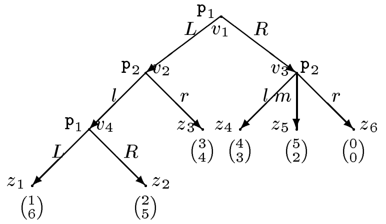
Subgame Perfect Nash Equilibria in Dynamic Logic

Paul Harrenstein[*], Wiebe van der Hoek, John-Jules Meyer
Department of Computer Science, Utrecht University
& Cees Witteveen
Technical University, Delft

Multi-agent environments comprise decision makers whose deliberations involve reasoning about the expected behaviour of other agents. Our research concerns the development of specification languages for such multi-agent systems. As such our particular interest is with concepts of rational choice as they have been studied and formalised within game theory as well as with their integration in a qualitative and logical framework. Here we are concerned with the incorporation of the important game-theoretical notion of a subgame perfect Nash equilibrium in a modified version of Propositional Dynamic Logic (PDL). Extensive forms of games are linked up to Kripke frames. This enables us to characterize those frames in which the strategic choices of the agents correspond to Nash-optimal solutions by means of a formula scheme of our logical language. The present analysis is restricted to generic and finite games with perfect information and no chance moves. Furthermore, the focus will be on qualitative, rather than on quantitative features.

In extensive form games are represented as trees, the internal nodes of which are each assigned to a player. As the edges of the game tree represent the possible moves the players can make, a strategy σ for a player comprises complete instructions how to move at each of the nodes assigned to him/her. Given strategies $\sigma_1, \ldots, \sigma_n$, one for each of the players $\mathtt{p}_1, \ldots, \mathtt{p}_n$, a strategy profile $\bar{\sigma} = \langle \sigma_1, \ldots, \sigma_n \rangle$ determines a unique outcome of the game. In strategic form an n-person game is represented by setting all strategies of the respective players against one another in an n-dimensional matrix. For an example, see figure 1:



$\mathtt{p}_1 \backslash \mathtt{p}_2$	ll	lm	lr	rl	rm	rr
LL	1\6	1\6	1\6	3\4	3\4	3\4
LR	2\5	2\5	2\5	3\4	3\4	3\4
RL	**4\3**	5\2	0\0	**4\3**	5\2	0\0
RR	**4\3**	5\2	0\0	**4\3**	5\2	0\0

Figure 1

A strategy profile $\bar{\sigma} = \langle \sigma_1, \ldots, \sigma_n \rangle$ is in Nash-equilibrium if for each player $\mathtt{p}_i$, $\bar{\sigma}$ guarantees him/her the highest payoff *given the strategies of the other players*, i.e. for each $\sigma'_i \neq \sigma_i$, the strategy profile $\langle \sigma_1, \ldots, \sigma'_i, \ldots, \sigma_n \rangle$ does not grant $\mathtt{p}_i$ a higher pay-off than $\bar{\sigma}$ does. A strategy profile is in subgame perfect Nash equilibrium if it is in Nash-equilibrium for all its subgames.

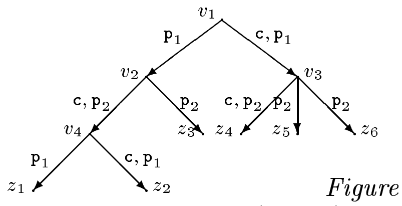
[*]Paul Harrenstein is sponsored by the CABS-project and Technical University, Delft.

In establishing a Nash-optimal strategy profile one will be confronted by certain fixed point concerns. When focussing on extensive forms these can be resolved by the backward induction algorithm. Backward induction defines a relation (the backward induction relation) on the game tree, which will here be denoted by $\lessdot$.

Apart from the payoff structure and the players, a game in extensive form $\mathcal{G}$ is a Kripke frame $\mathcal{F}$ ($\mathcal{G} \simeq \mathcal{F}$). The nodes correspond to states and the edges define the accessibility relation. Our approach is based on the idea that each player, $\mathtt{p}$, can be conceived of as a non-deterministic atomic program, interpreted as the transitions that correspond to the edges emanating from the nodes assigned to $\mathtt{p}$ in the game. Furthermore, a choice program $\mathtt{c}$ will be introduced to the language. This program is meant to reflect the strategy profile under consideration. In order to account for the (generic) payoff structure of the game for each player $\mathtt{p}$, a total order is imposed on the terminal states such that $v <_{\mathtt{p}} v' \iff \mathtt{u}(\mathtt{p})(v) < \mathtt{u}(\mathtt{p})(v')$, where $\mathtt{u}(\mathtt{p})(v)$ is the payoff $\mathtt{p}$ receives whenever the game terminates in node v. In order to gain expressive power with respect to this order, a new operator $\boxminus$ is added to the language with the following semantics:

$M, s \models \boxminus_{\mathtt{p}} \varphi :\iff$ for all leaf-nodes, z, z', such that sR^*z & $sR^*z' : M, z \models \varphi$ & $M, z' \not\models \varphi \Longrightarrow z' <_{\mathtt{p}} z$

As such, given the strategy profile $\langle RR, ll \rangle$, the game of figure 1 corresponds to the following frame:



$z_6 <_{\mathtt{p}_1} z_1 <_{\mathtt{p}_1} z_2 <_{\mathtt{p}_1} z_3 <_{\mathtt{p}_1} z_4 <_{\mathtt{p}_1} z_5$
$z_6 <_{\mathtt{p}_2} z_5 <_{\mathtt{p}_2} z_4 <_{\mathtt{p}_2} z_3 <_{\mathtt{p}_2} z_2 <_{\mathtt{p}_2} z_1$

Figure 2

In figure 1 the strategy profile $\langle RR, ll \rangle$ is in subgame perfect Nash-equilibrium. In figure 2 the corresponding $\mathtt{c}$ program coincides with the the backward induction relation of the corresponding game. Although some restrictions are imposed on the $\mathtt{c}$ program — it should be deterministic, irreflexive and defined on each internal node — this is not always the case. Our objective is to characterize those frames for which the $\mathtt{c}$ program coincides with the backward induction relation of the corresponding game by means of a formula scheme of the augmented language L.

In order to achieve this goal let for each $\mathtt{P}' \subseteq \mathtt{P}$, $\alpha_{\mathtt{P}'}$ be the complex progam:

$\mathtt{do}\ [\langle \mathtt{p}_1 \rangle \top \to \mathtt{p}_1 \Box \ldots \Box \langle \mathtt{p}_n \rangle \top \to \mathtt{p}_n \Box \bigvee_{\mathtt{q}_i \notin \mathtt{P}'} \langle \mathtt{q}_i \rangle \top \to \mathtt{c}]\ \mathtt{od}$,

where $\mathtt{P}' = \{\mathtt{p}_1, \ldots, \mathtt{p}_n\}$. Each run of the program $\alpha_{\mathtt{P}'}$ executes $\mathtt{p}$ for a $\mathtt{p} \in \mathtt{P}$ where possible, and $\mathtt{c}$ otherwise, and terminates if a final state is reached. Accordingly, $\alpha_{\{\mathtt{p}\}}$ corresponds to the non-deterministic program that terminates in exactly those states that $\mathtt{p}$ can guarantee the game to terminate in by varying his strategy provided the other players remain true to their strategs as encoded in $\mathtt{c}$. The program $\alpha_\emptyset$ consists of a repeated execution of $\mathtt{c}$ until it reaches a final state.

Finally, the desired correspondence result can be obtained:

Theorem 1 For each game $\mathcal{G}$ and each Kripke structure $\mathcal{F}$ such that $\mathcal{G} \simeq \mathcal{F}$:

$R_{\mathtt{c}} = \lessdot \iff$ for all $\mathtt{p} \in \mathtt{P}$: $\mathcal{F} \models (\langle \mathtt{p} \rangle \top \wedge \boxminus_{\mathtt{p}} \varphi \wedge \langle \alpha_{\{\mathtt{p}\}} \rangle \varphi) \to [\alpha_\emptyset] \varphi$

This theorem establishes that some model checking settles the question whether, when operating in circumstances that can be described as a perfect information game, agents behave as if they use backward induction in their reasoning.