Imperfect-Information Games in Computing

Dietmar Berwanger

RWTH Aachen  ➔  LSV, CNRS & ENS Cachan, Paris

1st LINT Workshop, Amsterdam 2008
Normal vs extensive form

- Normal-form game for \(n\) players:

\[
\Gamma = \left( (S^i)_{i<n}, (u^i)_{i<n} \right)
\]

Each player picks a strategy \(s^i \in S^i\):

\[\text{outcome } s = (s^0, s^1, \ldots, s^{n-1}) \quad \Rightarrow \quad \text{payoff } u^i(s).\]

- Extensive form: \textbf{structure} in strategies.
Perfect information: transition structure
Strategy with perfect information
Pruning
Imperfect information: + information structure
Strategy with imperfect information
Strategy with imperfect information
Strategy with imperfect information
Imperfect-information strategy

Diagram showing decision nodes and outcomes.

- Nodes represent decision points.
- Arrows indicate possible moves.
- Terminal nodes represent outcomes.

Note: The diagram illustrates the structure of decision-making processes in games with imperfect information, where players do not have complete knowledge of the state of the game.
Strategy with imperfect information
Characterisation

- Imperfect information:

  strategy: \{\text{information set}\} \rightarrow \text{action}

- Perfect information:

  all information sets are singletons
Another characterisation

The perfect-information property of a strategy set:

$\in S \land \in S \Rightarrow \in S$

Imperfect information is about available sets of strategies
Games in Computation

Wishlist:

- model plays of infinite duration
- capture uncertainty about
  - order of moves
  - initial state
  - implementation details
- compositionality and tractable complexity
Why and how

- Automata theory
  - nondeterminism
  - synchronisation/homing sequences

- Controller synthesis
  - Plant, supervisor, observable/controllable event
  - Safety conditions

- Distributed computation
  - private/public variables, alternation, scheduling
  - Acceptance: reachability conditions
Verification: reactive dynamics

System: actions \(\uparrow, \downarrow\)  
Environment: observations \(\bullet, \bullet, \circ\)
Winning conditions

- **Reachability**: observe a good event

- **Safety**: never observe a bad one

- **Büchi**: something good over and over again

- **Parity**: observation priorities — least one seen infinitely often is even
  - nested Reachability & Safety

- **ω-regular**: finite-state monitor is satisfied

Transform a reactive model into a game.
Parity games with imperfect information

Parity games are generic for $\omega$-regular specifications.

Strategy: (Observations)$^*$ → Actions.

Questions:

- **decide** whether the system can ensure a win
- **construct** a winning strategy

Assumptions here:

- strictly turn based
- observable winning condition
- sur win
Classical solution [Reif84]

Powerset construction:

- keeps track of what the system can distinguish from memory
- yields game with perfect information (over the powerset)

Winning positions and strategy can be transferred back and forth.

**Corollary:** Memoryless determinacy / perfect-information

- **Finite-memory determinacy** / imperfect information
Complexity

- Problem is \texttt{EXPTIME}-hard
- Exponential \texttt{memory} might be needed.

However, the information-set construction

- uses \texttt{exponential} space
- has \texttt{no on-the-fly} solution
- is independent of \texttt{objective}

Can we do better?
Antichains

Interesting sets have a particular structure
Antichains

- Interesting sets have a particular structure
  - downwards-closed
Antichains

- Interesting sets have a particular structure
  - downwards-closed

- Interesting operations preserve it:
  - $\text{CPre}$, $\cup$, $\cap$, iteration

$\text{CPre}(X) = \{ Y : \text{System can force the play into} \ X \}$. 
Antichains

- Interesting sets have a particular structure
  - downwards-closed
- Interesting operations preserve it:
  - \( \text{CPre, } \cup, \cap, \text{ iteration} \)

\[ \text{CPre}(X) = \{ Y : \text{System can force the play into } X \} . \]
Algorithm with antichains / imperfect information

[CDHR07]

- Evaluates characterisation of winning positions as a $\mu$-calculus formula over the lattice of antichains

- Strategy **construction** more tricky, but works.
Three players = Trouble

Two players, with common winning condition $W \subseteq V^*$ (all finite!)
the third one is indifferent.

Q: Are there strategies $(s^1, s^2)$ such, that for all $s^3$, the outcome is in $W$.

Folk argument. This problem is undecidable.
Reduction from Halting Problem

Imperfect information + the third player can enforce coordination on
the $n$-th configuration of a Turing Machine $M = (Q, \Sigma, q_0, \delta)$
in round $n$.

Ingredients:

- Actions: $\Sigma \cup Q \cup \{\uparrow, \blacksquare\}^2$,
- Observations: $\{\uparrow, \blacksquare\}$
- Mask: Player $i$ sees $\blacksquare$ iff in component $i$; otherwise $\uparrow$;
The actions of the two protagonists can produce descriptions \((x, x')\) of machine configurations.

The transition structure can enforce that either

\[
(G_{\Leftarrow}) \quad x \Leftarrow x',
\]

\[
(G_{\Rightarrow}) \quad x \Rightarrow x', \text{ or}
\]

\[
(G_{=}) \quad x = x'
\]
Construction
Undecidability

- any distributed winning strategy \((s^1, s^2)\) must produce the \(n\)-th configuration of \(M\) after observing \(\triangleright^n\)
  
  – defined only, if machine never halts.

**Conclusion** The distributed control problem is in general undecidable.
Outlook

- Interactive vs Distributed
  - player aggregation
  - inductive solution concepts
- Specific kinds of imperfect information
  - uncertainty about initial state
  - concurrency
- Abstraction, interface theories