Finite Causal Games

Finite Environment
- Game “world” with finitely many states
- One initial state and one or more terminal states
- Fixed finite number of players
- Each with finitely many “percepts” and “actions”
- Each with one or more goal states

Causal Model
- Environment changes only in response to moves
- Synchronous and asynchronous actions
Finite Causal Games

Finite Environment
Game “world” with finitely many states
One initial state and one or more terminal states

Fixed finite number of players
Each with finitely many “percepts” and “actions”
Each with one or more goal states

Causal Model
Environment changes only in response to moves
Synchronous and asynchronous actions
Games as State Machines
Initial State and Terminal States
Goal States
Simultaneous Actions
Game Model

An $n$-player game is a structure with components:

- $S$ - set of states
- $A_1, \ldots, A_n$ - $n$ sets of actions, one for each player
- $l_1, \ldots, l_n$ - where $l_i \subseteq A_i \times S$, the legality relations
- $n: S \times A_1 \times \ldots \times A_n \rightarrow S$ - update function
- $s_1 \in S$ - initial game state
- $t \subseteq S$ - the terminal states
- $g_1, \ldots, g_n$ - where $g_i \subseteq S$, the goal relations
Direct Description

Since all of the games that we are considering are finite, it is possible in principle to communicate game information in the form of lists (of states and actions), and tables (for legality, update, etc.)

Problem: Size of description. Even though everything is finite, the necessary tables can be large.

Solutions:
Reformulate in modular fashion
Use compact encoding
States versus Features

In many cases, worlds are best thought of in terms of features, e.g. red or green, left or right, high or low. Actions often affect subset of features.

States represent all possible ways the world can be. As such, the number of states is exponential in the number of “features” of the world, and the action tables are correspondingly large.

Idea - represent features directly and describe how actions change individual features rather than entire states. (Reference: STRIPS.)
States versus Features

In many cases, worlds are best thought of in terms of features, e.g. red or green, left or right, high or low. Actions often affect a subset of features. States represent all possible ways the world can be. As such, the number of states is exponential in the number of “features” of the world, and the action tables are correspondingly large.

Idea - represent features directly and describe how actions change individual features rather than entire states. (Reference: STRIPS.)

Propositional Net Components

Propositions

Connectives

Transitions
Propositional Net

A marking for a propositional net is a function from the propositions $P$ to boolean values.

$$m : P \rightarrow \{true, false\}$$

A marking $m$ is partial if and only if $m$ is a partial function. Otherwise, it is total.
Markings

A marking for a propositional net is a function from the propositions $P$ to boolean values.

$$m: P \rightarrow \{\text{true}, \text{false}\}$$

A marking $m$ is partial if and only if $m$ is a partial function. Otherwise, it is total.
Acceptability

A marking is *acceptable* if and only if it obeys the logical properties of all connectives.

Negation with input $x$ and output $y$:

\[ m(x) = \text{false} \iff m(y) = \text{true} \]

Conjunction with inputs $x$ and $y$ and output $z$:

\[ m(x) = \text{true} \land m(y) = \text{true} \iff m(z) = \text{true} \]

Disjunction with inputs $x$ and $y$ and output $z$:

\[ m(x) = \text{true} \lor m(y) = \text{true} \iff m(z) = \text{true} \]
Update

A transition is *enabled* by a marking $m$ if and only if all of its inputs are marked *true*.

The *transitional marking* for $m$ is the partial marking that assigns *true* to the outputs of all transitions enabled by $m$ and *false* to the outputs of all others.

An *input marking* for a propositional net is a marking for the extrinsic propositions.

The *update* $m'$ of a marking $m$ (with transitional marking $t$) and input marking $e$ is any acceptable marking consistent with $t$ and $e$. 
Example

Buttons and Lights

Pressing button a toggles p.

Pressing button b interchanges p and q.
Buttons and Lights

Pressing button \(a\) toggles \(p\).
Pressing button \(b\) interchanges \(p\) and \(q\).
Propositional Net for Buttons and Lights
Tic-Tac-Toe

X

O

X
Partial Propositional Net for Tic-Tac-Toe

- \textit{black}(1,1)
- \textit{cell}(1,1,b)
- \textit{cell}(1,1,o)
- \textit{black}(1,2)
- \textit{cell}(1,2,b)
- \textit{cell}(1,2,o)
- \textit{black}(1,3)
- \textit{cell}(1,3,b)
- \textit{cell}(1,3,o)

Logical Description

Direct encoding in relational logic:
\[\text{next(\textit{cell}(1,1,o))} \leq \text{next(\textit{cell}(3,3,o))} \leq \text{true(\textit{black}(1,1))} \land \ldots \land \text{true(\textit{black}(3,3))} \land \text{true(\textit{cell}(1,1,b))} \land \text{true(\textit{cell}(3,3,b))} \]

Use of variables to compact description:
\[\text{next(\textit{cell}(M,N,o))} \leq \text{true(\textit{black}(M,N))} \land \text{true(\textit{cell}(M,N,b))} \]

Other Compression Techniques:
- Logical nesting
- Views (subroutines)
Logical Description

Direct encoding in relational logic:
\[
\text{next(cell}(1,1,o)) \leq \text{true(black}(1,1)) \& \ldots \text{true(black}(3,3)) \& \text{true(cell}(1,1,b)) \leq \text{true(cell}(3,3,b))
\]

Use of variables to compact description:
\[
\text{next(cell}(M,N,o)) \leq \text{true(black}(M,N)) \& \text{true(cell}(M,N,b))
\]

Other Compression Techniques:
Logical nesting
Views (subroutines)
Relational Logic

Object Variables: $x, y, z$
Object Constants: $a, b, c$
Function Constants: $f, g, h$
Relation Constants: $p, q, r$
Logical Operators: $\neg, \&$, $|$, $\implies$, $\leq$, $\leq$, $\text{distinct}$

Terms: $x, y, z, a, b, c, f(a), g(a, b), h(a, b, c)$
Relational Sentences: $p(a, b)$
Logical Sentences: $r(x, y) \leq p(x, y) \& \neg q(y)$

Standard first-order semantics.
Vocabulary

Game-specific names for:
players, e.g. robot, white, black
actions, e.g. a, b, mark(1,2)
propositions, e.g. p, q, cell(1,1,b)
views, e.g. row(2,x)

Game-independent relation constants:
init(proposition)
true(proposition)
does(player,action)
next(proposition)
legal(player,action)
goal(proposition)
terminal
Initial State

\[
\begin{align*}
\text{init}(\text{cell}(1,1,b)) \\
\text{init}(\text{cell}(1,2,b)) \\
\text{init}(\text{cell}(1,3,b)) \\
\text{init}(\text{cell}(2,1,b)) \\
\text{init}(\text{cell}(2,2,b)) \\
\text{init}(\text{cell}(2,3,b)) \\
\text{init}(\text{cell}(3,1,b)) \\
\text{init}(\text{cell}(3,2,b)) \\
\text{init}(\text{cell}(3,3,b)) \\
\text{init}(\text{control}(x))
\end{align*}
\]
Legality

\[
\text{legal}(W, \text{mark}(X,Y)) \iff \\
\quad \text{true}(\text{cell}(X,Y,b)) \land \\
\quad \text{true}(\text{control}(W))
\]

\[
\text{legal}(\text{white}, \text{noop}) \iff \\
\quad \text{true}(\text{cell}(X,Y,b)) \land \\
\quad \text{true}(\text{control}(o))
\]

\[
\text{legal}(\text{black}, \text{noop}) \iff \\
\quad \text{true}(\text{cell}(X,Y,b)) \land \\
\quad \text{true}(\text{control}(x))
\]
Update

\[ \text{next}(\text{cell}(M,N,x)) \iff \]
\[ \text{does(white,mark}(M,N)) \land \]
\[ \text{true(cell}(M,N,b)) \]

\[ \text{next}(\text{cell}(M,N,o)) \iff \]
\[ \text{does(black,mark}(M,N)) \land \]
\[ \text{true(cell}(M,N,b)) \]

\[ \text{next}(\text{cell}(M,N,W)) \iff \]
\[ \text{true(cell}(M,N,W)) \land \]
\[ \text{distinct}(W,b) \]

\[ \text{next}(\text{cell}(M,N,b)) \iff \]
\[ \text{does}(W,\text{mark}(J,K)) \land \]
\[ \text{true(cell}(M,N,b)) \land \]
\[ (\text{distinct}(M,J) \mid \text{distinct}(N,K)) \]
Update (continued)

\[
\text{next(control(x))} \iff \\
\text{true(control(o))}
\]

\[
\text{next(control(o))} \iff \\
\text{true(control(x))}
\]
Termination

\[
\begin{align*}
\text{terminal} & \iff \text{line}(W) \\
\text{terminal} & \iff \neg \text{open}
\end{align*}
\]

\[
\begin{align*}
\text{line}(W) & \iff \text{row}(M, W) \\
\text{line}(W) & \iff \text{column}(N, W) \\
\text{line}(W) & \iff \text{diagonal}(W)
\end{align*}
\]

\[
\begin{align*}
\text{open} & \iff \text{true}(\text{cell}(M, N, b))
\end{align*}
\]
Supporting Concepts

row(M,W) <=
  true(cell(M,1,W)) ∧
  true(cell(M,2,W)) ∧
  true(cell(M,3,W))

column(N,W) <=
  true(cell(1,N,W)) ∧
  true(cell(2,N,W)) ∧
  true(cell(3,N,W))

diagonal(W) <=
  true(cell(1,1,W)) ∧
  true(cell(2,2,W)) ∧
  true(cell(3,3,W))

diagonal(W) <=
  true(cell(1,3,W)) ∧
  true(cell(2,2,W)) ∧
  true(cell(3,1,W))
Goals

\[
goal(white, 100) \iff \text{line(x)}
\]
\[
goal(white, 50) \iff \neg\text{line(x)} \& \neg\text{line(o)} \& \neg\text{open}
\]
\[
goal(white, 0) \iff \text{line(o)}
\]

\[
goal(black, 100) \iff \text{line(o)}
\]
\[
goal(white, 50) \iff \neg\text{line(x)} \& \neg\text{line(o)} \& \neg\text{open}
\]
\[
goal(white, 0) \iff \text{line(x)}
\]
Completeness

Of necessity, game descriptions are logically incomplete in that they do not uniquely specify the moves of the players.

Every game description contains complete definitions for legality, termination, goalhood, and update in terms of the primitive actions and the does relation.

The upshot is that in every state every player can determine legality, termination, goalhood, and, given a joint move, can update the state.
Playability

A game is *playable* if and only if every player has at least one legal move in every non-terminal state.

Note that in chess, if a player cannot move, it is a stalemate. Fortunately, this is a terminal state.

In GGP, we guarantee that every game is playable.
Winnability

A game is strongly winnable if and only if, for some player, there is a sequence of individual moves of that player that leads to a terminating goal state for that player.

A game is weakly winnable if and only if, for every player, there is a sequence of joint moves of the players that leads to a terminating goal state for that player.

In GGP, every game is weakly winnable, and all single player games are strongly winnable.
Knowledge Interchange Format

Knowledge Interchange Format is a standard for programmatic exchange of knowledge represented in relational logic.

Syntax is prefix version of standard syntax. Some operators are renamed: not, and, or. Case-independent. Variables are prefixed with ?.

\[
\begin{align*}
    r(X,Y) & \leq p(X,Y) \ & \lnot q(Y) \\
    (\leq (r ?x ?y) (\text{and} (p ?x ?y) (\lnot (q ?y)))) \\
    (\leq (r ?x ?y) (p ?x ?y) (\lnot (q ?y)))
\end{align*}
\]

Semantics is the same.