Dynamic Restriction of Choices: A Preliminary Logical Report

Soumya Paul, R. Ramanujam and Sunil Simon
The Institute of Mathematical Sciences
CIT Campus, Taramani
Chennai - 600 113, India
e-mail: {soumya,jam,sunils}@imsc.res.in

Abstract

We study games in which the choices available to players are not fixed, and may change during the course of play. Specifically, we consider a model in which players may switch strategies, and a global (social) decision may remove some choices, based on the strategies being adopted by players. We propose a logical formalism in which such choices are specified, and a model of bounded memory strategies in which the eventual implications of such choices can be computed, and present preliminary results.

1 Introduction

Outfitting oneself in India before the 1980s was an elaborate affair. One had to buy cloth from a store, decide on the design and style of the garment, and get it stitched by a tailor. Along the way, many personal preferences and current fashion trends would play a role. But gradually, as readymade garments came into the market, it became clear that this was a cheaper and quicker option, though this severely limited one's say in the finer design details. As more people bought readymades, they became cheaper still, and with fewer customers, tailors had to charge more to sustain themselves. Today, there are very few practising tailors, and getting one's clothes custom made is a luxury. Whether this choice will even be available a few generations from now is unclear.

Economists are, of course, well aware of such phenomena. The availability of individual choices is, in general, determined by choices by the society as a whole, and in turn, social choices are influenced by patterns of individual choices. In this process, the set of choices may expand or contract over time.

However, there is a political or philosophical value attached to availability of individual choices. A strategy a may be justified by the presence of another option b but if eventually b is forced out, the rationale for a may disappear, though a is the only one present. In a world where all possible eventual consequences can be computed, the cost of such disappearance of choices can also be taken into account, but (as we see in the case of environment conservation) realisation typically comes post-facto.

To see this, consider a toll booth on a busy road which is manually operated. A vehicle driving through has to stop, tender cash and only then is allowed to proceed. Hence it is suggested that toll collection be RFID based. A vehicle equipped with RFID can speed through an automatic lane, and the requisite amount will be debited from the bank account of the owner of the vehicle. While this is welcome, protesters point to loss of privacy, since the movements of the car owner can then be tracked. RFID is defended on the grounds that anyone worried about privacy can always use the lane with the manual booth. Thus speed and privacy are traded off against each other and the RFID system is introduced. Gradually, as more people use the fast lanes, only one lane is operated manually, and there comes a point when the manual booth is removed on the grounds that it is too expensive to maintain. Interestingly, there is almost no public debate when this is done.

What happened to the trade-off between speed and privacy? It can be argued that a strategy valued by so few is socially a luxury to maintain, but the point remains that the rationale and terms of debate have changed entirely. While the question of whether this is acceptable or not is interesting for political philosophers, we suggest that there is at least a clear case for models that compute such eventual consequences of social decisions.

The general situation is as follows. At every stage, an individual has certain choices to make. But making a choice also comes with a cost which is associated with that choice and which the individual has to incur in making the choice. On the other hand, society also incurs a certain cost in making these choices available to individuals. This cost is a function of the choices being provided as well as the profile of choices made by individuals.

From time to time, based on the history of choice profiles and predictions of the future, society revises the choices it provides to individuals as well as the cost individuals have to incur to make these choices. This in turn has an effect on individuals' strategies, who switch between available choices. The dynamics of this back and forth process can be quite interesting and complicated.

Indeed, the decision on whether a facility should be provided as a part of *social infrastructure* (as opposed to being individually maintained, based on affordability) is based on such computations. Society may well decide that it is in its interest to *ensure* that everyone gets access to a facility, however uniform, rather than having a range of choices available but only to a subset of people. (As an example, consider Singapore offering free island-wide WiFi connectivity for three years.)

In game theoretic models of such social phenomena, social rules are considered as game forms, and individual behaviour is regulated using payoffs. Rule changes are considered to be exogenous, and correspond to change of payoff matrices. In evolutionary game theory, rules are considered as game equilibria: individuals following rules are players, and the desired properties of rules are given by equilibrium strategies, thus describing enforced rules. However what we discuss here is endogenous dynamics of these rules that takes into account the fact that individual behaviour and rules operate mutually and concurrently. In this sense, individual rationality and social rationality are mutually dependent, and what we seek to study are the patterns of *reasoning* that inform such dependence.

We thus consider game forms which change dynamically, but according to pre-specified rules stated in a formal logic. If players had unlimited computational power, they could startegise about all possible game changes as well, but we consider players with bounded computational ability, who formulate initial strategic plans and revise them during course of play, based on observation. Such switching is again described logically. This, in turn, determines applicability of game changing rules, and so on. We can then ask, in this model, which action choices are eventually *stable* (in the sense that no further game changes will eliminate them), and under what conditions. We may also ask if a player eventually gets removed by the dynamics of the game, if eventually a particular action tuple becomes the only choice available forever or if the cost stabilises to some specific amount. We show that these questions are algorithmically solvable.

It is important to emphasise that we focus on qualitative reasoning rather than quantitative analysis. However, the framework is appropriate for any such analysis in which costs and payoffs take values from a finite set, which is realistic when limited to players who have only a bounded memory.

Related Work: Game Dynamics

Strategy switching by players and rule changing are by no means new notions for game theory, and have indeed been studied extensively in evolutionary game theory. Weibull ([Wei97]) studies how players observe payoffs obtained by others and change their behaviour accordingly. [PW07] studies a model where actions of players depend on the forecasted outcome. It is not required that all players arrive at the same expectation. Depending on past forecasting errors, players switch strategies and update their behaviour. The behaviour switching of the players in effect causes the game itself to change in a dynamic fashion reflecting the choices which fall out of favour for all the players.

Dynamic learning has also been extensively studied in game theory. Young ([PY93]) 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. In [PY00] he defines and studies the long run dynamics of a model of how innovations spread in a social network. [BS99] looks at equilibrium selection by players who revise strategies by a learning process. They note that the stable behaviour of agents depend on the dynamics of the game itself and argue that it is important to incorporate these changes into the model. Switching behaviour of players has also been studied in dynamical system models of social interaction ([SP00], [Hor05]).

Going further, Hashimoto and Kumagai ([HK03]) even propose a model in which interaction rules of replicator equations change dynamically, and offer computer simulations of dynamic game changes.

Given such extensive work by dynamical system theorists, the need for qualitative reasoning, with all its severe limitations, in the same framework may well be questioned. It should be noted that dynamical system models typically work with fixed evolution rules (which are complicated enough) and study changes in parameter values. The relative advantage of logics is that rather than working with a fixed rule, we can study an entire class of rules (that can be specified in the logic) in a general fashion. Moreover, use of automata theoretic techniques for solving the problems (as done here) gives us uniform algorithms, as opposed to numerical solutions and computer simulations. Further, we hope that inference systems can be built which explicate the logical connections between individual and collective rationality in such contexts.

Related Work: Logical Studies

Our work is situated in the logical foundations of game theory, and hence employs logical descriptions of strategies and automata based algorithms to answer questions. Modal logics have been used in various ways to reason about games and strategies. Notable among these is the work on alternating temporal logic (ATL) and its extensions ([AHK02], [HW03], [HJW05], [JHW05]): assertions are made on outcomes a coalition of players can ensure, and what strategy a player plays may depend on her intensions and epistemic attitudes. In [Ben01,Ben02] van Benthem uses dynamic logic to describe games as well as strategies. [Gho08] presents a complete axiomatisation of a logic describing both games and strategies in a dynamic logic framework where assertions are made about atomic strategies. [RS08] studies a logic in which not only are games structured, but so also are strategies. [Ben07] lists a range of issues to be studied in reasoning about strategies.

Somewhat different in approach, and yet related, is the work of De Vos and Vermeir ([VV00],[VV02]) 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.

In our earlier work ([PRS09]), we study a *process-like* notion of strategy in the context of strategy switching. In particular, we show the decidability of a stability question: given a game arena and a strategy specification, whether players eventually settle down to strategies without further switching. There are two critical differences between that study and the one presented here. Firstly, in [PRS09], the game arena is fixed, choices are static and players are the only actors; in contrast, this work is about changing game arenas, and the dependence of individual choices and social decisions on each other. Secondly, the logic in [PRS09] is much more intricate and includes an explicit switching operator. However, the techniques used (automata and transducer constructions) in the two papers are similar: one uses them to study a complex logic on static models, and the other to study a simple logic on dynamic models. We hope to integrate these into one framework, but the framework is too messy as yet to admit this.

2 Preliminaries

We study games in extensive form which are given as trees of game positions (game trees), with branching denoting choices of moves. However, since we wish to study games that change dynamically, what we need is not a single tree, but a collection of trees. Moreover, if the game duration is fixed, game changes can be predetermined statically, and hence we consider games of unbounded duration which can be modelled by infinite game trees. Since the main object of this paper concerns algorithmic analysis of games, we require that the game structure is presented in a finite fashion. One possibility is to describe the structure in terms of a finite set of rules. However, a simpler approach, which is the one we adopt in this discussion, is to present the game structure as a finite game arena which is a finite graph. The infinite extensive form game is then just the unfolding of this finite graph. We are thus led to unbounded duration games played on finite arenas. The model is formalised below.

2.1 Game Arena

Let $N = \{1, \ldots, n\}$ be the set of players. For each $i \in N$, let $A_i \cup \{\epsilon\}$ be a *finite* set of actions of player i. ϵ denotes a null action which is useful when we wish to record that the move made by a player was not *available*[1]. Let $A = \mathit{\Pi}_{i=1}^{n}(A_i \cup \{\epsilon\})$ and $\mathbb{A} = \bigcup_{i=1}^{n} A_i$. An arena $\mathcal{A} = (V, E)$ is a finite graph with the vertex set V and edge relation E. For $v \in V$, let $vE = \{(v, u) \in E\}$, i.e., the set of edges outgoing from v. The edges of the arena are labelled with labels from A. For an edge label $\mathbf{a} = (a_1, \ldots, a_n)$ we let $\mathbf{a}(i)$ denote the ith component of $\mathbf{a}$, i.e., $\mathbf{a}(i) = a_i$. We let $|\mathbf{a}|_a,\ a \in \mathbb{A} \cup \{\epsilon\}$, denote the number of a's present in the label $\mathbf{a}$, i.e., $|\mathbf{a}|_a = |\{i \mid 1 \leq i \leq n,\ a(i) = a\}|$. We assume that for every label $\mathbf{a}$ such that $|\mathbf{a}|_\epsilon \geq 1$ there always exists an edge in $\mathcal{A}$ with label $\mathbf{a}$. An initial vertex $v_0 \in V$ is distinguished and the game $G = (\mathcal{A}, v_0)$ consists of an arena $\mathcal{A}$ and the initial vertex v_0. A sub-arena $\mathcal{A}'$ of the arena $\mathcal{A}$ is a graph (V', E') such that $V' \subset V$ and E' is the set of edges induced by V'.

The game proceeds as follows. Initially a token is placed at v_0. If the token is at some vertex v, then players 1 to n simultaneously choose actions $a_1, \ldots, a_n$ from their action sets $A_1, \ldots, A_n$ respectively. This defines a tuple $\mathbf{a} = (a_1, \ldots, a_n)$. If $\mathbf{a}$ is the label of the edge (v, u) then the token is moved to u. If $\mathbf{a}$ is not present among the labels of the outgoing edges then for all $i : 1 \leq i \leq n$ such that the action $a(i)$ is not available to player i, $a(i)$ is replaced by ϵ in $\mathbf{a}$ to get $\mathbf{a}_\epsilon$. If (v, u) is the edge with label $\mathbf{a}_\epsilon$ then the token is moved to u. This defines a path $\pi = v_0 \xrightarrow{\mathbf{a}_0} v_1 \xrightarrow{\mathbf{a}_1} \ldots$ in the arena. Such a path is called a play. A finite play is also called a history.

Tree Unfolding

The tree unfolding of the arena $\mathcal{A}$ at a node v_0 is a subset $\mathcal{T}_\mathcal{A} \subset A^*$ such that $\epsilon \in \mathcal{T}_\mathcal{A}$ is the root and for all $t = \mathbf{a}_0\mathbf{a}_1\ldots\mathbf{a}_k \in \mathcal{T}_\mathcal{A}$ such that $v_0 \xrightarrow{\mathbf{a}_0} \ldots \xrightarrow{\mathbf{a}_k} v_k$ is the corresponding path in $\mathcal{A}$, $t\mathbf{a} \in \mathcal{T}_\mathcal{A}$ for all $(v_k, u) \in v_kE$ such that (v_k, u) is labelled with $\mathbf{a}$. For a node $t = \mathbf{a}_0\mathbf{a}_1\ldots\mathbf{a}_k \in \mathcal{T}_\mathcal{A}$ such that $v_0 \xrightarrow{\mathbf{a}_0} \ldots \xrightarrow{\mathbf{a}_k} v_k$ is the corresponding path in $\mathcal{A}$, we let $\lambda(t) = v_k$. We also use the notation $\mathcal{T}_G$ to denote the tree unfolding of the game $G = (\mathcal{A}, v_0)$.

2.2 Strategies

A strategy of a player tells her how to play the game. In other words, it prescribes at every position which move to make. Formally a strategy μ_i of player i is a function μ_i :

[1] The notion will be made precise in Section 3

$A^* \to A_i$. Note that the codomain of μ_i is A_i and not $A_i \cup \{\epsilon\}$. The empty action is *not* a strategic choice for a player; rather it is *forced* when the action she plays is not available. A strategy μ_i can equivalently be thought of as a subtree $\mathcal{T}_{\mu_i}$, the strategy-tree, of $\mathcal{T}_{\mathcal{A}}$ with root corresponding to the position v_0 such that:

- For any node $t = \mathbf{a}_0\mathbf{a}_1 \ldots \mathbf{a}_k$ if $\mu_i(t) = a$ then the children of t in $\mathcal{T}_{\mu_i}$ are exactly those nodes $t\mathbf{a} \in \mathcal{T}_{\mathcal{A}}$ such that $\mathbf{a}(i)$ is equal to a.

A strategy μ is said to be bounded memory if there exists a finite state machine $\mathcal{M} = (M, g, h, m^I)$ where M is a finite set denoting the memory of the strategy, m^I is the initial memory, $G : A \times M \to M$ is the memory update function, and $h : A \times M \to A_i$ the output function which specifies the choice of the player such that if $\mathbf{a}_0 \ldots \mathbf{a}_k$ is a play and $m_0 \ldots m_{k+1}$ is a sequence determined by $m_0 = m^I$ and $m_{i+i} = g(\mathbf{a}_i, m_i)$ then $\mu(\mathbf{a}_0 \ldots \mathbf{a}_k) = h(\mathbf{a}_k, m_{k+1})$. The strategy μ is said to be memoryless if M is a singleton.

2.3 Dynamic Game Restriction

Having defined game arenas, we now proceed to consider game restrictions. The crucial elements are: *when* a restriction is to be carried out in the course of play, and *what* the effects of a restriction are. We choose a very simple answer to the latter, namely to eliminate a subset of choices at selected game positions. The former is treated logically, to be defined in the next section, by tests for logical conditions.

Formally the restriction is triggered by a rule of the form $r = pre \mapsto \mathcal{A}'$ where pre is a precondition which is interpreted on partial plays and $\mathcal{A}'$ is a restriction of the arena. For an arena $\mathcal{A}$ and a partial (finite) play $t \in \mathcal{T}_{\mathcal{A}}$, we say that the rule $r = pre \mapsto \mathcal{A}'$ is enabled at $(\mathcal{A}, t)$ if the following conditions hold.

- The partial play t conforms to the precondition pre (this notion will be made precise in the next section).
- The arena $\mathcal{A}' = (V', E')$ is a sub-arena of $\mathcal{A}$.
- $\lambda(\rho) \in V'$, i.e. the node in $\mathcal{A}$ which corresponds to the partial play t is present in $\mathcal{A}'$ as well.

When the rule $r = pre \mapsto \mathcal{A}'$ is applied to a partial play t, the game proceeds to the new arena $\mathcal{A}'$ starting at the node $\lambda(t)$.

Induced Game Tree

The restriction rules are specified along with the initial game arena. Let $\mathbb{R} = \{r_1, \ldots, r_m\}$ be a *finite* set of restriction rules. For an arena $\mathcal{A}$, let $Sub(\mathcal{A})$ denote the set of all subarenas of $\mathcal{A}$. Given a game arena $(\mathcal{A}, v_0)$ and a finite set of rules $\mathbb{R}$, the extensive form game tree is the (infinite) tree $\mathcal{T} = (S, \Rightarrow, s_0)$ where $S \subseteq A^* \times Sub(\mathcal{A})$ and $s_0 = (\epsilon, \mathcal{A})$. We start unfolding $\mathcal{A}$ starting at the node v_0. $\mathcal{T}$ is generated by the repeated application of the following steps:

- At any node $t = (\mathbf{a}_0\mathbf{a}_1 \ldots \mathbf{a}_k, \mathcal{A}')$ of the tree, check if a rule $(r_j = pre_j \mapsto \mathcal{A}_j) \in \mathbb{R}$ is enabled at $(t, \mathcal{A}')$. If more than one rule is enabled then choose any one of them.
- The subgame rooted at t is the unfolding of $\mathcal{A}_j$ at the node $\lambda(\mathbf{a}_0\mathbf{a}_1 \ldots \mathbf{a}_k)$.

2.4 Strategising by Players

Strategies, as defined earlier, are functions from nodes of the resulting game tree to actions of players. A strategy thus specifies the complete plan of a player. However, in the case of bounded memory agents, a player typically starts playing the game with information on game structure and on other players' skills, as well as an initial set of possible strategies to employ. As 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 fact that other players are similarly revising strategies is recognised and iterated on. The observations made by the player take into account actions of others and her own cost computation for the history of play. In addition to this, in the presence of dynamic game restriction operations, the player can keep track of the restriction rules which are triggered by observing the history of play and adapt her strategy based on this information.

A strategy specification for a player would therefore be of the form $pre \mapsto a_i$ where, as earlier, pre is a precondition which is interpreted on partial plays and a_i is an action of player i. The specification asserts that if a partial play t conforms to the precondition pre, then the action a_i is taken by the player.

Note that a strategy specification of this form is partial, since it does not constrain game positions at which the precondition does not hold; the player is free to choose any enabled action. This makes sense especially in the context of players with bounded memory, since they cannot (in general) compute all possible futures, or even keep a record of the entire past. Thus, in our model, a player starts the game with a finite set of such specifications and switches between these specifications by taking into account the history of play.

3 Logical Specifications

In this section we show that game arena restriction can be specified in a succinct manner in terms of homomorphisms and that restriction preconditions can be represented in a simple tense logic formalism.

3.1 Homomorphisms

A homomorphism is a function $h : \mathbb{A} \cup \{\epsilon\} \rightarrow \mathbb{A} \cup \{\epsilon\}$ such that $h(a)$ is either a or ϵ and $h(\epsilon) = \epsilon$.

Given an arena $\mathcal{A}$ and a homomorphism h, the restriction of $\mathcal{A}$ with respect to h, $\mathcal{A}|_h$ is defined as follows. An edge with label $\mathbf{a} = (a_1, \ldots, a_n)$ in $\mathcal{A}$ gets label $h(\mathbf{a}) = (h(a_1), \ldots, h(a_n))$ in $\mathcal{A}|_h$.

A homomorphism on an arena is thus nothing but the removal of one (or more) action(s) from the labels of the edges of the arena. Thus, in order to describe a homomorphism, it is enough to specify the action(s) to be removed. However, given an action we may not wish to remove the action from an individual's choice at all possible points but only at selective ones. This can be achieved by associating the restriction with respect to certain observables of the players.

3.2 Restriction Specifications

We now formally describe how we can specify these restrictions that the society imposes on the actions of the players. Let $\mathcal{P}$ be a set of propositions and $Bool(\mathcal{P})$ be the set of boolean formulas over $\mathcal{P}$ (i.e. built using the syntax $p \in \mathcal{P} \mid \neg\beta \mid \beta_1 \vee \beta_2$). We also use the following abbreviations: $\top \equiv p \vee \neg p$ and $\bot \equiv p \wedge \neg p$. Let $Val : V \rightarrow 2^{\mathcal{P}}$ be a valuation function. Val can be lifted to $\mathcal{T}_{\mathcal{A}}$ in the natural way, i.e., $Val(t) = Val(\lambda(t))$. The truth of a formula $\beta \in Bool(\mathcal{P})$ at a game position v, denoted $v \models \beta$ is defined as follows:

- $v \models p \in \mathcal{P}$ iff $p \in Val(v)$.
- $v \models \neg\beta$ iff $v \not\models \beta$.
- $v \models \beta_1 \vee \beta_2$ iff $v \models \beta_1$ or $v \models \beta_2$.

Given an arena $\mathcal{A}$ the restriction rules imposed by the society consists of a collection of specification of the form $\varphi \supset h$, where φ is a precondition specification and h is a specification of the homomorphism. The formal syntax and semantics is presented below

Syntax of Homomorphism Specifications

Homomorphisms are specified using the following syntax:

$$\mathfrak{h} ::= h_{\beta:a} \mid h_1 \wedge h_2, \text{ where } a \in \mathbb{A} \text{ and } \beta \in Bool(\mathcal{P}).$$

Semantics

For an arena $\mathcal{A}$ and a homomorphism specification h, we define the restriction of $\mathcal{A}$ with respect to h (denoted $\mathcal{A}|_h$) inductively as follows:

- $h \equiv h_{\beta:a}$: $\mathcal{A}|_{h_{\beta:a}}$ is $\mathcal{A}$ with the label $\mathbf{a} = (a_1, \ldots, a_n)$ of every edge $(v, u) \in vE$ replaced by $(h(a_1), \ldots, h(a_n))$ such that $h(\epsilon) = \epsilon$,
  - $h(a_i) = \epsilon$ if $v \models \beta$ and $a_i = a$, $h(a_i) = a_i$ otherwise.
- $h \equiv h_1 \wedge h_2$: $\mathcal{A}|_{h_1 \wedge h_2} = (\mathcal{A}|_{h_1})|_{h_2}$

Note that using the above notation the removal of all 'a' actions in the arena can be specified by $h_{\top:a}$ and the removal of a player i from the arena by $\bigwedge_{a \in A_i} h_{\top:a}$.

Syntax of Restriction Precondition

$$\varphi ::= p \in \mathcal{P} \mid \neg\varphi' \mid \varphi_1 \vee \varphi_2 \mid \langle\mathbf{a}\rangle^{-}\varphi' \mid \boxminus\varphi' \mid \langle\mathbf{a}\rangle^{+}\varphi'$$

As usual $\ominus\varphi' \equiv \bigvee_{\mathbf{a} \in A} \langle\mathbf{a}\rangle^{-}\varphi'$, $\bigcirc\varphi' \equiv \bigvee_{\mathbf{a} \in A} \langle\mathbf{a}\rangle^{+}\varphi'$ and $\diamondminus\varphi' \equiv \neg\boxminus\neg\varphi'$.

Semantics

A formula φ is evaluated on the game tree $\mathcal{T}_G$. The truth of φ at a node t of $\mathcal{T}_G$, denoted $\mathcal{T}_G, t \models \varphi$ is defined inductively as:

- $\mathcal{T}_G, t \models p$ iff $p \in Val(t)$
- $\mathcal{T}_G, t \models \neg\varphi'$ iff $\mathcal{T}_G, t \not\models \varphi'$
- $\mathcal{T}_G, t \models \varphi_1 \vee \varphi_2$ iff $\mathcal{T}_G, t \models \varphi_1$ or $\mathcal{T}_G, t \models \varphi_2$
- $\mathcal{T}_G, t \models \langle\mathbf{a}\rangle^{-}\varphi'$ iff $t = t'\mathbf{a}$ and $\mathcal{T}_G, t' \models \varphi'$
- $\mathcal{T}_G, t \models \boxminus\varphi'$ iff for all prefixes t' of t, $\mathcal{T}_G, t' \models \varphi'$
- $\mathcal{T}_G, t \models \langle\mathbf{a}\rangle^{+}\varphi'$ iff $t' = t\mathbf{a} \in \mathcal{T}_G$ and $\mathcal{T}_G, t' \models \varphi'$

The modality $\langle\mathbf{a}\rangle^{+}\varphi'$ talks about one step future. It asserts the existence of an $\mathbf{a}$ edge after which φ' holds. Note that future time assertions up to any finite depth can be coded by iteration of this construct. $\langle\mathbf{a}\rangle^{-}\varphi'$ is the corresponding construct for one step past. $\boxminus\varphi'$ makes assertion about the unbounded past, it specifies the transitive closure of the one step past operator. i.e. all states in the past satisfies φ'. We can define the corresponding construct for future, $\Box\varphi'$ with the following interpretation:

- $\mathcal{T}_G, t \models \Box\varphi'$ iff for all t' such that t is a prefix of t', $\mathcal{T}_G, t' \models \varphi'$

The technical results of this paper goes through even with the addition of this construct. However, for the applications we have in mind, this construct is not required.

3.3 Strategy Specifications

The strategy of players depend on properties of the history of the play. These can therefore be specified as a collection of formulae of the form $\psi \supset a$ where ψ is a simple past time tense logic formula. The syntax of ψ is given as follows.

$$\psi ::= p \in \mathcal{P} \mid \neg\psi' \mid \psi_1 \vee \psi_2 \mid \langle \mathbf{a} \rangle^{-}\psi' \mid \boxminus\psi'$$

A formula ψ of player i is evaluated on the game tree $\mathcal{T}_G$. Then the truth of ψ at a node t of $\mathcal{T}_G$, denoted $\mathcal{T}_G, t \models \psi$ is defined inductively as given earlier.

3.4 Capturing Costs in the Logical Formalism

Following a strategy induces a certain cost for the player. The distribution of strategies chosen by players carry a social cost. In this model, we have taken an abstract view of costs associated with individual players and social costs associated with providing facilities. In a quantitative model, each move $a_i \in A_i$ made by player i would be associated with a local cost c_i. Given a history $\mathbf{a}_0 \ldots \mathbf{a}_k$, the accumulated cost of player i would be some function of the form $C_i = \Sigma_{i=0}^{k-1} \beta^i c_i$ where β is a discounting factor. The player would then reason about how to play based on her accumulated cost. The social cost typically depends on the history of the choices made by players in the past. When the social cost crosses some pre-defined threshold, it might be socially optimal to make certain facilities part of the common infrastructure which reduces the individual costs.

When the costs arise from a fixed finite set, they can be coded up using propositions in the logical framework on the lines of [Bon02]. The cost c can be represented using the proposition p_c and orderings are inherited from the implications available in the logic. Furthermore, costs can be dependent on the actions enabled at a game position. This can also be easily represented in the logical formalism by making use of the one step future modality.

3.5 Examples

Revisiting the tailor example, suppose there are two players 1 and 2. Each of them have two choices initially: t for going to a tailor and r for opting for readymade. Suppose initially the social cost is 5 units. Suppose the cost functions are as follows: the cost of going to a tailor is 2/5 times the social cost and the cost of going for a readymade is 3/5 times the social cost. Also suppose that initially both players play t. Player 1 has the condition that if at any point, the cost of t becomes 2.5 or more then she switches to r and player 2 has the condition that if at any point, the cost of t becomes 3 or more then she switches to r. Suppose the propositions for the social costs are $\{p_4, p_5, p_7, p_8\}$ where p_4 is supposed to mean that the social cost is 4 units and so on. Then the strategy of player 1 is: $\mathbb{S}_1 = \{\ominus(p_4 \vee p_5) \supset t, \ominus(p_7 \vee p_8) \supset r\}$ and that of player 2 is: $\mathbb{S}_2 = \{\ominus(p_4 \vee p_5 \vee p_7) \supset t, \ominus p_8 \supset r\}$.

Now suppose after 2 moves, the social cost rises to 7. This is modelled in the arena by having paths $v_0 \xrightarrow{(\cdot,\cdot)} v_1 \xrightarrow{(\cdot,\cdot)} v_2$ where $p_7 \in \mathit{Val}(v_2)$. Then player 1 switches to play r. Also suppose the social cost increases to 8 when 1 of the players play r. This is modelled in the arena by having paths $v \xrightarrow{(r,\cdot)/(\cdot,r)} v'$ where $p_8 \in \mathit{Val}(v')$. Then player 2 also switches to r. Further suppose if the social cost increases to 8 then the society decides to do away with all the tailors. This is given by the restriction specification $\ominus p_8 \mapsto h_{\top:t}$.

3.6 Stability

Let $G = (\mathcal{A}, v_0)$ be a game, $\mathbb{R}$ be a finite set of game restriction rules, $\{\mathbb{S}_i\}_{i \in N}$ be a finite set of strategy specifications or each player $i \in N$. Let α be a formula from the syntax:

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

We say α is stable in $(G, \mathbb{R}, \{\mathbb{S}_i\}_{i \in N})$ if there exists a sub-arena $\mathcal{A}'$ such that for all game positions $t \in \mathcal{T}_{\mathcal{A}'}$, we have: $\mathcal{T}_{\mathcal{A}'}, t \models \alpha$. Thus stability with respect to an observable property captures the existence of a subarena to which the game stabilises under the dynamics specified by $\mathbb{R}$ and $\{\mathbb{S}_i\}_{i \in N}$. For the applications we consider, we do not require the full power of temporal logic for α.

4 Results

In this section we present the main theorem of this paper. The questions addressed here are representatives of the kind of questions one can ask and prove of the model.

Theorem 1. *Given a game $G = (\mathcal{A}, v_0)$, a finite set of restriction rules $\mathbb{R}$, a finite set of strategy specifications $\{\mathbb{S}_i\}_{i \in N}$ and a formula α, the following question is decidable:*

– *Is α* stable *in $(G, \mathbb{R}, \{\mathbb{S}_i\}_{i \in N})$?*

Proof. Let $\mathbb{R} = \{(\varphi_1 \mapsto h_1), \ldots, (\varphi_m \mapsto h_m)\}$ and $\mathbb{S}_i = \{(\psi_1^i \supset a_1^i), \ldots, (\psi_{k_i}^i \supset a_{k_i}^i)\}$ for each player i. Let $Cl(\alpha)$ denote the sub-formula closure of a temporal formula α. For a homomorphism specification h, let $EL(h)$ denote the set of all atomic homomorphism specifications in h. For $H = \{h_1, \ldots, h_m\}$, let $EL(H) = EL(h_1) \cup \ldots \cup EL(h_m)$. The proof is carried out in the following steps.

Step 1. For each of the restriction rules $\varphi_j \mapsto h_j$, we construct a finite state automaton $\mathcal{R}_j$ which works as follows: the state space of $\mathcal{R}_j$ consists of the set of all maximal consistent subsets of φ_j (atoms of φ_j). The automaton runs on the game arena and keeps track of the game positions where φ_j is satisfied. We then construct the restriction automaton $\mathcal{R}$ which runs $\mathcal{R}_1, \ldots, \mathcal{R}_m$ in parallel. In addition, it also keeps track of the set $X \subseteq 2^{EL(H)}$ of atomic homomorphisms which are enabled. The set X is updated by the behaviour of the individual automata $\mathcal{R}_j$. At any point when the automaton $\mathcal{R}_j$ indicates that φ_j holds, the rule is triggered and $EL(h_j)$ is added to the set X. A formal definition of the automaton is given in the appendix.

Step 2. For each of the strategy specification $\psi^i_j \supset a^i_j$, we first construct a finite state automaton $\mathcal{S}_{\psi^i_j}$ which keeps track of whether ψ^i_j holds at a game position. As earlier, the state space of the automaton is the set of atoms of ψ^i_j.

For player i, we construct a finite state transducer $\mathcal{S}_i$ which generates the strategy of i in conformance with the specifications $\mathbb{S}_i$. $\mathcal{S}_i$ is a finite state machine equipped with an output function. It simulates the automata $\mathcal{S}_{\psi^i_j}$ for all j as well as the restriction automaton $\mathcal{R}$ in parallel. At every position suppose $\psi^i_{j_1}, \ldots, \psi^i_{j_l}$ holds at that position. $\mathcal{S}_i$ chooses one of $\psi^i_{j_1}, \ldots, \psi^i_{j_l}$ non-deterministically, say $\psi^i_{j_*}$. $\mathcal{S}_i$ then outputs action $a^i_{j_*}$ iff

- For all the atomic homomorphism specifications $\varphi \mapsto h_{\beta:a}$ trigerred by $\mathcal{R}$ so far, either $\varphi \notin s$ where s is the current state of $\mathcal{S}_{\psi^i_{j_*}}$ or $a^i_{j_*} \neq a$ or $a^i_{j_*} = a \Rightarrow \beta \notin s$.

The output is ϵ otherwise.

The formal automaton construction is provided in the appendix.

Step 3. A transducer $\mathcal{S}$ simulates all the $\mathcal{S}_j$'s, $1 \leq j \leq n$, in parallel. That is, $\mathcal{S}$ is a product of all the $\mathcal{S}_j$'s. It's output are action tuples which are the actions output by the individual transducers, $\mathcal{S}_j$'s. The restriction automaton $\mathcal{R}$ operates on the output of $\mathcal{S}$. Finally a master transducer $\mathcal{Q}$ simulates $\mathcal{R}$ and $\mathcal{S}$ in parallel. $\mathcal{Q}$ is a product of $\mathcal{R}$ and $\mathcal{S}$ and its output is the same as that of $\mathcal{S}$.

Figure 1 shows the interdependence between the various automata.

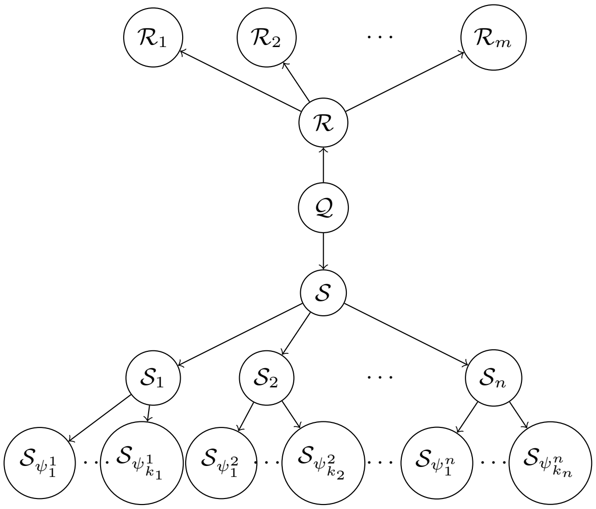

Fig. 1.

Step 4. Let $\mathcal{Q} = (Q, \rightarrow, I, f)$ be the master transducer constructed as above. For $q \in Q$, we say that $Val(q) = P$ iff for each component q_i of q, which are states of the restriction automaton $\mathcal{R}$ and the strategy transducer $\mathcal{S}$, $q_i \cap \mathcal{P} = P$.

We define the restriction of the game with respect to $\mathcal{Q}$, $G \restriction \mathcal{Q}$ as follows. $G \restriction \mathcal{Q} = (V', E', v'_0)$ where

- $V' = V \times Q$
- $E' \subset V' \times V'$ such that $(v, q) \xrightarrow{\mathbf{a}} (v', q')$ iff $(v, v') \in E$ and $q \xrightarrow{\mathbf{a}} q'$ and $f(q) = \mathbf{a}(i)$ for all $1 \leq i \leq n$ and $Val(v') = Val(q')$.
- $v'_0 = \{v_0\} \times I$ such that $(v_0, q) \in v'_0$ iff $Val(v_0) = Val(q)$.

To answer the stability question, construct the restricted graph $G \restriction \mathcal{Q}$ as described above.

- Check if there is a maximal connected component F in $G \restriction \mathcal{Q}$ and whether all paths starting from all initial vertices reach F. If no, then output 'NO' and quit.
- Check if α holds at all the game positions in F. If so output 'YES', else output 'NO'.

Corollary 1. *Given a game G and specifications $\mathbb{R}$ and $\{\mathbb{S}_i\}_{i \in N}$, the following questions are decidable:*

1. *Does player i eventually get removed by the dynamics of the game?*
2. *Does a particular action tuple $\mathbf{a}$ become the only choice available for ever?*
3. *Does the cost stabilise to a specific amount c?*

Proof. In each case, we come up with a formula α using the coding mentioned in section 3.4 such that answering the question amounts to checking the stability of α in $(G, \mathbb{R}, \{\mathbb{S}_i\}_{i \in N})$, which is decidable by Theorem 1.

For (1), we can code the positions of player i using a proposition $turn_i$ and check if $\alpha = \neg turn_i$ is stable in $(G, \mathbb{R}, \{\mathbb{S}_i\}_{i \in N})$. This asks whether it is the case that the rules of the society and the behaviour of other players drive a particular player out of the game. The negation of this question can also be answered: Does player i survive till the end of the game?

For (2), we check if $\alpha = \langle \mathbf{a} \rangle^+ \top \wedge \bigwedge_{\mathbf{a}_* \neq \mathbf{a}} \langle \mathbf{a}_* \rangle^+ \bot$ is stable in $(G, \mathbb{R}, \{\mathbb{S}_i\}_{i \in N})$. This corresponds to deciding whether the action $\mathbf{a}$ eventually becomes part of the social infrastructure. The choices available to players disappear in such a scenario. (3) follows from a similar argument.

Complexity

Let p be the maximum size of all the ψ^i_j formulae and $k = \max_{i \in N} |\mathbb{S}_i|$. The size of each $\mathcal{S}_j$, $1 \leq j \leq n$ is

$\mathcal{O}(k \cdot 2^p)$ and therefore the size of $\mathcal{S}$ is $\mathcal{O}(nk \cdot 2^p)$. Likewise, let q is the maximum size of the φ_i formulae and $l = |EL(H)|$. The size of $\mathcal{R}$ is $\mathcal{O}(m \cdot 2^q \cdot 2^l)$ and hence that of $\mathcal{Q}$ is $\mathcal{O}(mnk \cdot 2^{p+q+l})$. The size of the restricted graph $G \upharpoonright \mathcal{Q}$ is therefore $\mathcal{O}(|G| \cdot mnk \cdot 2^{p+q+l})$. Checking for connected components can be done in time polynomial in the size of the graph. When α is a conjunction of boolean formulas and one step future formulas, the truth checking can be done efficiently in linear time. The complexity of the construction given in theorem 1 is $\mathcal{O}(|G| \cdot mnk \cdot 2^{p+q+l})$.

4.1 Consequences of Theorem 1

Theorem 1 implies that comparison between game restriction rules in terms of their imposed social cost is possible. Suppose the "type" of players is known in terms of the strategy specification employed (note that we do not insist on knowing the exact strategy) and we have two sets of game restriction rules $\mathbb{R}_1$ and $\mathbb{R}_2$. It is possible to compute the social cost with respect to $\mathbb{R}_1$ and $\mathbb{R}_2$ and deduce which is better suited. From the players' perspective, if the game restriction rules are known and the type of other players are known, then they can compare between their strategy specifications. For instance, in the tailor example, this process might help a tailor to adapt better to the competition from ready-made manufacturers. He might be able to change his service into something of a hybrid form where the basic stitching itself is mechanised with respect to a fixed range of sizes. However, certain specific personalisation can be done by employing fewer number of workers, thereby being cost efficient.

5 Discussion

We have presented a simple formalism for describing and reasoning about endogenous dynamics of games, specifically about social restrictions on individuals' choices. We wish to emphasise that the model formulation and the stability theorem are intended as preliminary results in a larger programme of study. Ongoing work includes the study of other questions such as *rule synthesis*: rather than specifying homomorphism specifications, given a goal α to be achieved, we seek to synthesise rules of the form $\varphi \mapsto h$ that ensure stability of α. Note that such a question is a natural analogue of mechanism design in our framework.

While we have confined our study here to removal of actions (and players), *introduction* of new actions and players is also interesting, and needs considerable changes in the framework. Another line of work relates to *hierarchies*: there is no reason to limit the interaction studied here to one level of social aggregation, except that of technical convenience.

6 Appendix

6.1 Restriction Automaton

The automaton $\mathcal{R}_i$ for $\varphi_i : 1 \leq i \leq m$ is defined as $\mathcal{R}_i = (R_i, \rightarrow_i, I_i, F_i)$ over alphabet $2^{\mathcal{P}}$ where

- $R_i = AT(\varphi_i)$ are the atoms (maximal consistent states) of the subformula closure of φ_i.
- I_i is the set of initial states. These are the states that do not contain subformulae of the form $\ominus\varphi$.
- F_i is the set of final states. These are states that contain φ_i.
- $r_1 \overset{P}{\rightarrow}_i r_2$ iff the following conditions hold.
  - For all $\langle \mathbf{a} \rangle^- \varphi \in Cl(\varphi_i)$, $\langle \mathbf{a} \rangle^- \varphi \in r_2$ iff $\varphi \in r_1$.
  - For all $[\mathbf{a}]^+ \varphi \in Cl(\varphi_i)$, $[\mathbf{a}]^+ \varphi \in r_1$ implies $\varphi \in r_2$.
  - $r_1 \cap \mathcal{P} = P$.

Let $EL(h)$ denote the set of all atomic homomorphism specifications in h. For $H = \{h_1, \ldots, h_m\}$, let $EL(H) = EL(h_1) \cup \ldots \cup EL(h_m)$. The restriction automaton $\mathcal{R}$ is a tuple $\mathcal{R} = (R, \rightarrow, I)$ over alphabet A where

- $R = \mathit{\Pi}_{j=1}^{m} R_j \times 2^{EL(H)}$.
- $I = I_1 \times \ldots \times I_m \times \emptyset$. That is, the initial state is one that corresponds to the identity homomorphism.
- $(q_1, \ldots, q_m, X) \overset{\mathbf{a}}{\rightarrow} (q'_1, \ldots, q'_m, Y)$ iff $q_j \rightarrow_j q'_j$, $\forall j : 1 \leq j \leq m$ and
  - $Y = X \cup EL(h_k)$ if $q_k \in F_k$. That is, if the k'th restriction has been enabled then the automaton keeps track of it by adding it to the set X.
  - $Y = X$ otherwise.

6.2 Strategy Transducer

The automaton $\mathcal{S}_{\psi^i_j}$ for ψ^i_j is a tuple

$$\mathcal{S}_{\psi^i_j} = (S_{\psi^i_j}, \rightarrow_{\psi^i_j}, I_{\psi^i_j}, F_{\psi^i_j})$$

over alphabet $2^{\mathcal{P}}$. $\mathcal{S}_{\psi^i_j}$ simulates the atoms of ψ^i_j similar to the construction of $\mathcal{R}_i$

The strategy transducer for player i, $\mathcal{S}_i$ is a tuple $\mathcal{S}_i = (S_i, \rightarrow_i, I_i, f_i)$ over input alphabet A and output alphabet $A_i \cup \{\epsilon\}$ where S_i is the set of states, $\rightarrow_i$ is the transition relation, I_i is the initial state and f_i is the output function. The transducer output function f_i generates the strategy of player i.

- $S_i = \mathit{\Pi}_{j=1}^{k_i} S_{\psi^i_j} \times R$. $\mathcal{S}_j$ where R is the state space of $\mathcal{R}$.
- $I_i = \mathit{\Pi}_{j=1}^{k_i} I_{\psi^i_j} \times I$

- $f_i(s_1, \ldots, s_{k_i}, r) = a^i_j$ iff s_j is a final state of $\mathcal{S}_{\psi^i_j}$ and of all the atomic homomorphisms $\varphi \mapsto h_{\beta:a}$ enabled by $\mathcal{R}$ so far, $\varphi \notin s_j$ or $a^i_j \neq a$ or $\beta \notin s_j$. $f_i(s_1, \ldots, s_{k_i}, r) = \epsilon$ otherwise.
- $(s_1, \ldots, s_{k_i}, r) \xrightarrow{\mathbf{a}}_i (s'_1, \ldots, s'_{k_i}, r')$ iff $s_i \rightarrow_{\psi^i_j} s'_i$, $\forall j : 1 \leq j \leq k_i$ and $r \xrightarrow{\mathbf{a}} r'$.

References

[AHK02] Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. *Journal of ACM*, 49(5):672–713, 2002.

[Ben01] J. van Benthem. Games in dynamic epistemic logic. *Bulletin of Economic Research*, 53(4):219–248, 2001.

[Ben02] J. van Benthem. Extensive games as process models. *Journal of Logic Language and Information*, 11:289–313, 2002.

[Ben07] J. van Benthem. In praise of strategies. In J. van Eijck and R. Verbrugge, editors, *Foundations of Social Software*, Studies in Logic, pages 283–317. College Publications, 2007.

[Bon02] G. Bonanno. Modal logic and game theory: Two alternative approaches. *Risk Decision and Policy*, 7:309–324, December 2002.

[BS99] K. Binmore and L. Samuelson. Evolutionary drift and equilibrium selection. *Review of Economic Studies*, 66:363–393, 1999.

[Gho08] S. Ghosh. Strategies made explicit in dynamic game logic. In *Logic and the Foundations of Game and Decision Theory*, 2008.

[HJW05] Wiebe van der Hoek, Wojciech Jamroga, and Michael Wooldridge. A logic for strategic reasoning. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 157–164, 2005.

[HK03] Takashi Hashimoto and Yuya Kumagai. Meta-evolutionary game dynamics for mathematical modelling of rules dynamics. In *Advances in Artificial Life*, volume 2801 of *LNCS*, pages 107–117. Springer, 2003.

[Hor05] Ulrich Horst. Dynamic systems of social interactions. In *NSF/CEME Mathematical Economics Conference at Berkeley*, 2005.

[HW03] Wiebe van der Hoek and Michael Wooldridge. Cooperation, knowledge, and time: Alternating-time temporal epistemic logic and its applications. *Studia Logica*, 75(1):125–157, 2003.

[JHW05] Wojciech Jamroga, Wiebe van der Hoek, and Michael Wooldridge. Intentions and strategies in game-like scenarios. In *Multi-agent Systems: Theory and Applications*, volume 3808 of *Lecture Notes in Computer Science*, pages 512–523, 2005.

[PRS09] S. Paul, R. Ramanujam, and S. Simon. Stability under strategy switching. In *Proceedings of Computability in Europe 2009 (to appear)*, 2009.

[PW07] W.R. Parke and G.A. Waters. An evolutionary game theory explanation of ARCH effects. *Journal of Economic Dynamic and Control*, 31:2234–2262, 2007.

[PY93] H Peyton Young. The evolution of conventions. *Econometrica*, 61(1):57–84, 1993.

[PY00] H Peyton Young. The diffusion of innovations in social networks. Economics Working Paper Archive 437, The Johns Hopkins University,Department of Economics, May 2000.

[RS08] 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.

[SP00] Brian Skyrms and Robin Pemantle. A dynamic model of social network formation. *Proceedings of the National Academy of Sciences*, 97(16):9340–9346, 2000.

[VV00] Marina De Vos and Dirk Vermeir. A logic for modeling decision making with dynamic preferences. In *Proceedings of the Logic in Artificial Intelligence (JELIA)*, volume 1919 of *LNAI*, pages 391–406, 2000.

[VV02] Marina De Vos and Dirk Vermeir. Dynamic decision-making in logic programming and game theory. In *Australian Joint Conference on Artificial Intelligence*, pages 36–47, 2002.

[Wei97] Joergen W. Weibull. *Evolutionary Game Theory*. MIT Press, 1997.