By lemma 6.6(proof given in appendix), there exists a path in the atom graph $t = t_1 \xrightarrow{a_1}_{\mathcal{AT}} t_2 \ldots \xrightarrow{a_k}_{\mathcal{AT}} t_k$ such that $\beta \notin t_k$ or t_k is a non-leaf node and $\mathit{moves}(t_k) \cap \sigma(t_k) = \varnothing$. t_1 can be extended to the MCS s. Let $t_2' = t_2 \cup \{\alpha | [a_1]\alpha \in s\}$. Its easy to check that t_2' is consistent. Consider any MCS s_2 extending t_2', we have $s \stackrel{a_1}{\Longrightarrow} s_2$. Continuing in this manner we get a path in $s = s_1 \stackrel{a_1}{\Longrightarrow} s_2 \ldots \stackrel{a_{k-1}}{\Longrightarrow} s_k$ in M which conforms to σ where either $\beta \notin s_k$ or s_k is a non-leaf node and $\mathit{moves}(s_k) \cap \sigma(s_k) = \varnothing$. Q.E.D.

7 Extensions for strategy specification

Until operator:

One of the natural extensions to strategy specification is to come up with a construct which asserts that a player strategy conforms to some specification σ until a certain condition holds. Once the condition is fulfilled, he is free to choose any action.

We can add the future modality $\Diamond\alpha$ in the logic defined in section 3 with the following interpretation.

- $M, s \models \Diamond\gamma$ iff there exists an s' such that $s \Longrightarrow^* s'$ and $M, s' \models \gamma$.

Let *Past*(Π) and *Future*(Π) denote the past and future fragment of Π respectively. i.e.

$$\mathit{Past}(\Pi^{P^i}) := p \in P^i \mid \neg\alpha \mid \alpha_1 \vee \alpha_2 \mid \diamondminus\alpha$$
$$\mathit{Future}(\Pi^{P^i}) := p \in P^i \mid \neg\alpha \mid \alpha_1 \vee \alpha_2 \mid \Diamond\alpha$$

Let $\Box\alpha = \neg\Diamond\neg\alpha$ and $\boxminus\alpha = \neg\diamondminus\neg\alpha$. We can enrich $\mathit{Strat}^i(P^i)$ with the until operator $\sigma\mathbf{U}\varphi$, where $\varphi \in \mathit{Past}(\Pi^{P^i}) \cup \mathit{Future}(\Pi^{P^i})$, with the following interpretation:

- $(\sigma\mathbf{U}\varphi)(s) = \begin{cases} \Sigma & \text{if } \exists j : 0 \leq j \leq m \text{ such that } \rho_{s_0}^{s_j}, j \models \varphi \\ \sigma(s) & \text{otherwise} \end{cases}$

Note that until does not guarantee that φ will eventually hold. We can extend the axiomatization quite easily to handle the new construct. Firstly we need to add the following axiom and the derivation rule for the future modality.

(*Ax-box*) $\Box\alpha \equiv (\alpha \wedge [N]\Box\alpha)$

(*Ind*) $\dfrac{\alpha \supset [N]\alpha}{\alpha \supset \Box\alpha}$

Using the above axiom and inference rule one can easily show the analogue of lemma 6.2 and lemma 6.3 for the future modality. For the until operator we have the following axiom.

(*Ax-Until*) $(\sigma\mathbf{U}\varphi)_i : c \equiv \neg\diamondminus\varphi \supset (\sigma)_i : c$

We can show that lemma 6.4 holds once again, for the extended syntax.

Lemma 7.1. For all i, for all $\sigma \in \mathit{Strat}^i(P^i)$, for all $c \in \Sigma$, for all $s \in S$, $(\sigma)_i : c \in s$ iff $c \in \sigma(s)$.

Proof. The proof is by induction on the structure of σ as seen before. The interesting case is when $\sigma \equiv \sigma'\mathbf{U}\varphi$:
($\Rightarrow$) Suppose $(\sigma'\mathbf{U}\varphi)_i : c \in s$. It suffices to show that $\forall j : 0 \leq j \leq k$, $\rho_{s_0}^{s_j}, j \not\models \varphi$ implies $c \in \sigma'(s)$. From axiom (*Ax-Until*), we have $\neg\diamondminus\varphi \supset (\sigma')_i : c \in s$. Rewriting this, we get $\diamondminus\varphi \in s$ or $(\sigma')_i : c \in s$.

- if $\diamondminus\varphi \in s$, then by lemma 6.2, $\exists j : 0 \leq j \leq k$ such that $\varphi \in s_j$. Therefore we have $\rho_{s_0}^{s_j} \models \varphi$.

- if $(\sigma)_i : c \in s$, then by induction hypothesis we have $c \in \sigma(s)$.

($\Leftarrow$) To show $(\sigma'\mathbf{U}\varphi)_i : c \notin s$ implies $c \notin (\sigma'\mathbf{U}\varphi)(s)$. It suffices to show that $\forall j : 0 \leq j \leq m$, $\rho_{s_0}^{s_j}, j \not\models \varphi$ and $c \notin \sigma'(s)$. From axiom (*Ax-Until*), we have $\neg\diamondminus\varphi \wedge \neg((\sigma')_i : c) \in s$. Rewriting this we get $\boxminus\neg\varphi \in s$ and $\neg((\sigma)_i : c) \in s$.

- $\boxminus\neg\varphi \in s$ implies $\forall j : 0 \le j \le m$, $\neg\varphi \in s_j$(by lemma 6.2). Since s_j is an MCS, $\alpha \notin s_j$. Therefore we have $\forall j : 0 \le j \le m$, $\rho_{s_0}^{s_j}, j \not\models \varphi$.
- $\neg((\sigma)_i : c) \in s$ implies $(\sigma)_i : c \notin s$(Since s is an MCS). By induction hypothesis we have $c \notin \sigma(s)$.

Q.E.D.

Nested strategy specification:

Instead of considering simple past time formulas as conditions to be verified before deciding on a move, we can enrich the structure to assert the opponents conformance to some strategy specification in the history of the play. This can be achieved by allowing nesting of strategy specification. We can extend the strategy specification syntax to include nesting as follows.

$$\Gamma^i := \psi \mid \sigma \mid \gamma_1 \wedge \gamma_2$$
$$Strat^i_{rec}(P^i) := [\gamma \mapsto a]^i \mid \sigma_1 + \sigma_2 \mid \sigma_1 \cdot \sigma_2$$

where $\psi \in Past(P^i)$, $\sigma \in Strat^i_{rec}(P^i)$ and $\gamma \in \Gamma^{\bar{i}}$. Below we give the semantics for the part that requires change. For the game tree $\mathcal{T}$ and a node s in it, let $\rho_{s_0}^s : s_0 \xrightarrow{a_1} s_1 \cdots \xrightarrow{a_m} s_m = s$ denote the unique path from s_0 to s

- $[\gamma \mapsto a]^i(s) = \begin{cases} a & \text{if } \ s \in W^i \text{ and } \rho_{s_0}^s, m \models \gamma \\ \Sigma & \text{otherwise} \end{cases}$
- $\rho_{s_0}^s, m \models \sigma$ iff $\forall j : 0 \le j < m$, $a_i \in \sigma(s_i)$.
- $\rho_{s_0}^s, m \models \gamma_1 \wedge \gamma_2$ iff $\rho_{s_0}^s, m \models \gamma_1$ and $\rho_{s_0}^s, m \models \gamma_2$.

For a past formula ψ, the notion of $\rho_{s_0}^s, m \models \psi$ is already defined in section 3. Let L denote the logic introduced in section 3 and L_{rec} be same as L except that $\sigma \in Strat^i_{rec}(P^i)$. We will show that L and L_{rec} have equivalent expressive power. Therefore one can stick to the relatively simple strategy specification syntax given in section 3 rather than taking into account explicit nesting.

It is easy to see that any formula $\gamma \in \Gamma^i$, can be rewritten in the form $\sigma' \wedge \psi$ where $\sigma' \in Strat^i_{rec}(P^i)$ and $\psi \in Past(P^i)$. This is due to the fact that if $\psi_1, \psi_2 \in Past(P^i)$ then $\psi_1 \wedge \psi_2 \in Past(P^i)$ and $\sigma_1 \wedge \sigma_2 \equiv \sigma_1 \cdot \sigma_2$(formally $\forall s, \rho_{s_0}^s, m \models \sigma_1 \wedge \sigma_2$ iff $\rho_{s_0}^s, m \models \sigma_1 \cdot \sigma_2$).

Given $\sigma_{rec} \in Strat^i_{rec}(P^i)$ the equivalent formula $\sigma \in Strat^i(P^i)$ is constructed inductively as follows.

$$\begin{aligned}
[\![[\psi \mapsto a]]\!] &= [\psi \mapsto a] \\
[\![\sigma_1 + \sigma_2]\!] &= [\![\sigma_1]\!] + [\![\sigma_2]\!] \\
[\![\sigma_1 \cdot \sigma_2]\!] &= [\![\sigma_1]\!] \cdot [\![\sigma_2]\!] \\
[\![[\sigma \mapsto a]]\!] &= [\![\sigma]\!] \Rightarrow [True \mapsto a] \\
[\![[\sigma \wedge \psi \mapsto a]]\!] &= [\![\sigma]\!] \Rightarrow [\psi \mapsto a]
\end{aligned}$$

Lemma 7.2. For all i, for all $s \in S$, for all $\sigma \in Strat^i(P^i)$, $\sigma(s) = [\![\sigma]\!](s)$.

Proof. The proof is by induction on the structure of σ. Let $s \in S$ and $\rho^s_{s_0} : s_0 \xrightarrow{a_1} s_1 \cdots \xrightarrow{a_m} s_m = s$ be the unique path from root to s.
$\sigma \equiv [\psi \mapsto a]$: Follows from the definition.
$\sigma \equiv \sigma_1 \cdot \sigma_2$ and $\sigma \equiv \sigma_1 + \sigma_2$ follows easily by applying induction hypothesis.
$\sigma \equiv [\pi \mapsto a]$: We need to show that for all s, $[\pi \mapsto a](s) = ([\![\pi]\!] \Rightarrow [True \mapsto a])(s)$. We have the following two cases:

- $\rho^s_{s_0}, m \models \pi$: In this case, we have $[\pi \mapsto a](s) = a$. $\rho^s_{s_0}, m \models \pi$ implies $\forall j : 0 \leq j < m$, $a_j \in \pi(s_j)$. From induction hypothesis, $a_j \in [\![\pi]\!](s_j)$, which implies $\rho^s_{s_0}$ conforms to $[\![\pi]\!]$. From the semantics, we get $([\![\pi]\!] \Rightarrow [True \mapsto a])(s) = ([True \mapsto a])(s) = a$.

- $\rho^s_{s_0}, m \not\models \pi$: In this case, we have $[\pi \mapsto a](s) = \Sigma$ and $\exists j : 0 \leq j < m$ such that $a_j \notin \pi(s_j)$. By induction hypothesis, we have $a_j \notin [\![\pi]\!](s_j)$ which implies that $\rho^s_{s_0}$ does not conform to $[\![\pi]\!]$. From semantics we get that $([\![\pi]\!] \Rightarrow [True \mapsto a])(s) = \Sigma$.

$\sigma \equiv [\pi \wedge \psi \mapsto a]$: The following two cases arise:

- $\rho^s_{s_0}, m \models \pi \wedge \psi$: We have $[\pi \wedge \psi \mapsto a](s) = a$. $\rho^s_{s_0}, m \models \pi \wedge \psi$ implies $\rho^s_{s_0}, m \models \pi$ and $\rho^s_{s_0}, m \models \psi$. $\rho^s_{s_0}, m \models \pi$ implies $\forall j : 0 \leq j < m$, $a_j \in \pi(s_j)$. By induction hypothesis, $a_j \in [\![\pi]\!](s_j)$ and as before we get $([\![\pi]\!] \Rightarrow [\psi \mapsto a])(s) = ([\psi \mapsto a])(s) = a$.

- $\rho^s_{s_0}, m \not\models \pi \wedge \psi$: We have the following two cases
  - $\rho^s_{s_0}, m \not\models \psi$: It is easy to see that $[\pi \wedge \psi \mapsto a](s) = ([\![\pi]\!] \Rightarrow [\psi \mapsto a])(s) = \Sigma$.
  - $\rho^s_{s_0}, m \not\models \pi$: In this case, $\exists j : 0 \leq j < m$ such that $a_j \notin \pi(s_j)$. By induction hypothesis, we have $a_j \notin [\![\pi]\!](s_j)$. By an argument similar to the one above we get $[\![\pi]\!] \Rightarrow [\psi \mapsto a])(s) = \Sigma$.

Q.E.D.

For the converse, given a $\sigma \in Strat^i(P^i)$, we can construct an equivalent formula $\sigma_{rec} \in Strat^i_{rec}(P^i)$. The crucial observation is the following equivalences in $Strat^i(P^i)$.

- $\pi \Rightarrow \sigma_1 + \sigma_2 \equiv (\pi \Rightarrow \sigma_1) + (\pi \Rightarrow \sigma_2)$
- $\pi \Rightarrow \sigma_1 \cdot \sigma_2 \equiv (\pi \Rightarrow \sigma_1) \cdot (\pi \Rightarrow \sigma_2)$
- $\pi_1 \Rightarrow (\pi_2 \Rightarrow \sigma) \equiv (\pi_1 \cdot \pi_2) \Rightarrow \sigma$

Using the above equivalences, we can write the strategy specification σ in a normal form where all the implications are of the form $\pi \Rightarrow [\psi \mapsto a]$. Then σ_{rec} is constructed inductively as follows:

$$\begin{array}{rcl} [\![[\psi \mapsto a]]\!] & = & [\psi \mapsto a] \\ [\![\sigma_1 + \sigma_2]\!] & = & [\![\sigma_1]\!] + [\![\sigma_2]\!] \\ [\![\sigma_1 \cdot \sigma_2]\!] & = & [\![\sigma_1]\!] \cdot [\![\sigma_2]\!] \\ [\![\pi \Rightarrow [\psi \mapsto a]]\!] & = & [[\![\pi]\!] \wedge \psi \mapsto a] \end{array}$$

Lemma 7.3. For all i, for all $s \in S$, for all $\sigma \in Strat^i(P^i)$, $\sigma(s) = [\![\sigma]\!](s)$.

Proof. The proof is by induction on the structure of formula. Let $\rho_{s_0}^s : s_0 \stackrel{a_1}{\Longrightarrow} s_1 \cdots \stackrel{a_m}{\Longrightarrow} s_m = s$. The interesting case is when $\sigma \equiv \pi \Rightarrow [\psi \mapsto a]$. We need to show that for all s, $\pi \Rightarrow [\psi \mapsto a](s) = [\pi \wedge \psi \mapsto a](s)$. We have the following two cases:

- $\rho_{s_0}^s$ conform to π: We have $\pi \Rightarrow [\psi \mapsto a](s) = [\psi \mapsto a](s)$ and $\forall j : 0 \leq j < m$, $a_j \in \pi(s_i)$. By induction hypothesis, $a_j \in [\![\pi]\!](s_i)$ which implies that $\rho_{s_0}^s, m \models \pi$. Therefore $[[\![\pi]\!] \wedge \psi \mapsto a](s) = [\psi \mapsto a](s)$.
- $\rho_{s_0}^s$ does not conform to π: By an argument similar to the above, we can show that $\pi \Rightarrow [\psi \mapsto a](s) = [\pi \wedge \psi \mapsto a](s) = \Sigma$.

Q.E.D.

Theorem 7.4. Logics L and L_{rec} have equivalent expressive power. i.e.

- For every $\alpha \in \Pi$, there exists $\alpha_{rec} \in \Pi_{rec}$ such that $M, s \models \alpha$ iff $M, s \models \alpha_{rec}$.
- For every $\alpha_{rec} \in \Pi_{rec}$ there exists $\alpha \in \Pi$ such that $M, s \models \alpha_{rec}$ iff $M, s \models \alpha$.

Proof. The theorem follows from lemma 7.2 and lemma 7.3 by a routine inductive argument. Q.E.D.

8 Discussion

We have defined a logic for reasoning about composite strategies in games. We have presented an axiomatization for the logic and shown its completeness.

We again remark that the presentation has been given for two-player games only for easy readability. It can be checked that all the definitions and arguments given here can be appropriately generalized for n-player games.

While our emphasis in the paper has been on advocating syntactically constructed strategies, we make no claims to having the “right” set of connectives for building them. This will have to be decided by experience, gained by specifying several kinds of strategies which turn out to be of use in reasoning about games.

We believe that a framework of this sort will prove useful in reasoning about multi-stage and repeated games, where strategy revision based on learning other players' strategies (perhaps partially) plays an important role.

Appendix

Lemma 8.1. For atoms t_1 and t_2, the following statements are equivalent.

1. $\widehat{t_1} \wedge \langle a\rangle \widehat{t_2}$ is consistent.

2. $\langle \overline{a}\rangle \widehat{t_1} \wedge \widehat{t_2}$ is consistent.

Proof. Suppose $\langle \overline{a}\rangle \widehat{t_1} \wedge \widehat{t_2}$ is consistent, from (A3b) we have $\langle \overline{a}\rangle \widehat{t_1} \wedge [\overline{a}]\langle a\rangle \widehat{t_2}$ is consistent. Therefore, $\langle \overline{a}\rangle(\widehat{t_1} \wedge \langle a\rangle \widehat{t_2})$ is consistent, which implies $\not\vdash [\overline{a}]\neg(\widehat{t_1} \wedge \langle a\rangle \widehat{t_2})$. From ($NG$-), $\not\vdash \neg(\widehat{t_1} \wedge \langle a\rangle \widehat{t_2})$, thus we have that $\widehat{t_1} \wedge \langle a\rangle \widehat{t_2}$ is consistent.

Suppose $\widehat{t_1} \wedge \langle a\rangle \widehat{t_2}$ is consistent, from (A3a) we have $[a]\langle \overline{a}\rangle \widehat{t_1} \wedge \langle a\rangle \widehat{t_2}$ is consistent. Therefore, $\langle a\rangle(\langle \overline{a}\rangle \widehat{t_1} \wedge \widehat{t_2})$ is consistent, which implies $\not\vdash [a]\neg(\langle \overline{a}\rangle \widehat{t_1} \wedge \widehat{t_2})$. From ($NG$-), $\not\vdash \neg(\langle \overline{a}\rangle \widehat{t_1} \wedge \widehat{t_2})$, thus we get that $\langle \overline{a}\rangle \widehat{t_1} \wedge \widehat{t_2}$ is consistent. Q.E.D.

Lemma 8.2. Consider the path $t_k \xrightarrow{a_k} t_{k-1} \dots \xrightarrow{a_1} t_0$ where t_k is a root atom,

1. For all $j \in \{0, \dots, k-1\}$, if $[\overline{a}]\alpha \in t_j$ and $t_{j+1} \xrightarrow{a} t_j$ then $\alpha \in t_{j+1}$.

2. For all $j \in \{0, \dots, k-1\}$, if $\langle \overline{a}\rangle\alpha \in t_j$ and $t_{j+1} \xrightarrow{b} t_j$ then $b = a$ and $\alpha \in t_{j+1}$.

3. For all $j \in \{0, \dots, k-1\}$, if $\diamondminus\alpha \in t_j$ then there exists $i : j \le i \le k$ such that $\alpha \in t_i$.

Proof. (1) Since $t_{j+1} \xrightarrow{a} t_j$, we have $\widehat{t_{j+1}} \wedge \langle a\rangle \widehat{t_j}$ is consistent, By lemma 8.1, $\widehat{t_j} \wedge \langle \overline{a}\rangle \widehat{t_{j+1}}$ is consistent, which implies $[\overline{a}]\alpha \wedge \langle \overline{a}\rangle \widehat{t_{j+1}}$ is consistent (by omitting some conjuncts). Therefore $\langle \overline{a}\rangle(\alpha \wedge \widehat{t_{j+1}})$ is consistent. Using (NG-) we get $\alpha \wedge \widehat{t_{j+1}}$ is consistent and since t_{j+1} is an atom, we have $\alpha \in t_{j+1}$.
(2) Suppose $t_{j+1} \xrightarrow{b} t_j$, we first show that $b = a$. Suppose this is not true, since $t_{j+1} \xrightarrow{b} t_j$, we have $\widehat{t_j} \wedge \langle \overline{b}\rangle \widehat{t_{j+1}}$ is consistent. And therefore $\widehat{t_j} \wedge \langle \overline{b}\rangle\, True$ is consistent. From axiom (A2c) $\widehat{t_j} \wedge [\overline{a}]False$ is consistent. If $\langle \overline{a}\rangle\alpha \in t_j$, then we get $\langle \overline{a}\rangle\alpha \wedge [\overline{a}]False$ is consistent. Therefore $\langle \overline{a}\rangle(\alpha \wedge False)$ is consistent. From (NG-) we have $\alpha \wedge False$ is consistent, which is a contradiction.

To show $\alpha \in t_{j+1}$ observe that $\langle\overline{a}\rangle\alpha \in t_j$ implies $[\overline{a}]\alpha \in t_j$ (by axiom (A2b) and closure condition). By previous argument we get $\alpha \in t_{j+1}$.
(3) Suppose $\diamondminus\alpha \in t_j$ and $t_{j+1} \xrightarrow{a} t_j$. If $\alpha \in t_j$ then we are done. Else by axiom (A4b) and the previous argument, we have $\langle\overline{a}\rangle\diamondminus\alpha \in t_j$. From (2) we have $\diamondminus\alpha \in t_{j+1}$. Continuing in this manner, we either get an i where $\alpha \in t_i$ or we get $\diamondminus\alpha \in t_k$. Since t_k is the root, this will give us $\alpha \in t_k$.

Q.E.D.

Lemma 6.6: For all $t \in \mathcal{AT}$, $\sigma \leadsto_i \beta \notin t$ implies there exists a path $\rho_t^{t_k} : t = t_1 \xrightarrow{a_1}_{\mathcal{AT}} t_2 \ldots \xrightarrow{a_{k-1}}_{\mathcal{AT}} t_k$ which conforms to σ such that one of the following conditions hold.

- $\beta \notin t_k$.
- t_k is a non-leaf node and $moves(t_k) \cap \sigma(t_k) = \varnothing$.

Proof. Consider the least set R containing t and closed under the following condition:

- if $t_1 \in R$ then for every transition $t_1 \xrightarrow{a} t_2$ such that $a \in \sigma(t_1)$ we have $t_2 \in R$.

If there exists an atom $t' \in R$ such that $\beta \notin t'$ or if t' is a non-leaf node and $moves(t') \cap \sigma(t') = \varnothing$, then we are done. Suppose not, then we have $\vdash \widetilde{R} \supset \beta$ and $\vdash (\widetilde{R} \wedge \neg leaf) \supset \bigvee_{a\in\Sigma} (\langle a\rangle True \wedge (\sigma)_i : a)$.

Claim 8.3. The following are derivable.

1. $\vdash (\widetilde{R} \wedge \tau_i \wedge (\sigma)_i : a) \supset [a]\widetilde{R}$.
2. $\vdash (\tau_{\overline{i}} \wedge \widetilde{R}) \supset [N]\widetilde{R}$.

Assume claim 8.3 holds, then applying (IND) rule we get $\vdash \widetilde{R} \supset \sigma \leadsto_i \beta$. But $t \in R$ and therefore $\vdash \widehat{t} \supset \sigma \leadsto_i \beta$, contradicting the assumption that $\sigma \leadsto_i \beta \notin t$.

Q.E.D.

Proof of claim 8.3: To prove 1, suppose the claim does not hold. We have that $(\widetilde{R} \wedge \tau_i \wedge (\sigma)_i : a) \wedge \langle a\rangle\neg\widetilde{R}$ is consistent. Let $R' = \mathcal{AT} - R$. If $R' = \varnothing$ then $R = \mathcal{AT}$ in which case its easy to see that the claim holds. If $R' \neq \varnothing$, then we have $(\widetilde{R} \wedge \tau_i \wedge (\sigma)_i : a) \wedge \langle a\rangle\widetilde{R'}$ is consistent. Hence for some $t_1 \in R$ and $t_2 \in R'$, we have $(\widehat{t_1} \wedge \tau_i \wedge (\sigma)_i : a)) \wedge \langle a\rangle\widehat{t_2}$ is consistent. Which implies $t_1 \xrightarrow{a}_{\mathcal{AT}} t_2$ and this transition conforms to σ. By closure condition on R, $t_2 \in R$ which gives us the required contradiction.

Proof of 2 is similar.

References

[AHK02] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49:672–713, 2002.

[Ben02] Johan Van Benthem. An essay on sabotage and obstruction. In D.Hutter and S.Werner, editors, *Festschrift in honour of Prof. Joerg Siekmann*, Lecture Notes in Artificial Intelligence. Springer-Verlag, 2002.

[Bon91] G. Bonanno. The logic of rational play in games of perfect information. *Economics and Philosophy*, 7:37–65, 1991.

[Ehr61] Andrzej Ehrenfeucht. An application of games to the completeness problem for formalized theories. *Fundamenta Mathematicae*, 49:129–141, 1961.

[Gor01] V. Goranko. Coalition games and alternating temporal logics. *Proceedings of 8th conference on Theoretical Aspects of Rationality and Knowledge (TARK VIII)*, pages 259–272, 2001.

[HKT00] David Harel, Dexter Kozen, and Jerzy Tiuryn. *Dynamic Logic*. The MIT Press, October 2000.

[HvdHMW03] Paul Harrenstein, Wiebe van der Hoek, John-Jules Meyer, and Cees Witteven. A modal characterization of Nash equilibrium. *Fundamenta Informaticae*, 57:2–4:281–321, 2003.

[Lan02] Martin Lange. *Games for modal and temporal logics*. PhD thesis, University of Edinburgh, 2002.

[Par85] Rohit Parikh. The logic of games and its applications. *Annals of Discrete Mathematics*, 24:111–140, 1985.

[Pau01] Marc Pauly. *Logic for Social Software*. PhD thesis, University of Amsterdam, October 2001.

[vB01] Johan van Benthem. Games in dynamic epistemic logic. *Bulletin of Economic Research*, 53(4):219–248, 2001.

[vB03] Johan van Benthem. Logic games are complete for game logics. *Studia Logica*, 75:183–203, 2003.

[vD00] Hans P. van Ditmarsch. *University of Amsterdam*. PhD thesis, University of Amsterdam, November 2000.

[vdHJW05] Wiebe van der Hoek, Wojtek Jamroga, and Michael Wooldridge. A logic for strategic reasoning. *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 05)*, pages 157–164, 2005.