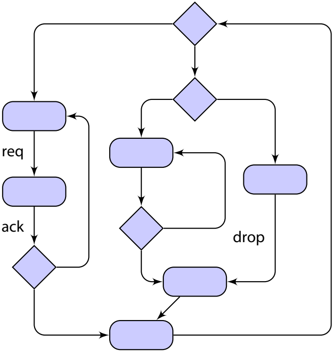
Game Quantification Patterns

Dietmar Berwanger and Sophie Pinchinat
ENS Cachan & CNRS
IRISA Rennes

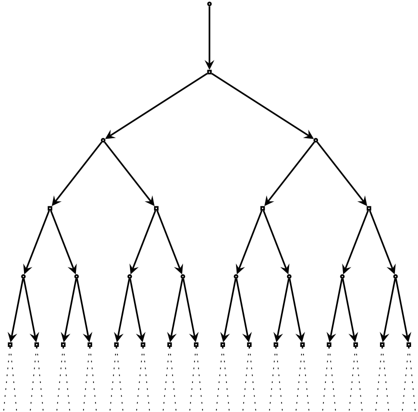
ICLA, Chennai 2009

Logics of Computation

- Model: transition structure



computation tree, path



- Specification dynamic PDL · branching time CTL* · linear time LTL

MSO, μ-calculus, automata

Computation and Interaction

1980: Shift of paradigma ▶ **reactiveness**

- Interactive **control**
  - system vs environment
  - multi-component systems
- Specification as an objective of **conflict**
  - system as a decision-maker
  - model checking games, verification

Game metaphor: interactive transition structure + objective/utility

Players, agents?

Logics for Interaction

Describe how external agents gear into the system:

- atomic transitions
- composition (sequential, iteration)

Game Logic [Parikh 1983]

- ► generalises Program Dynamic Logic PDL - internal view

  programs $\rightsquigarrow$ protocols between two agents

Alternating Time Logic [Alur, Henzinger, Kupferman 1998]

- ► generalises Computation Tree Logic CTL* - external view

  1-agent $\rightsquigarrow$ n-agent systems

Local interaction, global utility.

Game Logic

- Story: Angel and Demon
- Transition structures: neighbourhood models

  - effectivity functions - enforcible outcomes in atomic transitions
- Syntax

  - regular expression γ: rules of a game between Angel and Demon

$$\gamma := a \mid \varphi? \mid \gamma;\gamma \mid \gamma \cup \gamma \mid \gamma^* \mid \gamma^{\mathrm{d}}$$

  - formula φ: a property of states

  $\rightsquigarrow$ modal operator $\langle\gamma\rangle\varphi$
- Semantics

  Angel has a strategy to play γ such that
  φ holds in the state at which the game ends.

Alternating-Time Logics

- Story: system with n-agents
- Transition structures: concurrent game structures
  - game matrix describes outcomes of simultaneous atomic moves
- Syntax
  - formula φ, a linear-time property of paths

$$\varphi := p \mid \varphi \vee \varphi \mid \neg \eta \mid \textbf{next}\ \varphi \mid \varphi\ \textbf{until}\ \varphi$$

  - strategy quantifier with a coalition $C \subseteq \{1, \ldots, n\}$

  $\rightsquigarrow$ relativisation construction: $\langle C \rangle \varphi$
- Semantics

  Coalition C of agents has a strategy such that
  φ holds on any path following the strategy.

Comparing Game Logic with ATL

- Models, interpretation of atoms: Embedding

  Neighbourhood models vs Concurrent game structures

  Extensive game structures

- Automata to capture effects of composition:
  - Game Logic: complex procedural rule, simple winning condition
    - iterated alternation $(g^*)^d$ -- highly expressive
    - game modaltity relates sets of states
  - ATL: simple procedural rule, complex winning condition

In vivo *vs* ***post factum*** *interpretation.*

(1) Disenchanting the meta-language

ATL actually speaks about two-player, sequential zero-sum games -- just like Game Logic.

- atomic games $\rightsquigarrow$ **game forms** - just outcomes, no preferences
  - untyped forms - actions partitioned but not attributed
  - types: attribute strategy sets to players
- **non-intentional agents**, act on behalf of a player
  - multi-agent scenarios (matrices) induce untyped game forms
  - meaning of swapping players
- **sequentialisations** of a concurrent game are particular types

(2) Ensure model compatibility

Extensive game structures

(Q, Prop, Γ: $Q \longrightarrow$ untyped games)

- extend both concurrent game structures and neighbourhood models.

Effectivity functions and agent forms are untyped game forms.

Theorem. Strategic equivalence under sequential play:

> If two untyped game forms have the same effectivity,
> their sequentialisations are 1-step equivalent:
> -- undistinguishable by atomary transitions of Game Logic or ATL.

(3) Compare recursion patterns

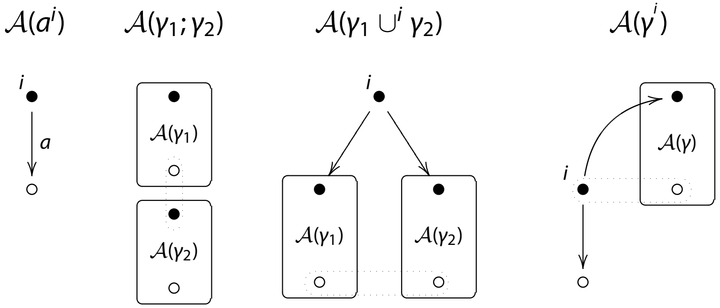
Game automaton:

- state set Q partitioned into existential and universal
- alphabet: atomic propositions p
- transition function $\delta(Q, p) \longrightarrow (Q, Q) \cup (a, Q)$:
  - update internal state or execute a transition of type a
  - sequentalisation order explicit in type
- acceptance condition: parity $\Omega : Q \longrightarrow \mathbb{N}$

Theorem.

Every formula of Alternating Temporal Logic or Game Logic can be translated effectively into a game automaton.

Details: Game Logic to Automata



Theorem

A class of models is definable in Game Logic iff it is recognisable by an automaton with single-entry single-exit transition graph.

ATL to Automata

- bottom-up compostition
- determinisation of counter-free word automata

Remarks:

- connected components in transition graph have all the same type
- translation involves determinisation: exponential blow-up

Conclusions

- At the **atomic** level, Game Logic and ATL do not differ:
  - they distinguish the same models
  - concurrency and multi-agent features in ATL are semantically irrelevant
- The efficient fragment ATL of ATL^* is subsumed by Game Logic
- ATL^* can be exponentially more **succinct** than Game Logic.
- The **recursion** mechanisms are indeed distinct.
  - easy to find properties expressible in Game-Logic but not in ATL^*.
  - converse is hard.