Centrum voor Wiskunde en Informatica

REPORT*RAPPORT*

Game Logic for Game Theorists

M. Pauly

Information Systems (INS)

INS-R0017 September 30, 2000

Report INS-R0017
ISSN 1386-3681

CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands

Game Logic for Game Theorists

Marc Pauly
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

ABSTRACT
Game Logic (*GL*), introduced in [18], is examined from a game-theoretic perspective. A new semantics for *GL* is proposed in terms of *untyped* games which are closely related to extensive game forms of perfect information. An example is given of how *GL* can be used as a formal model of game situations, and some metatheoretic results are presented in the context of their game-theoretic relevance.

2000 Mathematics Subject Classification: 03B99, 91A10, 91A40
1998 ACM Computing Classification System: F.3.0, F.4.1
Keywords and Phrases: Dynamic Logic, Game Theory, Semantics
Note: Work carried out in the cluster INS "Information Systems".

1. Introduction

Over the last years, various logics have been proposed for formalizing certain aspects of reasoning in and about games. Most of these logics focus on the epistemic aspects of game-theoretic reasoning (see e.g. [13, 9, 3]): Given certain assumptions about the players' beliefs and knowledge, one can prove that a certain solution of the game (e.g. Nash equilibrium or backward induction) will be obtained. Non-epistemic game logics on the other hand have been investigated e.g. in the computer science literature, where system specifications can be modeled as a game between the system and its environment (see e.g. [4, 1]).

This paper investigates *Game Logic* (*GL*), a non-epistemic logic introduced in [18]. We hope that logics like *GL* can extend the game-theoretic research agenda in three ways: First, since *GL* is an adaptation of a logic used to reason about computer programs, the modeling approach taken differs from standard game-theoretic approaches, most importantly in the use of external states of the world which are independent of the positions in a game. The resulting model is that of a *game web* which is introduced in section 2 and which may be of interest to the game theorist. Along with this model also comes a non-standard notion of a game, an *untyped game*, which combines extensive games and game forms into one notion and which can be used to reason about games with incomplete information (section 3).

Second, *GL* presents a formal tool for modeling game situations. Such a tool may help to clarify terminological ambiguities or conceptual difficulties, as shown e.g. in [6]. On a more applied level, it may be used as a computational device, since conjectures about properties of a given game can be translated into formulas of the object language which can be given to an automated theorem prover (section 6). Section 5 applies *GL* to model an example of a game of incomplete information. It also demonstrates that a formal language for game construction allows one to address a simple instance of the implementation problem in a new way, namely by generalizing program synthesis to game synthesis.

Third, besides working *within* the language of *GL*, we can also investigate the logic itself to obtain metatheorems (section 6) which supply new game-theoretic information: It is shown e.g. that for the game operations under consideration, winning strategies of complex games can be constructed out of winning strategies for the subgames and that the game operations preserve determinacy. Furthermore, we mention some laws governing the interaction of *GL*'s game operations as well as additional game operations and the possibility of showing that a certain set of operations is sufficient to construct a particular class of games.

2. A Model of Interaction: Game Webs

A *game web*, the model of (inter-)action underlying Game Logic, is a generalization of the state-space model representing how a program transforms one computational state (an assignment of values to variables) into another. We assume that the world can be in a number of different states (states of the world, possible worlds, situations), and that actions taken by the agents can change this state in various ways. Interaction between the players may lead to a number of different resulting states, and the agents may prefer certain states of the world over others, e.g. an agent may prefer a state where she is happy over a state where she is unhappy, and/or a state where she has $100 over a state where she has $10. The resulting states of an interaction will usually again allow for further interactions to take place, leading to new resulting states, etc.. While the formalization of this model will be given in the next section, this informal description should already highlight the crucial features of this approach:

First, different interactions may be possible even at one single state of the world. Put differently, it may be possible to play various games in a given situation, so our model is not restricted to the analysis of a single game. The fact that resulting states of a game may be starting points for new games also shows that we are not modeling isolated games but a "web" of games.

Second, note that standard game-theoretic models such as extensive games do not include states of the *world*; what matters are the states of the *game*, and preferences can be defined over these without referring to states of the world. However, if we want to talk about a complex structure of interaction with possibly many games, general preferences over states of the world seem simpler and more natural than preferences over terminal states of all the games involved. Game-theoretically, one can think of states of the world as possible consequences resulting from interaction, and the agents' preferences are defined over these consequences rather than the outcomes of the interaction directly (see e.g. the alternative definition of strategic games in [17]).

Third, game webs allow to model situations with incomplete information: The agents' preferences may be only partially known, or they may not be known at all. However, we do not (yet) include probabilities or information sets in our model which are usually used in a game theoretic analysis of games of incomplete information. Instead, we will be able to reason explicitly about different types of players.

3. Interaction in Game Webs: Untyped Games

As mentioned in the previous section, game webs are based on a set of situations at which interaction can take place. Since interaction in game webs causes changes in the state of the world, and since the preferences of the agents may not be known, standard game-theoretic game models will have to be adopted to fit the situation at hand, resulting in the notion of an untyped game.

Given a nonempty set of states S, we define an *untyped game on* S between the players Angel (A, player 1) and Demon (D, player 2) as a 5-tuple

$$G = (H, P_A, W_A, W_D, \delta)$$

where H is a set of sequences (histories, plays, runs) subject to the following three standard conditions (see the definition of extensive games in [17]): (1) The empty sequence $\langle\rangle \in H$, (2) if $q = \langle q_0, q_1, \ldots, q_n\rangle \in H$ (where n may be infinite) and $m < n$ then $\langle q_0, q_1, \ldots, q_m\rangle \in H$, and (3) if for an infinite sequence $q = \langle q_0, q_1, \ldots\rangle$ we have $\langle q_0, q_1, \ldots, q_m\rangle \in H$ for every positive integer m, then $q \in H$. A history $q \in H$ is called *terminal* if it is either infinite or $q = \langle q_0, q_1, \ldots, q_n\rangle$ and there is no q_{n+1} such that $\langle q_0, q_1, \ldots, q_{n+1}\rangle \in H$. Let $H^t \subseteq H$ be the set of terminal runs and let $H^\infty \subseteq H^t$ be the set of infinite runs.

As for the other components of the definition, $P_A \subseteq \overline{H^t}$ is the set of nonterminal positions where it is Angel's turn to move; at all other nonterminal positions, it is Demon's turn. $W_A \subseteq H^t$ denotes the set of terminal runs won by Angel, and similarly $W_D \subseteq H^t$ denotes the set of terminal runs won by Demon. We require that $H^\infty \subseteq W_A \cup W_D$ and that $W_A \cap W_D = \emptyset$. We do *not* require that $W_A \cup W_D = H^t$, i.e. we allow for *open* or *undecided* terminal runs where neither player wins. Lastly, the function $\delta : \overline{H^\infty} \rightarrow S$ associates with every position (i.e. finite history) of the game a state of the world.

As with extensive games, we can depict untyped games as decorated trees. Nodes corresponding to terminal positions shall be drawn as squares, non-terminal positions as circles. A letter in a circle or square indicates the player whose turn it is to move, or the winner of the play in case of a terminal run. Thus, empty squares denote open terminal positions. States associated with positions by the δ-function are written to the right of the corresponding node. Figure 1 shows an example of an untyped game on the natural numbers which has one infinite run winning for Demon.

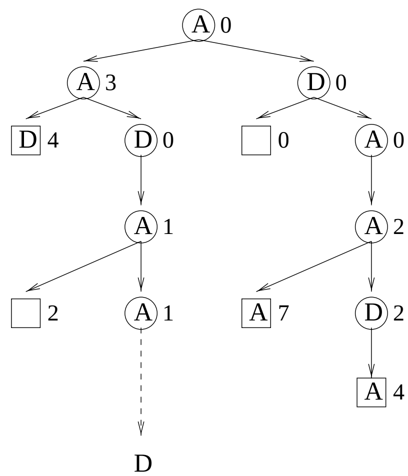

Figure 1: Example of an untyped game

Untyped games are closely related to extensive games of perfect information (see [17]). To be more precise, an untyped game is similar to a strictly competitive extensive game of perfect information between two players where the players' payoffs are either 1 or ⇔1. There are however two points in which untyped games differ from these extensive games. Both of these point are motivated by the game web model described. First, untyped games contain as an additional component a δ function which maps game positions to external states of the world. It allows us to keep track of how the state of the world changes through the actions of the players.

Second, untyped games can contain terminal histories without payoffs. From a game-theoretic point of view, untyped games are thus only semi-games, since they allow for these undecided terminal positions. The state associated with such a terminal position represents an abstract outcome, without specifying the players' valuation of the outcome; thus, an untyped game should be thought of more as an extensive game form rather than as a single extensive game. Extensive games and extensive game forms are the two extreme cases of untyped games: Extensive games are simply untyped games without undecided terminal nodes, i.e. $W_A \cup W_D = H^t$, whereas modulo infinite runs, extensive game forms are untyped games where all finite terminal runs are undecided, i.e. $W_A, W_D \subseteq H^\infty$. While in extensive game forms, an abstract outcome may also be associated with infinite plays, the present approach views infinite runs as changing the state of the world indefinitely, never converging to any definite outcome. Since players' preferences are over states, this requires us to specify directly which player wins such an infinite run. The definition of a winning strategy below will make this point clear.

The generality obtained by allowing for undecided terminal positions can be used to model incomplete information. Every set of states $X \subseteq S$ can be viewed as a possible player type, e.g. an angelic type. Hence, X is a set of outcomes which a certain angelic player may prefer. Given such an angelic type, we can consider all undecided terminal histories h such that $\delta(h) \in X$ a win for Angel. The result is a standard strictly competitive extensive game of perfect information, and we can ask e.g. whether Angel has a winning strategy in this typed game. Note that in case the original untyped game did not have any undecided terminal nodes, the type of the player is irrelevant for determining winning strategies.

The approach taken here to uncertainty about players' preferences should be sharply contrasted with Harsanyi's standard theory of incomplete information. In Harsanyi's approach, we start out with a number of extensive games of imperfect information, one for each type. Next it is assumed that one can attach probabilities to the different types and hence the original games can all be put together into one extensive game of imperfect information with chance move(s). In the approach taken here, the original extensive games are combined into one untyped game with undecided terminal position. The example of section 5 will demonstrate how these untyped games can be used to reason about a player's winning strategies against one or more types of opponents.

Given an untyped game $G = (H, P_A, W_A, W_D, \delta)$, a *strategy* for Angel in G is a function $\sigma : P_A \to H$ such that $\sigma(\langle q_0, \dots, q_n\rangle) = \langle q_0, \dots, q_n, k\rangle$; strategies for Demon are defined analogously. A run $q = \langle q_0, q_1, \dots q_n\rangle$ (where n may be infinite) *obeys Angel's strategy* σ iff for all $\langle q_0, q_1, \dots, q_m\rangle \in P_A$ with $m < n$ we have $\sigma(\langle q_0, \dots, q_m\rangle) = \langle q_0, \dots, q_m, q_{m+1}\rangle$. For any $Q \subseteq H$, let $Q^\sigma := \{q \in Q \mid q \text{ obeys } \sigma\}$.

Now as to the key concept, given a set of states $X \subseteq S$, a strategy σ for Angel is an X*-strategy* for Angel in G iff for all runs $q \in H^t$ obeying σ, either (1) $q \in W_A$ or (2) $q \notin W_A \cup W_D$ and $\delta(q) \in X$. The definition for Demon is again analogous, exchanging W_A with W_D. Call an untyped game G on S *determined* iff for every $X \subseteq S$, either Angel has an X-strategy in G or Demon has a $\overline{X}$-strategy in G, where $\overline{X} = S \Leftrightarrow X$. Note that by definition, it cannot happen that both have such a strategy. Given the previous remarks on how a type X transforms an untyped game into an extensive game, this definition of determinacy generalizes the standard notion of determinacy to games of incomplete information.

4. Syntax and Semantics of Game Logic

Game Logic is an extension of Propositional Dynamic Logic (*PDL*, see [11, 14]) which makes use of a more general semantics and adds a new program operator to *PDL*. The language of *GL* consists of two sorts, games and propositions. Given a set of atomic games $,_0$ and a set of atomic propositions Φ_0, games γ and propositions φ can have the following syntactic forms, yielding the set of games , and the set of propositions/formulas Φ:

$$\gamma := \quad g \mid \varphi? \mid \gamma;\gamma \mid \gamma \cup \gamma \mid \gamma^* \mid \gamma^d$$
$$\varphi := \quad \bot \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\gamma\rangle\varphi \mid [\gamma]\varphi$$

where $p \in \Phi_0$ and $g \in ,_0$. As usual, we define $\top := \neg\bot$, $\varphi \wedge \psi := \neg(\neg\varphi \vee \neg\psi)$ and $\varphi \to \psi := \neg\varphi \vee \psi$. Furthermore, we shall define a second demonic choice construct $\gamma_1 \cap \gamma_2$ as $(\gamma_1^d \cup \gamma_2^d)^d$.

Given an untyped game γ, the formula $\langle\gamma\rangle\varphi$ expresses that Angel has a φ-strategy in the untyped game γ. To provide some first intuition regarding the game operations, $\gamma_1 \cup \gamma_2$ denotes the game where Angel chooses which of the two subgames to continue playing. The sequential composition $\gamma_1;\gamma_2$ of two games consists of first playing γ_1 and then γ_2, and in the iterated game γ^*, Angel chooses how many γ games (possibly none) she wants to play. Playing the dual game γ^d is the same as playing γ with the players' roles reversed, i.e. any choice made by Angel in γ will be made by Demon in γ^d and vice versa. Hence, $\gamma_1 \cap \gamma_2$ will refer to the game where Demon chooses which subgame to play, leaving the roles of the players in γ_1 and γ_2 intact. The test game $\varphi?$ consists of checking whether a proposition φ holds at that position. This construction can be used to define conditional games such as $(p?;\gamma_1) \cup (\neg p?;\gamma_2)$: If p holds at the present state of the game, γ_1 is played, and otherwise γ_2.

Turning now towards the formal semantics, given $,_0$ and Φ_0, define a *game web* (game model) as a triple $\mathcal{I} = (S, \{G(g,s) | g \in ,_0 \text{ and } s \in S\}, V)$ where S is a set of states (the universe), and the $G(a,s)$ are determined untyped games on S such that the initial game position $\langle\rangle$ is associated with state s.

$V : \Phi_0 \to \mathcal{P}(S)$ is the valuation function which associates with every atomic proposition the set of states where it holds.

By simultaneous induction, we define truth in a game web on the one hand and the games which can be played in that model/web on the other hand. Formally, truth of a formula φ in a model $\mathcal{I}$ at a state s (denoted as $\mathcal{I}, s \models \varphi$) is defined as follows:

$$\begin{array}{ll}
\mathcal{I}, s \not\models \bot & \\
\mathcal{I}, s \models p & \text{iff } p \in \Phi_0 \text{ and } s \in V(p) \\
\mathcal{I}, s \models \neg\varphi & \text{iff } \mathcal{I}, s \not\models \varphi \\
\mathcal{I}, s \models \varphi \vee \psi & \text{iff } \mathcal{I}, s \models \varphi \text{ or } \mathcal{I}, s \models \psi \\
\mathcal{I}, s \models [\gamma]\varphi & \text{iff Demon has a } \varphi^{\mathcal{I}}\text{-strategy in game } G(\gamma, s) \\
\mathcal{I}, s \models \langle\gamma\rangle\varphi & \text{iff Angel has a } \varphi^{\mathcal{I}}\text{-strategy in game } G(\gamma, s)
\end{array}$$

where $\varphi^{\mathcal{I}} := \{s \in S | \mathcal{I}, s \models \varphi\}$. A formula φ is *valid in a model* $\mathcal{I}$ with universe S, denoted as $\mathcal{I} \models \varphi$, iff $\varphi^{\mathcal{I}} = S$, and φ is *valid* (denoted as $\models \varphi$) iff for all models $\mathcal{I}$ we have $\mathcal{I} \models \varphi$. Lastly, φ is a *consequence* of a set of formulas Δ (notation $\Delta \models \varphi$) iff φ is valid in every model in which all formulas of Δ are valid.

Given model $\mathcal{I}$, we define the untyped game $G(\gamma, s)$ for non-atomic games by induction on γ for all $s \in S$. Given two sequences q and q' where q is finite, let qq' denote the concatenation of the two sequences, and given a set of sequences Q, let $qQ := \{qq' | \ q' \in Q\}$. Figure 2 illustrates the formal definitions of the game constructions.

1. $G(\varphi?, s)$: If φ holds, then the game can be continued; otherwise, Demon has won. So we define

$$G(\varphi?, s) := \begin{cases} (\{\langle\rangle\}, \emptyset, \emptyset, \emptyset, \{(\langle\rangle, s)\}) & \text{if } \mathcal{I}, s \models \varphi \\ (\{\langle\rangle\}, \emptyset, \emptyset, \{\langle\rangle\}, \{(\langle\rangle, s)\}) & \text{otherwise} \end{cases}$$

2. $G(\alpha \cup \beta, s)$: Angel can choose whether to play α or β. Suppose we are given $G(\alpha, s) = (H_\alpha, P_\alpha, W_\alpha, W'_\alpha, \delta_\alpha)$ and $G(\beta, s) = (H_\beta, P_\beta, W_\beta, W'_\beta, \delta_\beta)$. Then let

$$G(\alpha \cup \beta, s) := \big(\ \{\langle\rangle\} \cup \langle 0\rangle H_\alpha \cup \langle 1\rangle H_\beta, \ P_A, \ W_A, \ W_D, \ \delta \ \big)$$

where $\delta(\langle\rangle) := s$, $\delta(\langle 0\rangle q) := \delta_\alpha(q)$ and $\delta(\langle 1\rangle q) := \delta_\beta(q)$. For the other parameters,

$$\begin{array}{rl}
P_A := & \{\langle\rangle\} \cup \langle 0\rangle P_\alpha \cup \langle 1\rangle P_\beta \\
W_A := & \langle 0\rangle W_\alpha \cup \langle 1\rangle W_\beta \\
W_D := & \langle 0\rangle W'_\alpha \cup \langle 1\rangle W'_\beta
\end{array}$$

3. $G(\alpha; \beta, s)$: Game β is played after game α, provided α has not already resulted in a win for one of the players. Suppose we are given $G(\alpha, s) = (H_\alpha, P_\alpha, W_\alpha, W'_\alpha, \delta_\alpha)$. Let $O_\alpha := H^t_\alpha \cap \overline{W_\alpha} \cap \overline{W'_\alpha}$ be the set of undecided (finite) terminal runs of $G(\alpha, s)$, and let $G(\beta, t) := (H_t, P_t, W_t, W'_t, \delta_t)$ for every $t \in O_\alpha$. Then in $G(\alpha; \beta, s) := (H, P_A, W_A, W_D, \delta)$, let

$$H := H_\alpha \cup \bigcup_{\substack{q \in O_\alpha \\ \delta_\alpha(q) = t}} qH_t$$

and $\delta(q) := \delta_\alpha(q)$ for $q \in H_\alpha$, and $\delta(qr) := \delta_t(r)$ for $q \in O_\alpha$ and $\delta_\alpha(q) = t$. This is well-defined since for $q \in O_\alpha$, $\delta_\alpha(q) = \delta_t(\langle\rangle) = t$. Furthermore,

$$P_A := P_\alpha \cup \bigcup_{\substack{q \in O_\alpha \\ \delta_\alpha(q) = t}} qP_t, \quad W_A := W_\alpha \cup \bigcup_{\substack{q \in O_\alpha \\ \delta_\alpha(q) = t}} qW_t, \quad W_D := W'_\alpha \cup \bigcup_{\substack{q \in O_\alpha \\ \delta_\alpha(q) = t}} qW'_t$$

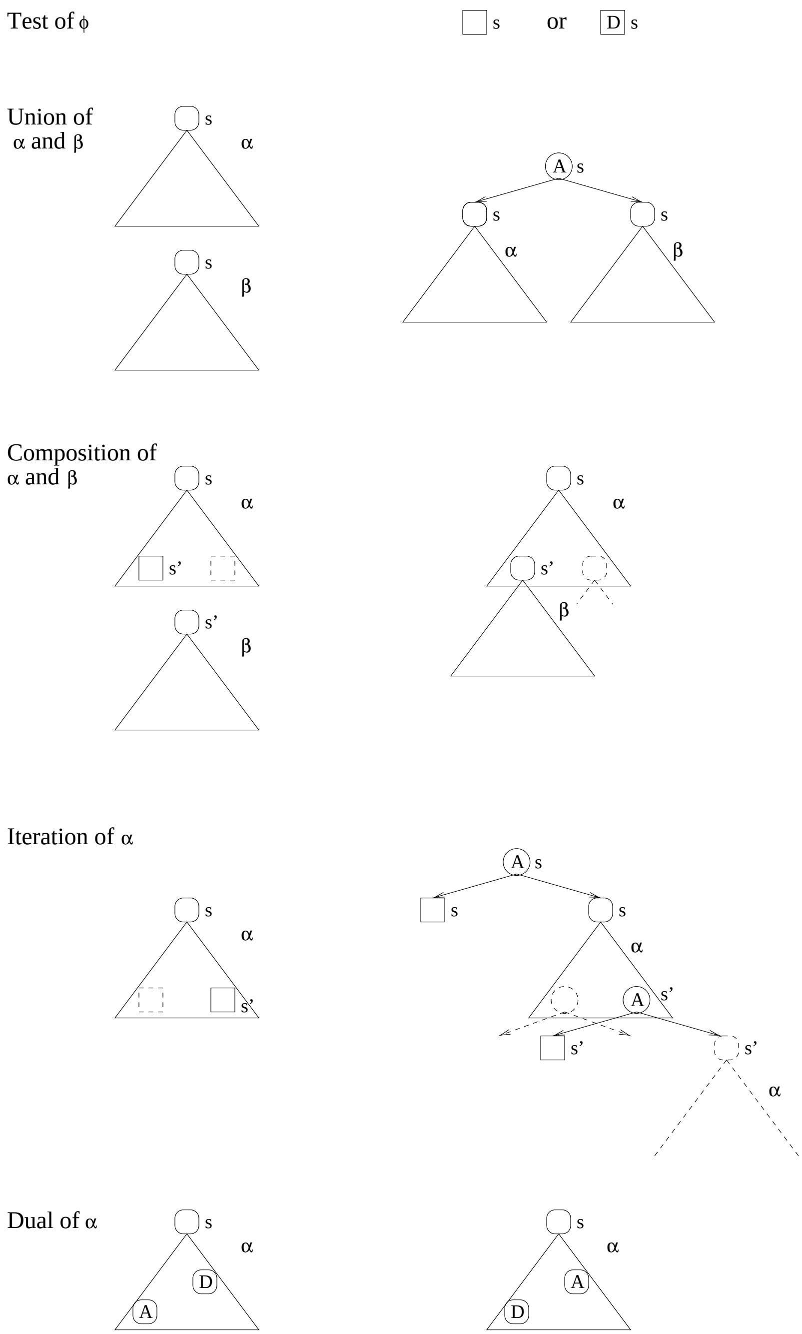

Figure 2: The construction of non-atomic games $G(\gamma, s)$

4. $G(\alpha^d, s)$: We simply interchange the roles of Angel and Demon, i.e. if $G(\alpha, s) = (H, P_A, W_A, W_D, \delta)$, we define

$$G(\alpha^d, s) := (H, \overline{H^t} \Leftrightarrow P_A, W_D, W_A, \delta)$$

5. $G(\alpha^*, s)$: Angel can choose whether or not to play α. If α has been played, she can choose to play it again, and so on, the only requirement being that Angel cannot choose to play α forever. We shall define the game in stages by inductively defining the game $G_n(\alpha, s)$ (or G_n for short) consisting of at most n α-iterations. At the same time, we shall define sets A_n which will serve to mark those sequences where we need to plug in another α-copy. For the base case, set $G_0 := (\{\langle\rangle\}, \emptyset, \emptyset, \emptyset, \{(\langle\rangle, s)\})$ and let $A_0 := \{\langle\rangle\}$.

   Now suppose we constructed game $G_n = (H_n, P_n, W_n, W'_n, \delta_n)$ and set A_n, where we can assume that $A_n \subseteq O_n = H^t_n \cap \overline{W_n} \cap \overline{W'_n}$. Assume further that we are given $G(\alpha, t) = (H_t, P_t, W_t, W'_t, \delta_t)$ for any $t \in \delta_n(A_n)$. Now define $G_{n+1} := (H_{n+1}, P_{n+1}, W_{n+1}, W'_{n+1}, \delta_{n+1})$ where

$$\begin{aligned}
H_{n+1} &:= H_n \cup A_n\langle 0\rangle \cup \textstyle\bigcup_{\substack{q\in A_n \\ \delta_n(q)=t}} q\langle 1\rangle H_t \\
\delta_{n+1}(q) &:= \begin{cases} \delta_n(q) & \text{if } q \in H_n \\ \delta_n(r) & \text{if } q = r\langle 0\rangle \text{ and } r \in A_n \\ \delta_t(s) & \text{if } q = r\langle 1\rangle s,\ r \in A_n \text{ and } \delta_n(r) = t \end{cases} \\
P_{n+1} &:= P_n \cup A_n \cup \textstyle\bigcup_{\substack{q\in A_n \\ \delta_n(q)=t}} q\langle 1\rangle P_t \\
W_{n+1} &:= W_n \cup \textstyle\bigcup_{\substack{q\in A_n \\ \delta_n(q)=t}} q\langle 1\rangle W_t \\
W'_{n+1} &:= W'_n \cup \textstyle\bigcup_{\substack{q\in A_n \\ \delta_n(q)=t}} q\langle 1\rangle W'_t \\
A_{n+1} &:= \textstyle\bigcup_{\substack{q\in A_n \\ \delta_n(q)=t}} q\langle 1\rangle (H^t_t \cap \overline{W_t} \cap \overline{W'_t})
\end{aligned}$$

   Note that δ_{n+1} is well-defined: It is fully defined on H_{n+1}, and since $A_n \subseteq O_n$, the three clauses are mutually exclusive. Also observe that $A_{n+1} \subseteq O_{n+1} = H_{n+1} \cap \overline{W_{n+1}} \cap \overline{W'_{n+1}}$, and $W_{n+1} \cap W'_{n+1} = \emptyset$. Note that iterating this procedure will usually lead to infinite runs which should be winning for Demon, i.e. Angel must stop playing α eventually in order to win. Let Q^∞ be the set of infinite sequences such that all its finite initial subsequences are in $\bigcup_n H_n$. We then define

$$G(\alpha^*, s) := (Q^\infty \cup \bigcup_n H_n,\ \bigcup_n P_n,\ \bigcup_n W_n,\ Q^\infty \cup \bigcup_n W'_n,\ \bigcup_n \delta_n)$$

5. An Example: Beer or Quiche?

The following situation is an adaptation of the *Beer or Quiche?* game as introduced in [8]. The original game has been changed into a strictly competitive win/lose game for a better fit with our present framework. This modification is not crucial however since the main purpose of this section is to show how game webs and game logic can be used to model game situations with incomplete information.

The situation we are considering is the following: Two men, Al and Dick, meet in a bar. Dick is not particularly fond of Al and considers challenging him for a fight. Al can either drink a beer (and get drunk) or eat quiche. Subsequently, Dick has to decide whether or not to fight Al. What complicates the situation is that Dick is unsure about Al's strength. Considering Al's preferences, we assume that if he is weak, he prefers not to fight, whereas if he is strong, he prefers to fight if and only if he is sober. We assume the game is strictly competitive, so Dick's preferences are the opposite: If Al is weak, Dick prefers to fight him, but if Al is strong, Dick only wants to fight him if Al's reaction times have deteriorated due to alcohol. The situation is summarized by the game tree in figure 3.

We can model this situation with the game web pictured in figure 4. Four atomic games are under considerations, Al can drink beer (b), Al can eat quiche (q), Dick can fight (f) and Dick can do nothing

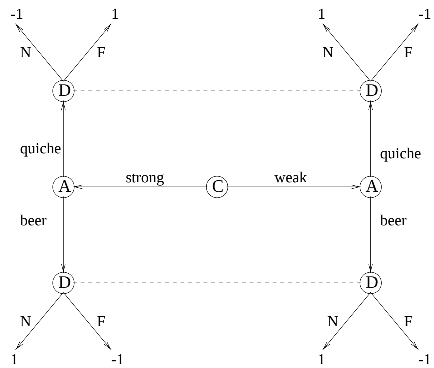

Figure 3: *Beer or Quiche?* with the payoffs for Al (Angel)

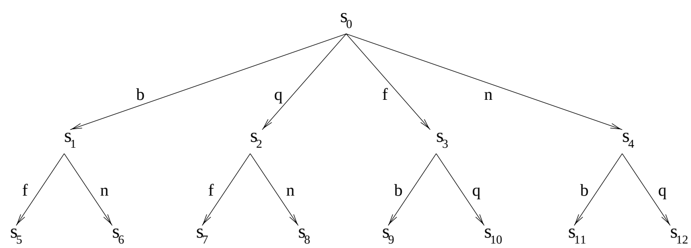

Figure 4: The game web for *Beer or Quiche?*

(n). These atomic untyped games are of a very simple kind, they are just deterministic actions for one of the players. The atomic untyped game corresponding to Al drinking beer in state s_0 is simply $(\{\langle\rangle, \langle 0\rangle\}, \{\langle\rangle\}, \emptyset, \emptyset, \delta)$ where $\delta(\langle\rangle) = s_0$ and $\delta(\langle 0\rangle) = s_1$. Similarly for the other atomic games. Since these atomic games are simply deterministic actions, figure 4 represents these games simply by arrows. After one of these actions has taken place in s_0, the resulting state again allows for actions being taken. Note that the modeling we have chosen in figure 4 is very general: It allows e.g. for the possibility that Dick decides to fight before waiting to see what Al eats/drinks. Also observe that technically, figure 4 only depicts part of the web $\mathcal{I}$, since we assumed in our definition of game webs that every game can be played everywhere.

Consider now the scenario described by the original story: First Al chooses whether to drink beer or eat quiche, then Dick chooses whether to fight or not. The expression $(b \cup q); (f \cap n)$ is a translation of this scenario into the language of GL, and figure 5 represents $G((b \cup q); (f \cap n), s_0)$.

The game depicted does not contain any information about preferences yet, it is untyped. Let us consider various types of players which can be represented by propositional letters. Given the earlier description of the players' preferences, let

$$S^{\mathcal{I}} := \{s_6, s_7, s_9, s_{10}\}$$

be the set of states which a strong Al prefers. Similarly, let

$$W^{\mathcal{I}} := \{s_6, s_8, s_{11}, s_{12}\}$$

be the set of states which a weak Al prefers. If Al is of the strong type (and hence Dick prefers $\neg S$), Dick has a winning strategy in the (now typed) game, i.e. $\mathcal{I}, s_0 \models [(b \cup q); (f \cap n)]\neg S$. Similarly, Dick

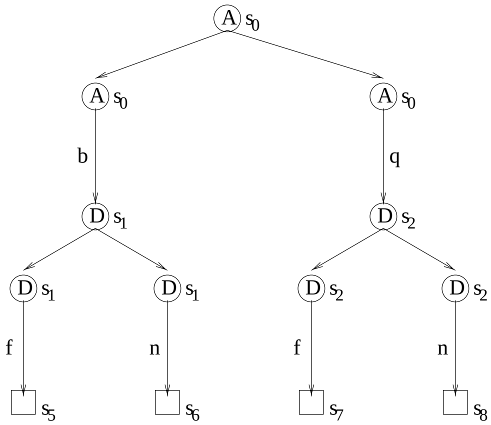

Figure 5: The untyped *Beer or Quiche?* game: $G((b \cup q); (f \cap n), s_0)$

can win against a weak Al, i.e. $\mathcal{I}, s_0 \models [(b \cup q); (f \cap n)]\neg W$, but Dick cannot win if he does not know whether Al is weak or strong,

$$\mathcal{I}, s_0 \not\models [(b \cup q); (f \cap n)](\neg W \wedge \neg S)$$

Also note that Al has no winning strategy no matter whether he is weak or strong:

$$\mathcal{I}, s_0 \models \neg\langle(b \cup q); (f \cap n)\rangle W \ \wedge \ \neg\langle(b \cup q); (f \cap n)\rangle S$$

We have seen that Dick cannot guarantee a win in the game if he does not know whether Al is weak or strong. The present framework allows us to ask a further question: Is there a game in which Dick *can* win independent of Al's type? The answer is yes, e.g. for $\gamma := b; (f \cap n)$, since

$$\mathcal{I}, s_0 \models [b; (f \cap n)](\neg W \wedge \neg S)$$

so if the owner of the bar only serves beer, Dick will have a winning strategy. This example shows that game webs and a formal language to construct games allow us to address questions of *game synthesis*: Given a game web which specifies which atomic games (i.e. actions) are available at every state, is there a complex game which gives one of the players a winning strategy against all opponents of certain types? This game synthesis problem should be seen as a particularly simple instance of the more general implementation problem (see [16]). The emphasis here is not on implementing a certain equilibrium outcome, but rather on implementing a winning strategy for one of the players against various types of opponents. Classic implementation problems such as Solomon's judgment can at least be reformulated in these terms, [1] though they may have no winning strategy implementation.

6. Metatheory

6.1 Game Equivalence

The language which *GL* provides to describe untyped games is very restricted. The formulas $\langle\gamma\rangle\varphi$ and $[\gamma]\varphi$ express the existence of a φ-strategy for one of the players in the game γ, other properties of γ cannot be expressed. This observation induces an equivalence notion for untyped games. We shall

[1] In the example of Solomon's judgment, we need a game in which Solomon has a winning strategy against two types of opponent, the real and the false mother; Solomon wins if he has given the correct wise judgment.

consider two untyped games as equivalent iff no player type can distinguish the two games in terms of winning strategies. Formally, call two untyped games G_1 and G_2 over the same set of states S *equivalent* (notation $G_1 \equiv G_2$) iff for all $X \subseteq S$, (1) Angel has an X-strategy in G_1 iff she has one in G_2 and (2) similarly for Demon. Note that if G_1 and G_2 are determined, the first condition suffices.

This definition of equivalence can be lifted to syntactic game expressions as well. Given $\gamma_1, \gamma_2 \in \Gamma$, call γ_1 and γ_2 *equivalent* (notation $\gamma_1 \equiv \gamma_2$) iff for all game webs $\mathcal{I}$ with universe S and for all $s \in S$, $G(\gamma_1, s) \equiv G(\gamma_2, s)$. It will be shown in section 6.3 that all *GL*-games are determined, so equivalently, $\gamma_1 \equiv \gamma_2$ iff $\models \langle\gamma_1\rangle p \leftrightarrow \langle\gamma_2\rangle p$.

The reader can check that the following three distribution principles hold:

$$\begin{aligned} (\alpha \cup \beta) \cap \gamma &\equiv (\alpha \cap \gamma) \cup (\beta \cap \gamma) \\ (\alpha \cup \beta); \gamma &\equiv (\alpha; \gamma) \cup (\beta; \gamma) \\ (\alpha; \beta)^d &\equiv \alpha^d; \beta^d \end{aligned}$$

The second and third equivalence could be called *literal equivalences*, for the untyped games which the game expressions give rise to are identical in every model. The first equivalence on the other hand is not literal, given that the untyped games $(\alpha \cup \beta) \cap \gamma$ and $(\alpha \cap \gamma) \cup (\beta \cap \gamma)$ differ in which player gets to move first.

As an example of an invalid distribution principle, observe that the game expression $\alpha; (\beta \cup \gamma)$ is not equivalent to $(\alpha; \beta) \cup (\alpha; \gamma)$.

6.2 An Alternative Semantics

Given the rather coarse notion of equivalence implicit in *GL*, the semantics of *GL* seems more complex than needed. Since all the information needed is which angelic types have winning strategies, an alternative semantic approach could simply model untyped games as relations between states and winning angelic types. For this project to succeed however, the winning types of complex games must be definable purely in terms of the winning types of its subgames, i.e. not using any additional information about the subgames. The semantics presented below succeeds in doing just that. It is the semantics originally proposed for *GL* in [18], a modal neighborhood semantics which associates with every game a neighborhood relation N between states and sets of states. In modal logic, neighborhood semantics is frequently used for non-normal modal logics, i.e. modal logics weaker than K (see [7]).

A *neighborhood model* $\mathcal{M} = (S, \{N_g | g \in \Gamma_0\}, V)$, consists of a set of states S, a valuation $V : \Phi_0 \to \mathcal{P}(S)$ for the propositional letters and a set of neighborhood relations $N_g \subseteq S \times \mathcal{P}(S)$ which are monotonic, i.e. sN_gX and $X \subseteq X'$ imply sN_gX'. The idea is that sN_gX holds whenever Angel has an X-strategy in game g. Truth in $\mathcal{M}$ according to the neighborhood semantics shall be denoted using $\vdash$. As usual, $\mathcal{M}, s \vdash \varphi$ is defined by induction on φ, and the cases for atomic propositions and boolean connectives are no different from the previous semantics. For $\langle\gamma\rangle\varphi$, we say that

$$\mathcal{M}, s \vdash \langle\gamma\rangle\varphi \text{ iff } sN_\gamma\varphi^{\mathcal{M}}$$

where in the present context $\varphi^{\mathcal{M}} := \{s \in S | \mathcal{M}, s \vdash \varphi\}$. As usual in modal logic (but different from the earlier semantics), we define $[\gamma]\varphi$ as $\neg\langle\gamma\rangle\neg\varphi$. The relation $N_\gamma \subseteq S \times \mathcal{P}(S)$ is now defined inductively for non-atomic programs γ. Let $N_\gamma(Y) := \{s \in S | sN_\gamma Y\}$. Then

$$\begin{aligned} N_{\alpha;\beta}(Y) &:= N_\alpha(N_\beta(Y)) \\ N_{\alpha\cup\beta}(Y) &:= N_\alpha(Y) \cup N_\beta(Y) \\ N_{\varphi?}(Y) &:= \varphi^{\mathcal{M}} \cap Y \\ N_{\alpha^d}(Y) &:= \overline{N_\alpha(\overline{Y})} \\ N_{\alpha^*}(Y) &:= \mu X.(Y \cup N_\alpha(X) \subseteq X) \end{aligned}$$

In this definition, $\mu X.f(X)$ denotes the smallest set X satisfying $f(X)$. It can be shown that monotonicity of the N_g-relations is preserved under the program connectives, so this fixpoint always exists.

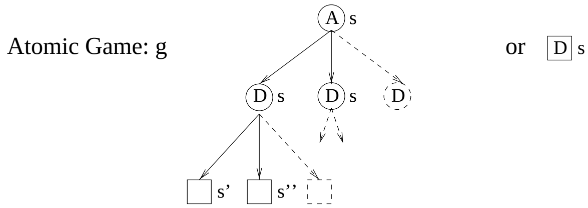

Figure 6: Constructing an atomic game from a given relation N_g

We say that a game model $\mathcal{I} = (S, \{G(g,s)|g \in$, $_0$ and $s \in S\}, V)$ and a neighborhood model $\mathcal{M} = (S, \{N_g|g \in$, $_0\}, V)$ *correspond* iff for any $s \in S$ and $X \subseteq S$,

$$sN_gX \iff \text{Angel has an } X\text{-strategy in } G(g,s) \tag{6.1}$$

Note that since the two players cannot both have winning strategies for complementary winning positions, and since every $G(g,s)$ was assumed to be determined, corresponding models also have the following property:

$$\text{not } sN_gX \iff \text{Demon has a } \overline{X}\text{-strategy in } G(g,s) \tag{6.2}$$

Given any game model, we can define a corresponding neighborhood model simply by using (6.1) to define the atomic neighborhood relations. Conversely, given any neighborhood model, we can define a corresponding game model as follows: Given the monotonic relation N_g and a state s, let $\Pi_s := \{X \subseteq S | sN_gX\}$. We define $G(g,s)$ as shown in figure 6.

In case $\Pi_s = \emptyset$, let $G(g,s) := (\{\langle\rangle\}, \emptyset, \emptyset, \{\langle\rangle\}, \{(\langle\rangle, s)\})$. Otherwise, we allow Angel to choose a possible set $X \in \Pi_s$. Next, Demon may choose an element $x \in X$. δ maps each X to state s, while each $x \in X$ is mapped to x itself. In case $X = \emptyset$, Angel wins immediately. We will not write out the formal definition of this game, but it can be checked that the game $G(g,s)$ constructed in this way is determined and stands in the required relationship (6.1) to N_g.

The following theorem contains the main technical result of this paper: If we choose to identify corresponding models, the two semantics are equivalent. In other words, atomic correspondence extends to composite games and formulas.

Theorem 1 (Equivalence) *Let $\mathcal{I}$ be a game model and $\mathcal{M}$ be a corresponding neighborhood model. Then for all formulas φ and games γ we have*

(i) $\mathcal{M}, s \vdash \varphi$ *iff* $\mathcal{I}, s \models \varphi$
(ii) $sN_\gamma X \Leftrightarrow$ *Angel has an X-strategy in $G(\gamma, s)$*
(iii) *not* $sN_\gamma X \Leftrightarrow$ *Demon has a $\overline{X}$-strategy in $G(\gamma, s)$*

6.3 Determinacy and Complexity of Games

Given , $_0$, Φ_0 and a game model $\mathcal{I}$ with universe S, define the *class of GL-games* as $G^{\mathcal{I}}_{GL} := \{G(\gamma,s)|\gamma \in$, and $s \in S\}$, i.e. as the class of all untyped games which can be constructed by means of the game constructions in the model.

When isolating a certain class of finite and infinite games, one of the most basic game-theoretic questions concerns determinacy: Is it always the case that one of the two players has a winning strategy? The answer to this question is an easy corollary to the preceding theorem:

Corollary 2 *For every $\mathcal{I}$, all games in $G^{\mathcal{I}}_{GL}$ are determined.*

In Hintikka's game-semantics for first-order logic [12], it turned out that the law of excluded middle was just determinacy in disguise. This link between a logical principle and a basic game-theoretic property has its analogue in *GL*, namely in the Box-Diamond duality $\langle\gamma\rangle\varphi \leftrightarrow \neg[\gamma]\neg\varphi$. The validity of this principle is equivalent to the validity of (1) $\neg(\langle\gamma\rangle\varphi \wedge [\gamma]\neg\varphi)$ and (2) $\langle\gamma\rangle\varphi \vee [\gamma]\neg\varphi$. Formula (1) is valid by our definition of winning strategies, formula (2) states determinacy, so it is valid by the previous corollary. As shall be shown subsequently, proving determinacy directly is not a trivial task, so giving a non-game-theoretic semantics such as neighborhood semantics can be seen (among other things) as a technique to establish determinacy.

To prove determinacy directly using game-theoretic techniques, one has a number of standard results at one's disposal, such as Zermelo's theorem [20] and the Gale-Stewart theorem [10]. Note first that Zermelo's result does not apply to *GL*-games, since it states that *finite* games are determined, i.e. games where all runs are finite. Since we introduced infinite runs for iteration, Zermelo's theorem cannot be applied.

As for the Gale-Stewart theorem, we shall show that it is also too weak to cover all *GL*-games. Let $G = (G, P_A, W_A, W_D, \delta)$ be an untyped game. $Q \subseteq H^\infty$ is a *basic open* subset of H^∞ iff there is some integer n such that

$$\forall q \in Q\, \forall h \in H^\infty : \quad q|n = h|n \Rightarrow h \in Q$$

where $q|n$ denotes the sequence consisting of the first n elements of q. $Q \subseteq H^\infty$ is an *open* subset of H^∞ iff it is the countable union of basic open subsets of H^∞. Similarly, $Q \subseteq H^\infty$ is a *closed* subset of H^∞ iff it is the countable intersection of basic open subsets of H^∞. Finally, G is basic open, open, or closed iff W_A or W_D is respectively a basic open, open, or closed subset of H^∞. A class of untyped games has one of these three properties iff all games in the class have the property. While originally not formulated for the games we are interested in here, the Gale-Stewart theorem can be applied to untyped games as well:

Theorem 3 (Gale-Stewart) *Open games are determined.*

It is not difficult however to think of an example of a game which is neither open nor closed, demonstrating that also the Gale-Stewart theorem is too weak to establish the determinacy of all *GL*-games.

Theorem 4 *For all $\mathcal{I}$, $G^{\mathcal{I}}_{GL}$ is neither open nor closed.*

For infinite games, Martin proved in [15] that all *Borel games* are determined. This result subsumes the result of Gale-Stewart, and given certain assumptions about the atomic games (e.g. only a countable number of undecided positions), we conjecture that all *GL*-games are Borel, and hence determined. A more detailed analysis could probably establish an upper bound on the Borel rank of *GL*-games.

6.4 Study of Game Constructions

While the operations provided by *GL* (such as composition and union) seem natural enough, there are other operations one might consider such as parallel execution of games as studied e.g. in the game-theoretic analysis of linear logic [5, 2], or an alternative version of iteration as shown in figure 7.

In this version of iteration, Angel has to decide in advance how often she wants to play α. A winning strategy for Angel in this alternative version of iteration is also a winning strategy for the standard game α^*, but the converse is not true. If we replaced standard iteration with this alternative iteration, all games would be finite, and determinacy would simply follow from Zermelo's theorem.

The variety of possible game operations suggests a natural metatheoretic question: Can we find an "interesting" characterization of the class of untyped games which can be constructed (in a given

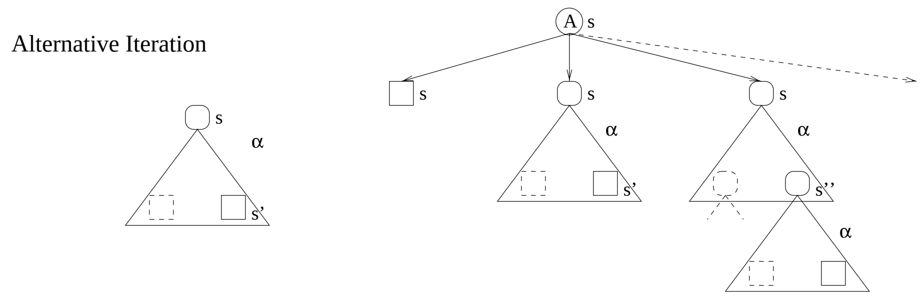

Figure 7: An alternative version of iteration

model) by means of a certain set of game operations? Or starting at the other end, given a class of untyped games we are interested in, what game operations are sufficient to construct every game in that class? That such an investigation is feasible is shown in [19]. Working in models where the atomic games can be expressed by binary relations over states, it is shown that *GL*-games without iteration are all definable by means of a formula of first-order logic. Conversely, it is shown that any formula of first-order logic which defines an untyped game can be obtained using only game operations of iteration-free *GL*. In other words, if we are only interested in games which can be described in first-order logic, the game operations of union, composition, test and dual are sufficient to construct every game. This means that within first-order logic, no "new" game operations can be discovered.

6.5 Axiomatic Reasoning about Games

When introducing a logical language together with a semantics for that language, one of the standard exercises for the logician is to provide a set of axioms and inference rules which together allow one to derive all and only the validities of the logic. This exercise has been carried out in [18] with partial success: While there is a clear candidate for such an axiomatization, a completeness proof exists only for *GL* without the dual-operator and for *GL* without iteration. None the less, the same paper contains an argument showing that *GL* is decidable, i.e. there is an algorithm which can determine of any formula whether or not it is true in some model in *GL*. While this algorithm takes time exponential in the length of the formula, it is not known whether better algorithms exist.

Besides the theoretical interest in knowing the complexity of reasoning about untyped games, results regarding axiomatization and complexity are also of practical relevance. Looking back at the example of section 5, such results can also provide a (naive) method for solving the implementation problem in its game synthesis form, provided our target games do not contain iteration: If we are looking for a game γ such that every model satisfying our assumptions Δ about atomic games also satisfies $[\gamma](\neg W \wedge \neg S)$, simply enumerate the possible games γ and check whether $\Delta \models [\gamma](\neg W \wedge \neg S)$ holds. The result will be a game which guarantees Dick to win against both a weak and a strong Al, for example $b; (f \cap n)$.

7. Conclusions and Future Work

We described an interaction model (game webs) based on states of the world at which different games can be played. These games associate a state of the world with each position of the game, keeping track of how the game changes the state of the world. Since different games can be played at each state, implementation questions can also be addressed within this framework, where the focus is not on an equilibrium analysis but rather on a winning strategy analysis. The games involved are untyped, which allows for reasoning with incomplete information, by reasoning about winning strategies against different types of players. Viewed differently, untyped games combine extensive games, extensive game forms and everything in between in one notion. Finally, the most crucial aspect of this approach is to consider games as objects which are constructed by means of certain operations. This view leads to a whole set of new questions concerning game operations, game equivalence, and so on.

Next to research on metatheoretic questions, extensions of the basic language of GL seem desirable to increase the applicability of GL as a game-theoretic modeling tool. We shall now briefly discuss two such extensions.

First, the restriction to zero-sum win/lose games limits the applicability of the present system. In principle, it is possible to model a wider range of games even in the present framework by simply introducing propositional letters for the different payoffs of a player. As an example, let p denote player 1 getting payoff 3, and let q denote player 2 getting payoff 5. Then $\langle\gamma\rangle(p \wedge q)$ expresses that player 1 has a strategy for reaching a state with the payoff profile $(3, 5)$. Especially when combined with different types of players however, such a solution may not be very appealing conceptually, and it may be more natural to extend the model by preference relations over states.

Second, one could add operators which allow one to construct games of imperfect information. A candidate operation to add would be $\cup^i$, where i indicates the information set to which the game position belongs. Such an extension is difficult to handle using neighborhood semantics, for given the neighborhood relations of γ_1 and γ_2, what should the relation for $\gamma_1 \cup^i \gamma_2$ be? The answer depends on the information sets of the two subgames, but the neighborhood relations do not contain any information about information sets, nor is it clear how to enrich them with such information. Untyped games on the other hand contain enough information about the game structure, so that we can simply assign the game positions to information sets. Making the move to games of imperfect information would result in a loss of determinacy, yielding a rather unusual modal logic in which there will be games γ for which $\langle\gamma\rangle\varphi \leftrightarrow \neg[\gamma]\neg\varphi$ is not valid anymore.

Appendix

Proof of theorem 1: By simultaneous induction on programs and formulas. For (i), the case for atomic formulas holds by definition and the boolean cases are immediate. For $\varphi = [\gamma]\psi$ and $\varphi = \langle\gamma\rangle\psi$, we use the induction hypothesis (i) for ψ and (ii) and (iii) for γ. The work lies in proving (ii) and (iii) by induction on γ. Note that it is sufficient to prove (ii) and (iii) from left to right given the definition of winning strategies.

In the proof below, given a game $G_k = (H_k, P_k, W_k, W'_k, \delta_k)$, O_k will denote the set of open terminal runs of G_k, i.e. $O_k := H^t_k \cap \overline{W_k} \cap \overline{W'_k}$.

For an atomic program a, (ii) and (iii) hold by the definition of correspondence.

1. Test $\varphi?$: $sN_{\varphi?}X$ iff $\mathcal{I}, s \models \varphi$ (by induction hypothesis (i))and $s \in X$. Hence, doing nothing is an X-strategy for Angel in $G(\varphi?, s)$.

   On the other hand, if $\neg sN_{\varphi?}X$, $\mathcal{I}, s \not\models \varphi$ or $s \notin X$. In both cases, doing nothing is a $\overline{X}$-strategy for Demon.

2. Union $\alpha \cup \beta$: By induction hypothesis, Angel has an X-strategy for $G(\alpha, s)$ or $G(\beta, s)$. Suppose w.l.o.g. that Angel has an X-strategy σ_0 for $G(\alpha, s)$. So when playing $G(\alpha \cup \beta, s)$, Angel can choose α and continue playing according to σ_0. Formally, any σ with $\sigma(\langle\rangle) = \langle 0\rangle$ and $\sigma(\langle 0\rangle q) = \sigma_0(q)$ is an X-strategy for Angel in $G(\alpha \cup \beta, s)$.

   On the other hand, if not $sN_{\alpha\cup\beta}X$, Demon has $\overline{X}$-strategies σ_0 and σ_1 for respectively $G(\alpha, s)$ and $G(\beta, s)$, and then no matter how Angel chooses at the beginning of $G(\alpha \cup \beta, s)$, Demon can win. Hence, σ defined by $\sigma(\langle 0\rangle q) = \sigma_0(q)$ and $\sigma(\langle 1\rangle q) = \sigma_1(q)$ is a $\overline{X}$-strategy for Demon in $G(\alpha \cup \beta, s)$.

3. Composition $\alpha; \beta$: Suppose $s \in N_{\alpha;\beta}(X) = N_\alpha(N_\beta(X))$. Then Angel has a $N_\beta(X)$-strategy σ_α in $G(\alpha, s) = (H_\alpha, P_\alpha, W_\alpha, W'_\alpha, \delta_\alpha)$, that is $\delta_\alpha(O^{\sigma_\alpha}_\alpha) \subseteq N_\beta(X)$. So by induction hypothesis, for any $t \in \delta_\alpha(O^{\sigma_\alpha}_\alpha)$, Angel has an X-strategy σ_t in $G(\beta, t)$. So after playing according to σ_α, Angel

will either win or reach a position $q \in O_\alpha$ where she can start playing according to σ_t (where $\delta_\alpha(q) = t$) which will be X-winning. Thus, σ satisfying

$$\begin{aligned} \sigma(q) &= \sigma_\alpha(q) && \text{for } q \in H_\alpha \\ \sigma(qr) &= \sigma_t(r) && \text{for } q \in O_\alpha^{\sigma_\alpha} \text{ and } \delta_\alpha(q) = t \end{aligned} \tag{7.1}$$

provides an X-strategy for Angel in $G(\alpha; \beta, s)$.

Conversely, suppose that Demon has a $\overline{N_\beta(X)}$-strategy σ_α in $G(\alpha, s)$. Then by induction hypothesis, for any $t \in \delta_\alpha(O_\alpha^{\sigma_\alpha})$, Demon has an $\overline{X}$-strategy σ_t in $G(\beta, t)$. Then σ (as defined in (7.1)) is a $\overline{X}$-strategy for Demon in $G(\alpha; \beta, s)$.

4. Dual α^d: $sN_{\alpha^d}X$ iff not $sN_\alpha\overline{X}$ which by induction hypothesis implies that Demon has an X-strategy σ in $G(\alpha, s)$, in which case σ is also an X-strategy for Angel in $G(\alpha^d, s)$.

   Analogously for (iii).

5. Iteration α^*: Let Z be the set of all states s such that Angel has an X-strategy in $G(\alpha^*, s)$. We need to show that $\mu Y.(X \cup N_\alpha(Y) \subseteq Y) \subseteq Z$. For this, it is sufficient to show that $X \cup N_\alpha(Z) \subseteq Z$. If $s \in X$, then Angel clearly has an X-strategy, namely to stop right away. In that case, σ with $\sigma(\langle\rangle) = \langle 0\rangle$ is an X-strategy. Otherwise, if $s \in N_\alpha(Z)$, then Angel has a Z-strategy σ_α in $G(\alpha, s) = (H_\alpha, P_\alpha, W_\alpha, W'_\alpha, \delta_\alpha)$, by induction hypothesis. But then Angel can simply choose to do α at s playing according to σ_α. This will lead to states t where Angel has an X-strategy σ_t in $G(\alpha^*, t)$. Formally, σ defined as

$$\begin{aligned} \sigma(\langle\rangle) &= \langle 1\rangle \\ \sigma(\langle 1\rangle q) &= \sigma_\alpha(q) && \text{for } q \in H_\alpha \\ \sigma(\langle 1\rangle qr) &= \sigma_t(r) && \text{where } q \in O_\alpha^{\sigma_\alpha} \text{ and } \delta_\alpha(q) = t \end{aligned}$$

   is an X-strategy for Angel in $G(\alpha^*, s)$, so $s \in Z$.

   Finally, let us sketch the proof of (iii) for iteration. Note first that $\overline{N_{\alpha^*}(X)} = \nu Y.(Y \subseteq \overline{X} \cap \overline{N_\alpha(\overline{Y})})$, i.e. the greatest set Y satisfying the relation given. Let Z be the set of states s where Demon has a $\overline{X}$-strategy in $G(\alpha^*, s)$. It suffices to show that for all $Y \subseteq S$, if $Y \subseteq \overline{X} \cap \overline{N_\alpha(\overline{Y})}$ then $Y \subseteq Z$. So assume $s \in Y$, and hence $s \in \overline{X} \cap \overline{N_\alpha(\overline{Y})}$. We shall show by induction on n that Demon has a Y-strategy σ_n in $G_n(\alpha, s)$, as defined on page 7. For $n = 0$ it suffices that we have $s \in Y$, so doing nothing does the job. For the inductive step, assume that Demon has a Y-strategy σ_n in G_n. Now Demon can play G_{n+1} as follows: While in G_n, play according to σ_n. At any $q \in O_n^{\sigma_n}$, $\delta_{n+1}(q) = \delta_n(q) \in Y$. On the other hand, for $q \in A_n^{\sigma_n}$, $\delta_{n+1}(q\langle 0\rangle) = \delta_{n+1}(q\langle 1\rangle) = \delta_n(q) = t \in Y \subseteq \overline{N_\alpha(\overline{Y})}$, so by induction hypothesis, Demon has a Y-strategy σ_t in $G(\alpha, t)$ according to which she can play if Angel chooses to play α again. This yields the following Y-strategy σ_{n+1} for G_{n+1}:

$$\begin{aligned} \sigma_{n+1}(q) &= \sigma_n(q) && \text{for } q \in H_n \\ \sigma_{n+1}(q\langle 1\rangle r) &= \sigma_t(r) && \text{where } q \in A_n^{\sigma_n} \text{ and } \delta_n(q) = t \end{aligned}$$

   Since the infinite runs created by this iterative process are winning for Demon anyway, $\bigcup_n \sigma_n$ is a Y-strategy (and hence also an $\overline{X}$-strategy) for $G(\alpha^*, s)$, so $s \in Z$. □

Proof of theorem 4: Let $\mathcal{I}$ be an arbitrary game model. Let $g_d := \top?^*; \bot?$ and let

$$g := (\top?^{*d}; g_d) \cup (\top?^*; {g_d}^d)$$

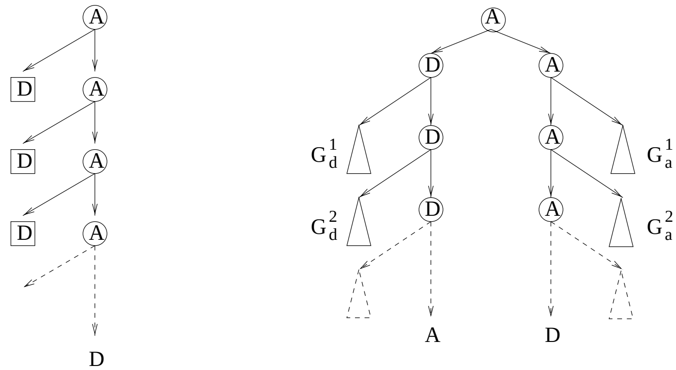

Figure 8: Game G_d on the left and G on the right

Let $G_d := G(g_d, s)$, $G_a := G({g_d}^d, s)$ and $G := G(g, s)$ for an arbitrary state s. Note that every position of these games is associated with state s, and that there are no undecided final positions. Figure 8 shows games G_d and G. As shown, we have numbered the different copies of the subgames G_a/G_d we plugged in from top to bottom. Let us refer to the infinite branch in game G_d^i (which is winning for Demon) as d^i, and to the infinite branch in game G_a^i (which is winning for Angel) as a^i. The two remaining infinite branches will be referred to as a^ω and d^ω depending on the winner of the branch. We claim that $G = (H, P_A, W_A, W_D, \delta)$ is neither open nor closed.

Suppose by reductio that $W_A = \{a^1, a^2, \ldots, a^\omega\}$ is open, i.e. $W_A = \bigcup_i Q_i$ where all Q_i are basic open subsets of H^∞. Then the infinite branch a_ω must be in some Q_j. But since Q_j was assumed to be a basic open subset of H^∞, there is some k such that all infinite branches which agree with a_ω up to k must also be in Q_j. Given the construction of G, this means that there must be some infinite branch $d^i \in Q_j$, a contradiction since $d^i \notin W_A$. By the same reasoning, $W_D = \{d^1, d^2, \ldots, d^\omega\}$ is not open.

Suppose on the other hand that W_A is closed, i.e. $W_A = \bigcap_i Q_i$ where all Q_i are basic open subsets of H^∞. This means that for every Q_i, $W_A \subseteq Q_i$. Since every Q_i contains every a^j for $j < \omega$, d^ω must also be in every Q_i and hence in W_A, a contradiction. Similarly for W_D. □

References

1. Martin Abadi, Leslie Lamport, and Pierre Wolper. Realizable and unrealizable specifications of reactive systems. In Giorgio Ausiello et al., editors, *Automata, languages and programming*, pages 1–17. Springer, 1989.
2. Samson Abramsky and Radha Jagadeesan. Games and full completeness for multiplicative linear logic. *Journal of Symbolic Logic*, 59:543–574, 1994.
3. Geir Asheim. On the epistemic foundations of backward induction. In Harrie de Swart, editor, *Logic, Game Theory and Social Choice*, pages 8–23. Tilburg University Press, 1999.
4. Ralph-Johan Back and Joakim von Wright. *Refinement Calculus - A systematic introduction*. Springer, 1998.
5. Andreas Blass. A game semantics for linear logic. *Annals of Pure and Applied Logic*, 56:183–220, 1992.
6. Giacomo Bonanno. The logic of rational play in games of perfect information. *Economics and Philosophy*, 7:37–65, 1991.
7. Brian Chellas. *Modal Logic - An Introduction*. Cambridge University Press, 1980.
8. I.K. Cho and D.M. Kreps. Signaling games and stable equilibria. *Quarterly Journal of Economics*, 102(2):179–221, 1987.
9. Thorsten Clausing. Behavioural vs. habitual rationality and backward induction. In Harrie de Swart, editor, *Logic, Game Theory and Social Choice*, pages 46–56. Tilburg University Press, 1999.
10. David Gale and F.M. Stewart. Infinite games with perfect information. In *Contributions to the Theory of Games, II*, volume 28 of *Annals of Mathematics Studies*, pages 245–266. 1953.
11. David Harel. Dynamic logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume II. Reidel, 1984.
12. J. Hintikka and G. Sandu. Game-theoretical semantics. In J. van Benthem and A. ter Meulen, editors, *Handbook of Logic and Language*, pages 361–410. Elsevier, 1993.
13. M. Kaneko and T. Nagashima. Game logic and its applications I. *Studia Logica*, 57:325–354, 1996.
14. Dexter Kozen and Jerzy Tiuryn. Logics of programs. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B. MIT Press, 1990.

15. Donald Martin. Borel determinacy. *Annals of Mathematics*, 102:363–371, 1975.

16. John Moore. Implementation, contracts, and renegotiation in environments with complete information. In Jean-Jacques Laffont, editor, *Advances in Economic Theory: Sixth World Congress*, volume 1. Cambridge University Press, 1992.

17. Martin Osborne and Ariel Rubinstein. *A Course in Game Theory*. MIT Press, 1994.

18. Rohit Parikh. The logic of games and its applications. In Marek Karpinski and Jan van Leeuwen, editors, *Topics in the Theory of Computation*, volume 24 of *Annals of Discrete Mathematics*. Elsevier, 1985.

19. Marc Pauly. From programs to games: Invariance and safety for bisimulation. In Peter Clote and Helmut Schwichtenberg, editors, *Computer Science Logic*, volume 1862 of *LNCS*. Springer, 2000.

20. Ernst Zermelo. Über eine Anwendung der Mengenlehre auf die Theorie des Schachspiels. In E. Hobson and A. Love, editors, *Proceedings of the 5th Intl. Congress of Mathematicians*, volume II, pages 501–504. Cambridge University Press, 1913.