Tableaux and Automata for Parikh's Game Logic

Govert van Drimmelen
University of the Witwatersrand

Infinity in Logic and Computation – November 2007

Background to Game Logic

Theories of interaction

- Leonid Hurwicz, Eric Maskin and Roger Myerson were awarded the Nobel prize in economics for 'having laid the foundations of **mechanism design theory**'.
- Rohit Parikh promotes the study of **social software** – describing and analyzing social procedures.

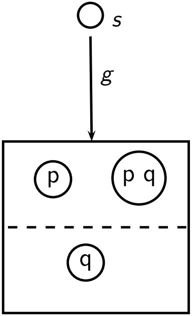
Game Logic, introduced by Parikh in 1983 and 1985, is motivated as a formalism to reason about such models of interaction.

Logical aspects of Game Logic

- Characterization as a modal logic
- Decision procedures and axiomatization
- Expressiveness – especially compared to μ-calculus

Game models and atomic games

We consider a discrete state-based model of two-player games.
The universe is represented by a set of abstract states with valuation of atomic propositions at each state.
An **atomic game** describes how the choices of the players lead to a successor state.



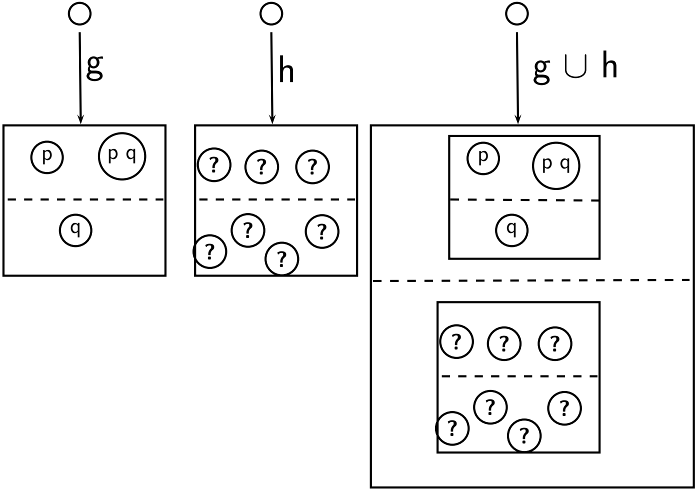
$$\mathcal{M}, s \models \langle g \rangle p$$

player 1 has a choice in game g
that ensures an outcome satisfying p.

$$\neg\langle g\rangle\neg q \equiv [g]\, q$$

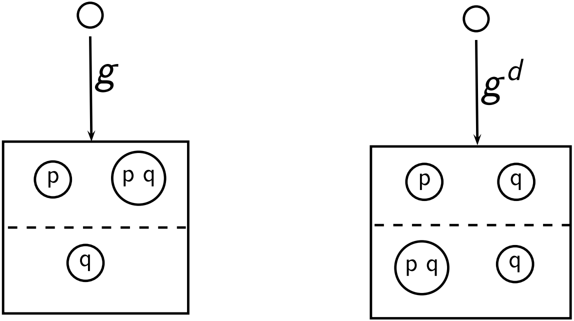
Game operations – Choice

Player 1 chooses whether game *g* or game *h* should be played.



Game operations – dual

Players reverse roles – the choices for player 1 are now the sets that player 2 could ensure in g.



$\langle g \rangle p$ $\qquad$ $\langle g^d \rangle q \equiv \neg \langle g \rangle \neg q \equiv [g]\, q$

Game operations

Game Logic syntax

- Games $\gamma := g \mid \gamma^d \mid \gamma_1 ; \gamma_2 \mid \gamma_1 \cup \gamma_2 \mid \gamma^* \mid \varphi?$
- Formulae $\varphi :=$ false $\mid p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \langle\gamma\rangle\varphi$

Composite games – from games α and β

- Role change α^d
- Choice $\alpha \cup \beta$ (and $\alpha \cap \beta$)
- Composition $\alpha ; \beta$
- Iteration α^* (and α°)
- Test games φ? (φ any GL formula)

Normal form

NNF, dual only on atomic games, and only atomic test games.

Game operations – iteration

Repeated games

- From game γ define the iterated game γ^*, where player 1 can choose to repeat γ after each play, but only finitely often.
- Defined in terms of **least fixed points**.
- The dual operation $\gamma^\circ := \gamma^{d^{*^d}}$ is a **greatest fixed point** – player 2 decides whether to continue, possibly forever.

Fixed-point unwindings

- Eventualities $\langle g^* \rangle p \leftrightarrow (p \lor \langle g \rangle \langle g^* \rangle p)$.
- Invariants $\langle g^\circ \rangle p \leftrightarrow (p \land \langle g \rangle \langle g^\circ \rangle p)$.

Game operations – iteration

The combination of eventualities and invariants allows reasoning about reactive systems that do not terminate.

There is an infinite g-path: $\langle g^\circ \rangle$true.

There is no infinite g-path: $\neg\langle g^\circ \rangle \text{true} \equiv \langle g^{d^*} \rangle$false.

The CTL formula $\forall p \mathcal{U} q$ can be written $\varphi := \langle (p?; g^d)^* \rangle q$. Along all g-paths q is eventually true, and p is true meanwhile.
One step unwinding: $q \vee (p \wedge \langle g^d \rangle \varphi)$.

Fixed point constructions can be nested, e.g. $\langle (g^*; h \cup h^\circ)^* \rangle p$.

Game Logic and μ-calculus

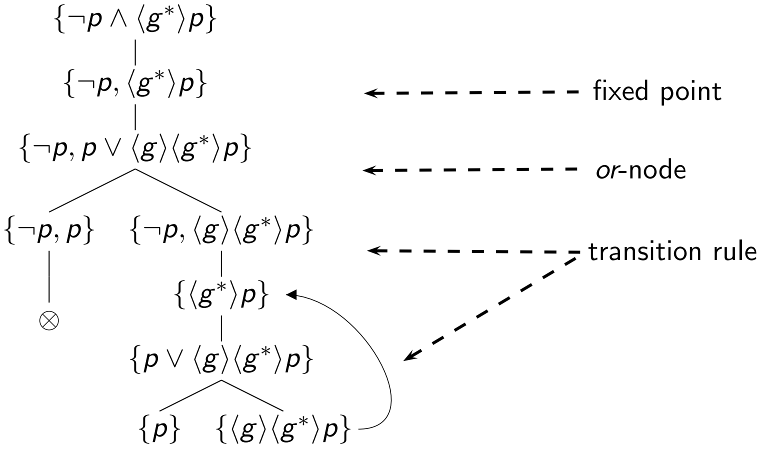
- GL embeds naturally into μ-calculus, needing only two fixed point variables
  $\langle (p?; g^d)^* \rangle q$ becomes $\mu X.q \vee (p \wedge [g]\, X)$

- Two independent hierarchies identified in μ-calculus:
  - Fixed point alternation hierarchy (nesting)
  - Variable hierarchy
- GL intersect with all levels of the alternation hierarchy, but fits (somewhere?) in the two-variable fragment.

Understanding μ-calculus

- Tableaux, Automata and Games.

Tableau example: $\neg p \wedge \langle g^* \rangle p$



Tableaux rules – transitional rule

The transitional rule determines branching for the atomic games:

$$\{\langle g\rangle p, \langle g\rangle q, \langle g^d\rangle r, \}$$

$$\{p, r\} \qquad \{q, r\}$$

- The transitional rule may only be applied when no other rules are applicable.
- Each atomic game is considered with its dual. For g and g^d we have

$$\frac{\langle g\rangle\varphi_1, \langle g\rangle\varphi_2, \ldots, \langle g^d\rangle\psi_1, \langle g^d\rangle\psi_2, \ldots}{\{\varphi_1, \psi_1, \psi_2, \ldots\} \mid \{\varphi_2, \psi_1, \psi_2, \ldots\} \mid \ldots}$$

Tableaux rules – fixpoint regeneration

Iterated games are just unwound ...

$$\frac{\langle\gamma^*\rangle\varphi}{\varphi \vee \langle\gamma\rangle\langle\gamma^*\rangle\varphi} \qquad \frac{\langle\gamma^\circ\rangle\varphi}{\varphi \wedge \langle\gamma\rangle\langle\gamma^\circ\rangle\varphi}$$

... but lead to infinite branches.
On nested invariants and eventualities we define an ordering that respects the syntactic nesting.

For $\langle(g^*; h \cup h^\circ)^*\rangle p$ we could take

$$\langle g^*\rangle\langle h\rangle\varphi < \langle h^\circ\rangle\varphi < \langle(g^*; h \cup h^\circ)^*\rangle p$$

For infinite branches we consider the largest fixed point formula that is regenerated infinitely often – this determines $*$-branch or $\circ$-branch.

Tableaux closure

A tableau for a given GL formula may be seen in two ways:

- To show unsatisfiability of the formula (**validity** of the negation): At universal branching nodes (*or*) select both children, at existential branching (transition) nodes we select one child. Show that leaf nodes are closed (contain contradictions) and that all infinite branches are $*$-branches.
- To show **satisfiability** of the formula, consider the complement: At *or* nodes select one child, at transition nodes select all children. Show that leaf nodes are not closed and that no infinite branch is a $*$-branch.

Exactly one of these cases hold (by determinacy of parity games).

Parikh's Game Logic replayed

GL is an interesting logic

- as a conservative extension to PDL suited to reasoning about ongoing computations of reactive systems,
- as a modal logic combining fixed point extensions monotone neighbourhood semantics, giving an indication of the interaction (or lack thereof) between these extensions, and
- as an expressive fragment of μ-calculus with a variable-free syntax.

Some open problems

- Proving completeness of Parikh's axiomatization
- Positioning GL in the two-variable fragment of μ-calculus

Tools – Tableaux and Automata (and Games) for Game Logic.