Stability under Strategy Switching

Soumya Paul, R. Ramanujam and Sunil Simon

The Institute of Mathematical Sciences
C.I.T. Campus, Chennai 600 113, India.
e-mail: {soumya,jam,sunils}@imsc.res.in

Abstract. We suggest that a *process-like* notion of strategy is relevant in the context of interactions in systems of self-interested agents. In this view, strategies are not plans formulated by rational agents considering all possible futures and (mutually recursively) taking into account strategies employed by other players. Instead, they are partial; players start with a set of potential strategies and dynamically switch between them. This necessitates some means in the model for players to access each others' strategies, and we suggest a syntax by which players' rationale for such switching may be specified and structurally composed. In such a model one can ask a stability question: given a game arena and a strategy specification, whether players eventually settle down to strategies without further switching. We show that this problem can be algorithmically solved using automata theoretic methods.

Keywords: Graphical games, strategy specifications, strategy switching.

1 Overview

Consider the game of cricket[1]. A bowler, starting on his run-up, considers: Should I bowl on his off-side or leg-side? Should I bowl a short-pitch ball? Should I bowl a slower one? Since he mis-hit the last bouncer I bowled to him, should I bowl one again? The batsman, on his part, considers as he takes his stance: If he bowls on my legs, should I pelt him for a boundary and reveal my strength off that flank? Or should I play it safe and settle for a single? I have already hit two boundaries in this over; if I hit him for too many runs, will he be taken off the attack?

In an ideal world, both bowler and batsman would have perfect information not only about each other's prowess but also about the nature of the pitch, and would play optimal mixed strategies, since they could go through all the reasoning above before a single ball is ever bowled. We could compute equilibria and predict rational cricket play.

Not only is the actual game far from ideal, it is also more interesting. If we are interested in predicting, in addition to outcomes, also how the play is likely

[1] Wikipedia-level understanding of cricket http://en.wikipedia.org/wiki/Cricket is enough to understand the points being made here, though some knowledge of cricket would surely help.

to progress (at some partial play), we need to correspondingly look not just at *which* strategies are available to players, but also how they *select* a strategy from among many. Such considerations naturally lead to partial strategies, and the notion of *switching* between (partial) strategies.

In such a view, a player enters the game arena with information on the game structure and on other players' skills, as well as an initial set of possible strategies to employ. As the play progresses, she makes observations and accordingly revises strategies, switches from one to another, perhaps even devises new strategies that she hadn't considered before. The dynamics of such interaction eventually leads to some strategies being eliminated, and some becoming stable.

Such considerations can be entirely eliminated by taking into account all possible futures while strategizing. However, such omniscient strategizing may be impossible, even in principle, for finitary agents (who have access only to finite resources). Dynamical system models of social interaction and negotiations have for long considered such switching behaviour ([7], [5]), and we suggest that such a consideration is relevant for computational models as well.

What questions can one study in such a model? Since the model describes dynamics, it is best suited to address questions that relate to eventual patterns in game evolution dictated by the dynamics in the model. For instance, since some strategies may simply get eliminated in the course of play, eventual game evolution may get restricted, and one can ask: "Does the play finally settle down to some subset of the entire arena?", "Can a player ensure certain objectives using a strategy which does not involve switching between a set of strategies?"

Another interesting question is, "Given a sub-arena of the game, is the strategy of a player *live* in that subset?" A strategy is live if for every history in the subset, the action it specifies is present in the set. Such questions are especially relevant in the context of bargaining and negotiations, as evidenced in many political contexts.

In this work, we look at algorithmic issues concerning the above questions. We give a simple but expressive syntax for specifying and composing strategies. We then show that in the case of bounded memory strategies, these questions can be algorithmically solved. At the heart of these questions lie the issue of *liveness* of a strategy which we formalise in the next section. However what is emphasized is the need and possibility of computational models that include process aspects of strategizing and application of algorithmic tools on them, rather than a study of the complexity of determining stability under switching.

Related Work

Dynamic learning has been extensively studied in game theory: for instance, Young ([15],[16]) considers a model in which each player chooses an optimal strategy based on a sample of information about what other players have done in the past. Similar analyses have been carried out in the context of cooperative game theory as well: here players decide dynamically which coalition to join. One asks how coalition structures change over time, and which coalition players will eventually arrive at ([2]). Evolutionary game theory ([14]) studies how players observe payoffs of other players in their neighbourhood and accordingly change strategies to maximise fitness.

Our work is located in the logical foundations of game theory, and hence employs logical descriptions of strategies and algorithms to answer questions. Modal logics have been used in various ways to reason about games. Notable among these is the work on alternating temporal logic (ATL) [1], a logic where assertions are made on outcomes a coalition of players can ensure. Various extensions of ATL ([10],[11]) has been proposed to incorporate knowledge of players and strategies explicitly into the logic. In [8, 9] van Benthem uses dynamic logic to describe games as well as strategies. [4] presents a complete axiomatisation of a logic describing both games and strategies in a dynamic logic framework where assertions are made about atomic strategies. [6] studies a logic in which not only are games structured, but so also are strategies.

Somewhat different in approach, and yet closely related is the work of De Vos and Vermeir ([12],[13]) in which the authors present a framework for decision making with circumstance dependent preferences and decisions (OCLP). It allows decisions that comprise of multiple alternatives which become available only when a choice between them is forced.

Due to space restrictions, detailed proofs have been omitted[2].

2 Preliminaries

We are interested in looking at infinite duration games. We first introduce extensive form games which constitutes our game models.

2.1 Extensive Form Games

Let $N = \{1, \ldots, n\}$ be the set of players. For each $i \in N$, let A_i be a finite set of actions, which represent the moves of the players. We assume that the action sets of the players are mutually disjoint, i.e., $A_i \cap A_j = \emptyset$ for $i \neq j$. Let $A = A_1 \times \cdots \times A_n$ denote the set of action tuples and $\widetilde{A} = A_1 \cup \cdots \cup A_n$ denote the set of actions of all the players. For any action tuple $\bar{a} = (a_1, \ldots, a_n) \in A$, we write $a \in \bar{a}$ if $a = a_i$ for some $1 \leq i \leq n$.

An extensive form game is a tree $\mathcal{T} = (T, \Rightarrow, t_0)$ where $T \subseteq A^*$ is a prefix-closed set called the set of nodes or game positions. The initial game position or the root of $\mathcal{T}$ is $t_0 = \epsilon$ (the empty word) and the edge relation is $\Rightarrow \subseteq T \times T$. A play in the game is just a path in $\mathcal{T}$ starting at t_0. For technical convenience we assume that all plays are infinite, i.e. for all $t \in T$, $\exists t'$ such that $t \Rightarrow t'$.

Strictly speaking, a game consists of a game tree along with *winning conditions* for the players. As we shall see later the winning conditions in our case will be some properties of the game model which are fairly general. Assuming that the outcomes and payoffs of the game arise from a fixed finite set, they can be coded up using propositions in our logical framework on the lines of [3]. Our

[2] For a full version see http://www.imsc.res.in/∼soumya/Files/Stability.pdf

main focus in this exposition is the strategies of players rather than the winning conditions themselves.

2.2 Strategies

A strategy for player i tells her at each game position, which action to choose. Given the game tree $\mathcal{T} = (T, \Rightarrow, t_0)$, a strategy μ for a player i is a function $\mu : T \to A_i$.

For a history $\bar{a}_1 \dots \bar{a}_k$ of the game, a strategy for player i after the history $\bar{a}_1 \dots \bar{a}_k$ is a function $\mu[\bar{a}_1 \dots \bar{a}_k] : \{\bar{a}_1 \dots \bar{a}_k u \in \mathcal{T}\} \to A_i$ where $u \in A^*$. Thus $\mu[\epsilon]$ is a strategy for the entire game and we denote it by μ itself. The function $\mu[\bar{a}_1 \dots \bar{a}_k]$ may be viewed as a subtree $\mathcal{T}^{\mu[\bar{a}_1 \dots \bar{a}_k]} = (T', \Rightarrow', t'_0)$ of $\mathcal{T}$ with root t'_0 such that $t'_0 = \bar{a}_1 \dots \bar{a}_k \in T'$ and

- For any node $t = \bar{a}_1 \dots \bar{a}_l \in T'$, $(l \geq k)$ if $\mu[\bar{a}_1 \dots \bar{a}_k](t) = a$ then the children of t in $\mathcal{T}^{\mu[\bar{a}_1 \dots \bar{a}_k]}$ are exactly those nodes $t\bar{a} \in \mathcal{T}$ such that the ith component of $\bar{a}$, $\bar{a}(i)$ is equal to a.

We shall call such a subtree $\mathcal{T}^{\mu[\bar{a}_1 \dots \bar{a}_k]}$, a strategy tree for the strategy $\mu[\bar{a}_1 \dots \bar{a}_k]$. Note that the values of $\mu[\bar{a}_1 \dots \bar{a}_k]$ at positions, $t \notin T'$ does not affect the outcome of a play conforming to $\mu[\bar{a}_1 \dots \bar{a}_k]$. Hence, we can interpret the semantics of a strategy in terms of its strategy tree without any loss of generality. We shall also use the terms 'strategy' and 'strategy tree' interchangeably. Let $\Omega_i(t)$ denote the set of all strategies of player i after history t in $\mathcal{T}$ and let $\Omega_i = \cup_{t \in \mathcal{T}} \Omega_i(t)$. Note that the set of strategies is infinite for any game $\mathcal{T}$.

Composition of Strategies: Let $\mu_1, \mu_2 \in \Omega_i$. Suppose player i starts playing the game $\mathcal{T}$ with strategy μ_1 and after k rounds $(k \geq 0)$, she decides to use the strategy μ_2 for the rest of the game. The resulting prescription is also a strategy μ (say) in the set of strategies of player i, that is, $\mu \in \Omega_i$. In a sense μ may be viewed as a composition of the strategies μ_1 and μ_2. We denote the strategy μ by $\mu_1^k \mu_2$.

The strategy tree $\mathcal{T}^{\mu_1^k \mu_2}$ for the strategy is obtained by taking $\mathcal{T}^{\mu_1}$ and removing all the nodes with height greater than or equal to $k+1$, resulting in a tree of height k, and pasting $\mathcal{T}^{\mu_2[\bar{a}_1 \dots \bar{a}_k]}$ at each leaf node $\bar{a}_1 \dots \bar{a}_k$ of this resulting tree.

2.3 Partial Strategies

Given $\mathcal{T} = (T, \Rightarrow, t_0)$, a history $\bar{a}_1 \dots \bar{a}_k \in \mathcal{T}$, a partial strategy $\sigma[\bar{a}_1 \dots \bar{a}_k]$ for player i after this history is a partial function

$$\sigma[\bar{a}_1 \dots \bar{a}_k] : \{\bar{a}_1 \dots \bar{a}_k u \in \mathcal{T}\} \rightharpoonup A_i$$

where $u \in A^*$, with the interpretation that if σ is not defined for some history $\bar{a}_1 \dots \bar{a}_k u \in \mathcal{T}$, the player may play *any* available action there.The strategy $\sigma[\epsilon]$ is identified with the strategy σ for the entire game.

The strategy tree $\mathcal{T}^{\sigma[\bar{a}_1 \ldots \bar{a}_k]} = (T', \Rightarrow', t'_0)$ is again a subtree of $\mathcal{T}$ with root $t'_0 = \bar{a}_1 \ldots \bar{a}_k \in T'$ and for any node $t = \bar{a}_1 \ldots \bar{a}_l \in T'$ $(l \geq k)$, if $\sigma[\bar{a}_1 \ldots \bar{a}_k](t) = a$, then the children of t are exactly those nodes $t\bar{a} \in \mathcal{T}$ such that the ith component of $\bar{a}$, $\bar{a}(i)$ is equal to a. On the other hand if $\sigma[\bar{a}_1 \ldots \bar{a}_k]$ is undefined on t, then the children of t are $\{t\bar{a} \mid t\bar{a} \in \mathcal{T}\}$, i.e., all the nodes that are the children of the node t in the game tree $\mathcal{T}$ itself.

We let $\Sigma_i(t)$ denote the set of all partial strategies of player i after history t in $\mathcal{T}$ and let $\Sigma_i = \cup_{t \in \mathcal{T}} \Sigma_i(t)$ denote the set of all partial strategies of player i.

A partial strategy may be viewed as a set of total strategies. Given the strategy tree $\mathcal{T}_G^\sigma$ for a partial strategy σ for player i we obtain a set of trees $\widetilde{\mathcal{T}}_G^\sigma$ of total strategies as follows. $\mathcal{T} = (T, \Rightarrow, t_0) \in \widetilde{\mathcal{T}}_G^\sigma$ if and only if $t_0 = \epsilon$ and

– If $\bar{a}_1 \ldots \bar{a}_k \in \mathcal{T}$ then $\bar{a}_1 \ldots \bar{a}_{k+1} \in \mathcal{T}$ if and only if $\bar{a}_1 \ldots \bar{a}_{k+1} \in \mathcal{T}_G^\sigma$ and for all $\bar{a}_1 \ldots \bar{a}_{k+1},\ \bar{a}_1 \ldots \bar{a}'_{k+1} \in \mathcal{T},\ \bar{a}_{k+1}(i) = \bar{a}'_{k+1}(i)$.

For any history $\bar{a}_1 \ldots \bar{a}_k$, the set $\widetilde{\mathcal{T}}_G^{\sigma[\bar{a}_1 \ldots \bar{a}_k]}$ of total strategy trees for the partial strategy $\sigma[\bar{a}_1 \ldots \bar{a}_k]$ of player i may be defined similarly.

It is convenient to define the maps $\mathcal{PT}_i$ and $\mathcal{TP}_i$ for all $i \in N$. $\mathcal{PT}_i : \Sigma_i \rightarrow 2^{\Omega_i}$, such that $\mathcal{PT}_i(\mathcal{T}_G^{\sigma[\bar{a}_1 \ldots \bar{a}_k]}) = \widetilde{\mathcal{T}}_G^{\sigma[\bar{a}_1 \ldots \bar{a}_k]}$. And $\mathcal{TP}_i : 2^{\Omega_i} \rightarrow \Sigma_i$, such that given a set $\widetilde{\mathcal{T}}_G^{\mu[\bar{a}_1 \ldots \bar{a}_k]}$ of total strategy trees of player i, $\mathcal{TP}_i(\widetilde{\mathcal{T}}_G^{\mu[\bar{a}_1 \ldots \bar{a}_k]})$ is the partial strategy tree $(T, \Rightarrow, t_0)$ such that $t_0 = \bar{a}_1 \ldots \bar{a}_k$ and

– $t \in T$ if and only if $t \in \mathcal{T}$ for some $\mathcal{T} \in \widetilde{\mathcal{T}}_G^{\mu[\bar{a}_1 \ldots \bar{a}_k]}$
– $\Rightarrow = \bigcup_{\mathcal{T} \in \widetilde{\mathcal{T}}_G^{\mu[\bar{a}_1 \ldots \bar{a}_k]}} \{\Rightarrow \in \mathcal{T}\}$.

2.4 Relevant Questions

Given the above notion of partial strategies, it makes sense to talk about what it means to compose several (usually simple) strategies to obtain another (more complex) strategy. A player will start out with a set (possibly finite) of *elementary* or *atomic* strategies, and as the game progresses, combine them to obtain new strategies. Switching from one strategy to another is based on certain observable properties of the game. Strategies thus generated may not be present in her initial set of strategies.

Given a region of the game arena, to check whether a player's strategy eventually becomes stable with respect to switching, we need to be able to first check whether a strategy is *live* in the region. For a subtree $\mathcal{T}'$ of $\mathcal{T}$ and a partial strategy σ_i of player i, we say σ_i is live in $\mathcal{T}'$ if $\forall t \in T'$ the following condition holds:

– if $\sigma_i(t)$ is defined and $\sigma_i(t) = a$ then $\exists t' = t\bar{a} \in T'$ such that $\bar{a}(i) = a$.

Given a game $\mathcal{T}$, natural questions of interest include:

– Given a subtree $\mathcal{T}'$ of $\mathcal{T}$ and a partial strategy σ_i, is σ_i live in $\mathcal{T}'$?
– Is it the case that a given strategy σ_i eventually becomes not live ?
– Find the set of all partial strategies which are live in a substructure.

Note that here we assume that every strategy is equally viable and switching between strategies does not involve any overhead. A model where different strategies have different *costs* would be interesting to study in its own right.

To solve these questions algorithmically and to subsequently address the stability issue, we need to present partial strategies and game trees in a finite manner. Below we show how this can be achieved.

3 Strategy Specifications

We present a syntax to specify partial strategies and their composition in a structural manner. We crucially use a construct which allows players to play the game with a strategy σ_1 up to some point and then switch to a strategy σ_2.

Syntax: The strategy set Π_i of player i is obtained by combining her atomic strategies as follows:

$$\Pi_i ::= \sigma \in \Sigma_i \mid \pi_1 \cup \pi_2 \mid \pi_1 \cap \pi_2 \mid \pi_1 {}^\frown \pi_2 \mid (\pi_1 + \pi_2) \mid \psi?\pi$$

Using the *test operator* $\psi?\pi$, a player checks whether an observable condition ψ holds and then decides on a strategy. We think of these conditions as past time formulas of a simple tense logic over an atomic set of observables.

In the atomic case, σ simply denotes a partial strategy. The intuitive meaning of the operators are given as:

- $\pi_1 \cup \pi_2$ means that the player plays according to the strategy π_1 or the strategy π_2.
- $\pi_1 \cap \pi_2$ means that if at a history $t \in \mathcal{T}$, π_1 is defined then the player plays according to π_1; else if π_2 is defined at t then the player plays according to π_2. If both π_1 and π_2 are defined at t then the moves that π_1 and π_2 specify at t must be the same (we call such a pair π_1 and π_2, compatible). Henceforth, we shall use the $\cap$ operator only for compatible pairs of strategies.
- $\pi_1 {}^\frown \pi_2$ means that the player plays according to the strategy π_1 and then after some history, switches to playing according to π_2. The position at which she makes the switch is not fixed in advance.
- $(\pi_1 + \pi_2)$ says that at every point, the player can choose to follow either π_1 or π_2.
- $\psi?\pi$ says at every history, the player tests if the property ψ holds of that history. If it does then she plays according to π.

Example: In the cricket example, let the bowler's set of atomic strategies be given as $\Sigma_{bowler} = \{\sigma_{short}, \sigma_{good}, \sigma_{outside-off}, \sigma_{legs}\}$ which corresponds to bowling a short-pitch, good length, off-side and leg-side ball respectively.

Let $p_{(short,sixer)}$ be the observable which says that the outcome of a short ball is a sixer. Then the following specification says that the bowler keeps bowling short balls till he is hit for a sixer after which he changes to good-length deliveries.

– $\neg\diamondminus(p_{(short,sixer)}?(\sigma_{short})) \cup \diamondminus(p_{(short,sixer)}?(\sigma_{good}))$

The specification $\sigma_{short}{}^\frown\sigma_{good}{}^\frown\sigma_{legs}$ for the bowler says that he starts by bowling short-pitch balls and after some point he switches to bowling at the batsman's legs and again switches to bowling good-length balls.

Semantics: Formally, given the game tree $\mathcal{T} = (T, \Rightarrow, t_0)$, the semantics of a strategy specification $\pi \in \Pi_i$ is a function $[\![\cdot]\!]_{\mathcal{T}} : \Pi_i \times T \to 2^{\Omega_i}$. That is, each specification at a node t of the game tree is associated with a set of total strategy trees after history t.

For any $t = \bar{a}_1 \ldots \bar{a}_k \in T$, $[\![\cdot]\!]_{\mathcal{T}}$ is defined inductively as follows:

- $[\![\sigma, (\bar{a}_1 \ldots \bar{a}_k)]\!]_{\mathcal{T}} = \mathcal{PT}_i(\mathcal{T}^{\sigma[\bar{a}_1 \ldots \bar{a}_k]})$.
- $[\![\pi_1 \cup \pi_2, (\bar{a}_1 \ldots \bar{a}_k)]\!]_{\mathcal{T}} = [\![\pi_1, (\bar{a}_1 \ldots \bar{a}_k)]\!]_{\mathcal{T}} \cup [\![\pi_2, (\bar{a}_1 \ldots \bar{a}_k)]\!]_{\mathcal{T}}$.
- $[\![\pi_1 \cap \pi_2, (\bar{a}_1 \ldots \bar{a}_k)]\!]_{\mathcal{T}} = [\![\pi_1, (\bar{a}_1 \ldots \bar{a}_k)]\!]_{\mathcal{T}} \cap [\![\pi_2, (\bar{a}_1 \ldots \bar{a}_k)]\!]_{\mathcal{T}}$.
- $[\![\pi_1{}^\frown\pi_2, (\bar{a}_1 \ldots \bar{a}_k)]\!]_{\mathcal{T}} = \bigcup_{l \geq k} [\![[\![\pi_1, (\bar{a}_1 \ldots \bar{a}_k)]\!]_{\mathcal{T}}{}^\frown(\pi_2, l)]\!]_{\mathcal{T}}$
  where $[\![[\![\pi_1, (\bar{a}_1 \ldots \bar{a}_k)]\!]_{\mathcal{T}}{}^\frown(\pi_2, l)]\!]_{\mathcal{T}}$ is defined as follows: For every tree $T \in [\![\pi_1, (\bar{a}_1 \ldots \bar{a}_k)]\!]_{\mathcal{T}}$, prune the tree T at depth l and call it T_l. Then $[\![[\![\pi_1, (\bar{a}_1 \ldots \bar{a}_k)]\!]_{\mathcal{T}}{}^\frown(\pi_2, l)]\!]_{\mathcal{T}}$ is the set of trees got by appending to every leaf node $\bar{a}_1 \ldots \bar{a}_l$ of such trees T_l, the trees in $[\![\pi_2, (\bar{a}_1 \ldots \bar{a}_l)]\!]_{\mathcal{T}}$.
- $[\![(\pi_1 + \pi_2), (\bar{a}_1 \ldots \bar{a}_k)]\!]_{\mathcal{T}} = \bigcup_{k_1, k_2, \ldots} [\![[\![[\![\pi_1, (\bar{a}_1 \ldots \bar{a}_k)]\!]_{\mathcal{T}}{}^\frown(\pi_2, k_1)]\!]_{\mathcal{T}}{}^\frown(\pi_1, k_2)]\!]_{\mathcal{T}} \cdots$
  where $k \leq k_1 \leq k_2 \ldots$.
- $[\![\psi?\pi, (\bar{a}_1 \ldots \bar{a}_k)]\!]_{\mathcal{T}}$:$[\![\psi?\pi, (\bar{a}_1 \ldots \bar{a}_k)]\!]_{\mathcal{T}}$ is obtained from $[\![\pi, (\bar{a}_1 \ldots \bar{a}_k)]\!]_{\mathcal{T}}$ and $\mathcal{T}$ as follows. Let $\mathcal{TP}_i([\![\pi, (\bar{a}_1 \ldots \bar{a}_k)]\!]_{\mathcal{T}}) = \mathcal{T}^{\pi[\bar{a}_1 \ldots \bar{a}_k]}$ be the partial strategy tree of $\pi[\bar{a}_1 \ldots \bar{a}_k]$. Then $[\![\psi?\pi, (\bar{a}_1 \ldots \bar{a}_k)]\!]_{\mathcal{T}}$ is a set of trees such that the following holds. $T \in [\![\psi?\pi, (\bar{a}_1 \ldots \bar{a}_k)]\!]_{\mathcal{T}}$ if and only if:
  - $\bar{a}_1 \ldots \bar{a}_k \in T$.
  - If $\bar{a}_1 \ldots \bar{a}_l \in T$ and ψ holds at $\bar{a}_1 \ldots \bar{a}_l$ then $\bar{a}_1 \ldots \bar{a}_{l+1} \in T$ if and only if $\bar{a}_1 \ldots \bar{a}_{l+1} \in \mathcal{T}^{\pi[\bar{a}_1 \ldots \bar{a}_k]}$ and for all $\bar{a}_1 \ldots \bar{a}_{l+1}, \bar{a}_1 \ldots \bar{a}'_{l+1} \in T, \bar{a}_{l+1}(i) = \bar{a}'_{l+1}(i)$. If $\bar{a}_1 \ldots \bar{a}_l \in T$ and ψ does not hold at $\bar{a}_1 \ldots \bar{a}_l$ then $\bar{a}_1 \ldots \bar{a}_{l+1} \in T$ if and only if $\bar{a}_1 \ldots \bar{a}_{l+1} \in \mathcal{T}$ and for all $\bar{a}_1 \ldots \bar{a}_{l+1}, \bar{a}_1 \ldots \bar{a}'_{l+1} \in T, \bar{a}_{l+1}(i) = \bar{a}'_{l+1}(i)$.

4 Finite Presentation of Games and Strategies

For algorithmic analysis, we need to present the infinite game in a finite fashion. In this paper, we assume that the game is presented as a finite graph. The extensive form game is just the unfolding of this graph.

Game Arena: The game arena is a finite graph $G = (W, \to, w_0)$ where W is a finite set of game positions, $w_0 \in W$ is the initial position and $\to: W \times A \to W$, is the set of edges. For $w \in W$, let $w_\to = \{\bar{a} \mid w \xrightarrow{\bar{a}} w' \text{ for some } w' \in W\}$. For technical convenience, we assume that for all $w \in W$, $w_\to \neq \emptyset$. The infinite extensive form game tree corresponding to G is obtained by the *tree-unfolding* of G.

For a word on notation, given an arena G and a strategy specification π, we denote the function $[\![\cdot]\!]_{\mathcal{T}_G}$ by just $[\![\cdot]\!]_G$.

Finite State Transducers and Bounded Memory Strategies: A finite state transducer (FST) over the input alphabet A and output alphabet A_i is a tuple $\mathcal{A} = (Q, \rightarrow, I, f)$ where Q is a finite set of states, $I \subseteq Q$ is the set of initial states, $\rightarrow: Q \times A \rightarrow 2^Q$ is the transition function and $f : Q \rightarrow A_i$ is the output function.

The semantics of strategy specifications is presented with respect to the set of all strategies. For algorithmic concerns we restrict our attention to bounded memory strategies. As we will see later, strategy specifications can only enforce bounded memory strategies.

A strategy σ of player i is said to be bounded memory if there exists an FST $\mathcal{A} = (Q, \rightarrow, I, f)$ where the set of states Q is the *memory* of σ, I is the initial memory, $\rightarrow$ is the *memory update function* and f is the *action output function* such that the following is true. When $\bar{a}_1 \dots \bar{a}_{k-1}$ is a play and the sequence $q_0, q_1, \dots, q_k$ is determined by $q_0 \in I$ and $q_i \xrightarrow{\bar{a}_i} q_{i+1}$ then $\sigma(\bar{a}_1 \dots \bar{a}_{k-1}) = f(q_k)$. The intuition is that the FST faithfully reflects the outputs of the strategy σ.

Given a strategy μ of player i, a run of an FST $\mathcal{A}$ on $\mathcal{T}_G^\mu$ is a Q labelled tree $(T, \Rightarrow, t_0, \chi)$. The labelling function $\chi : T \rightarrow Q$ is defined as: $\chi(t_0) = q_0 \in I$ and if $\bar{a}_1 \dots \bar{a}_k \Rightarrow \bar{a}_1 \dots \bar{a}_{k+1}$ then $\chi(\bar{a}_1 \dots \bar{a}_k) \in\rightarrow (\chi(\bar{a}_1 \dots \bar{a}_k), \bar{a}_{k+1})$.

We say that μ is accepted by $\mathcal{A}$ if there is a run χ of $\mathcal{A}$ on $\mathcal{T}_G^\mu$ satisfied the condition: $\forall t = \bar{a}_1 \dots \bar{a}_k \in \mathcal{T}_G^\mu, \bar{a}(i) = f(\chi(t))$. The language of $\mathcal{A}$, $\mathcal{L}(\mathcal{A}) = \{\mu \mid \mu$ is accepted by $\mathcal{A}\}$.

The following lemma relates strategy specifications to finite state transducers.

Lemma 4.1. *Given game arena G, a player $i \in N$ and a strategy specification $\pi \in \Pi_i$, where all the atomic strategies mentioned in π are bounded memory, we can construct an FST $\mathcal{A}_\pi$ such that for all $\mu \in \Omega_i$ we have $\mu \in [\![\pi]\!]_G$ iff $\mu \in \mathcal{L}(\mathcal{A}_\pi)$.*

5 Stability

Call a strategy π switch-free if it does not have any of the $\frown$ or the $+$ construct. Given a strategy $\pi \in \Pi_i$ of player i, the set of substrategies of π, S_π are just the subformulae of π. Let $SF(S_\pi)$ be the set of switch-free strategies of S_π. Note that $SF(S_\pi)$ is a finite set for a given π.

Given a game arena G and strategy specifications of the players, we may ask whether there exists some subarena of G that the game settles down to if the players play according to their strategy specifications. This subarena is in some sense the equilibrium states of the game. It is also meaningful to ask if the game settles down to such an equilibrium subarena, then whether the strategy of a particular player attains stability with respect to switching.

Let $G = (W, \rightarrow, w_0)$ be the game arena, $\pi \in \Pi_i$ and $\mathcal{A}_\pi = (Q, \rightarrow, I, f, \lambda)$ be the FST for π. We define the restriction of G with respect to $\mathcal{A}_\pi$ as $G \restriction \mathcal{A}_\pi = (W', \rightarrow', w'_0)$ where $W' = W \times Q$, $w'_0 = \{w_0\} \times I$ and $(w_1, q_1) \overset{\bar{a}}{\rightarrow}' (w_2, q_2)$ iff $w_1 \overset{\bar{a}}{\rightarrow} w_2$, $q \overset{\bar{a}}{\rightarrow} q_2$ and $f(q_1) = \bar{a}(i)$.

Theorem 5.1. *Given a game arena $G = (W, \rightarrow, w_0)$ with a valuation of the observables on W, a subarena R of G and strategy specifications $\pi_1, \ldots, \pi_n$ for players 1 to n, the question, "Do all plays conforming to these specifications eventually settle down to R?" is decidable.*

Proof. Construct the graph $G_\pi = (\cdots((G \restriction \mathcal{A}_{\pi_1}) \restriction \mathcal{A}_{\pi_2} \cdots) \restriction \mathcal{A}_{\pi_n}) = (W_\pi, \rightarrow_\pi, w_\pi)$. Let $F \subseteq G_\pi = (W', \rightarrow')$ such that $W' = \{(w, q_1, \ldots, q_n) \mid w \in R, q_1 \in Q_{\pi_1}, \ldots, q_n \in Q_{\pi_n}\}$ where $Q_{\pi_1}, \ldots, Q_{\pi_n}$ are the state sets of the FST's $\mathcal{A}_{\pi_1}, \ldots, \mathcal{A}_{\pi_n}$ respectively. Let $\rightarrow' = \rightarrow_\pi \cap (W' \times W')$.

1. Check if F is a maximal connected component in G_π. If so proceed to step 2, else output a 'NO'.
2. Check if all paths strating at all initial nodes $w' \in w_\pi$ reach F and output a 'YES'. Otherwise, output a 'NO'.

Theorem 5.2. *Given a game arena $G = (W, \rightarrow, w_0)$ with a valuation of the observables on W, a subarena R of G and strategy specifications $\pi_1, \ldots, \pi_n$ for players 1 to n, the question, "If all plays conforming to these specifications converge to R, does the strategy of player i become eventually stable with respect to switching?" is decidable in time $\mathcal{O}(m^m \cdot p \cdot 2^{np})$ where m is the size of the arena G and p is the maximum length of a specification formula π_1 or $\ldots$ or π_n.*

Proof. We first check if all the plays settle down to R. But in doing so we also have to keep track of all the strategy-switches ($\frown$'s) of player i along these plays. Because given a subformula of the form $\pi \frown \pi'$, once the player has switched to strategy π' she cannot play π later. We do this by an inductive procedure by first indexing all the subformulae of the specification π_i of player i and then augmenting the FST's with an output so that at each point they output the indices of only those subformulae that are still relevant at that point. We also keep track of the states each of the FST's are in when the plays reach R. Having done so, we check, by constructing FST's for each relevant switch free substrategy of π_i, whether the play stays inside R if player i plays according to that substrategy, given that all the FST's start at the states in which they were on reaching R.

6 Discussion

The framework presented here is intended only as an initial step of a research programme that studies computational models of social interaction. It is to be noted that the presentation of game arenas as graphs may be inappropriate for many contexts, and it may be more natural to define games by rules.

Moreover, the assumption of a fixed finite set of players may also be unrealistic for models of social dynamics. The notion of strategy switching is demonstrated quite naturally in a framework which consists of a population of players and a neighbourhood model. In such a set up, the players are parts of different neighbourhoods and can observe the outcomes within their neighbourhoods. Such a structure besides giving a rationale to the players for playing certain strategies and switching between them, would also model various game-theoretic and social scenarios more concretely. We hope that the study of formal models of dynamics in interaction will lead to new questions for games and computations.

References

1. R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of ACM*, 49(5):672–713, 2002.
2. T. Arnold and U. Schwalbe. Dynamic coalition formation and the core. *Journal of Economic Behavior and Organization*, 49:363–380, 2002.
3. G. Bonanno. Modal logic and game theory: Two alternative approaches. *Risk Decision and Policy*, 7:309–324, December 2002.
4. S. Ghosh. Strategies made explicit in dynamic game logic. In *Logic and the Foundations of Game and Decision Theory*, 2008.
5. U. Horst. Dynamic systems of social interactions. In *NSF/CEME Mathematical Economics Conference at Berkeley*, 2005.
6. R. Ramanujam and S. Simon. Dynamic logic on games with structured strategies. In *Proceedings of the Conference on Principles of Knowledge Representation and Reasoning*, pages 49–58, 2008.
7. B. Skyrms and R. Pemantle. A dynamic model of social network formation. *Proceedings of the National Academy of Sciences*, 97(16):9340–9346, 2000.
8. J. vanBenthem. Games in dynamic epistemic logic. *Bulletin of Economic Research*, 53(4):219–248, 2001.
9. J. vanBenthem. Extensive games as process models. *Journal of Logic Language and Information*, 11:289–313, 2002.
10. W. vanderHoek, W. Jamroga, and M. Wooldridge. A logic for strategic reasoning. In *Proceedings of AAMAS*, pages 157–164, 2005.
11. W. vanderHoek and M. Wooldridge. Cooperation, knowledge, and time: Alternating-time temporal epistemic logic and its applications. *Studia Logica*, 75(1):125–157, 2003.
12. M. D. Vos and D. Vermeir. A logic for modeling decision making with dynamic preferences. In *Proceedings of JELIA*, volume 1919 of *LNAI*, pages 391–406, 2000.
13. M. D. Vos and D. Vermeir. Dynamic decision-making in logic programming and game theory. In *Australian Joint Conference on Artificial Intelligence*, pages 36–47, 2002.
14. J. W. Weibull. *Evolutionary Game Theory*. MIT Press, 1997.
15. H. P. Young. The evolution of conventions. In *Econometrica*, volume 61, pages 57–84. Blackwell Publishing, 1993.
16. H. P. Young. The diffusion of innovations in social networks. Economics Working Paper Archive 437, The Johns Hopkins University,Department of Economics, May 2000.