Introduction

Michael Genesereth/Nat Love
Spring 2006

Game Playing

Human Game Playing
• Intellectual Activity
• Skill Comparison

Computer Game Playing
• Testbed for AI
• Limitations
General Game Playing

General Game Players are systems

able to accept descriptions of games at *runtime*

able to use such descriptions to play the games

Technology

Unlike specialized game players (e.g. Deep Blue), they do not use algorithms designed in advance for specific games.

Artificial Intelligence Technologies

knowledge representation
reasoning and learning
rational behavior
Tic-Tac-Toe

X

O

X

Chess
Knight-Zone Chess

Bughouse Chess
Other Games

Single Player “Games”
Activities

Testbed for General Game Playing
Games
Game Manager
Reference Players

Annual Competition
AAAI 2005 - Winner: Jim Clune, UCLA
AAAI 2006 - $10,000 prize to the winner

Website - http://games.stanford.edu
Research Papers
Educational Materials
Discussion groups
Research Profiles

AAA-05 Qualifying Teams

Jim Clune - UCLA and Parasoft
Stephan Schiffel, Michael Thielscher - Dresden University
David Kaiser - Florida International University
Xinxin Sheng - North Carolina State
Peter Stone, Greg Kuhlmann, Kurt Dresner - UT Austin
Matt Grounds - University of York
AAAI-05 Qualifying Match

AAAI-05 Semifinal Match
AAAII-05 Final Match

<table>
<thead>
<tr>
<th>White</th>
<th>Black</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>moveWhite(2,2,22) moveRock(2,2,22)</td>
</tr>
<tr>
<td>2</td>
<td>moveWhite(2,2,22) moveRock(2,2,22)</td>
</tr>
<tr>
<td>3</td>
<td>moveWhite(2,2,22) moveRock(2,2,22)</td>
</tr>
<tr>
<td>4</td>
<td>moveWhite(2,2,22) moveRock(2,2,22)</td>
</tr>
<tr>
<td>5</td>
<td>moveWhite(2,2,22) moveRock(2,2,22)</td>
</tr>
<tr>
<td>6</td>
<td>moveWhite(2,2,22) place(north)</td>
</tr>
<tr>
<td>7</td>
<td>moveWhite(2,2,22) moveRock(2,2,22)</td>
</tr>
<tr>
<td>Score</td>
<td>100</td>
</tr>
</tbody>
</table>

AAAII-05 Runner-up David Kaiser
AAAI-05 Winner Jim Clune

AAAI-05 Spectators
Programme for Today

General Game Playing

Game Description Language

General Game Players

General Game Playing Logistics
Finite Deterministic Games

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

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

Deterministic Operation
Environment changes only in response to moves
Synchronous and asynchronous actions
Single Player Game as a State Machine

Composite Actions for Multiple Players
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.
Game Description Language

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 transition graphs.

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

Ideas:
1. Different Conceptualization
2. Compact Encoding
Relational Logic

Object Variables: x, y, z, …
Object Constants: a, b, c, …
Function Constants: f, g, h, …
Relation Constants: p, q, r, …
Logical Operators: =, #, ~, |, :-

Terms: x, y, z, a, b, c, f(a), g(a, b), h(a, b, c)
Relational Sentences: p(a, b)
Rules: r(X,Y) :- p(X,Y) & ~q(Y) & X#b

Safe rules with stratified negation.
Complete definitions; so NAF is okay.

Vocabulary

Game-specific names for:
players, e.g. white, black
actions, e.g. mark(row, col)
fluen ts, e.g. cell(row, col, mark)
views, e.g. row(row, mark)

Game-independent relation constants:
init(fluent) - fluent is true in the initial state
t rue(fluent) - fluent is true in the current state
next(fluent) - fluent will be true in the next state
legal(player, action) - player may perform action in the current state
does(player, action) - player performs action in the current state
goal(player, score) - current state is worth score to player
ter minal - current state is 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

\[
\begin{align*}
\text{legal} & (W, \text{mark}(X,Y)) :- \\
& \text{true}(\text{cell}(X,Y,b)) \& \\
& \text{true}(\text{control}(W)) \\
\text{legal} & (\text{white}, \text{noop}) :- \\
& \text{true}(\text{cell}(X,Y,b)) \& \\
& \text{true}(\text{control}(\text{black})) \\
\text{legal} & (\text{black}, \text{noop}) :- \\
& \text{true}(\text{cell}(X,Y,b)) \& \\
& \text{true}(\text{control}(\text{white}))
\end{align*}
\]
Physics

```prolog
next(cell(M,N,x)) :-
does(white,mark(M,N))
next(cell(M,N,o)) :-
does(black,mark(M,N))
next(cell(M,N,Z)) :-
does(W,mark(M,N)) & true(cell(M,N,Z)) & Z#b
next(cell(M,N,b)) :-
does(W,mark(J,K)) & true(cell(M,N,b)) & (M#J | N#K)
next(control(white)) :-
true(control(black))
next(control(black)) :-
true(control(white))
```

Supporting Concepts

```prolog
row(M,W) :-
true(cell(M,1,W)) & true(cell(1,1,W)) & true(cell(M,2,W)) & true(cell(2,2,W)) & true(cell(M,3,W))
column(N,W) :-
true(cell(1,N,W)) & true(cell(1,3,W)) & true(cell(2,N,W)) & true(cell(2,2,W)) & true(cell(3,N,W))
line(W) :- row(M,W)
line(W) :- column(N,W)
line(W) :- diagonal(W)
open :- true(cell(M,N,b))
```
Goals and Termination

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

\[
\text{terminal} :\text{ line}(W) \\
\text{terminal} :\text{ ~open} \\
\]

No Built-in Assumptions

What we see:

\[
\text{next(cell}(M,N,x)) :- \\
\text{does(white,mark}(M,N)) \& \\
\text{true(cell}(M,N,b))
\]

What the player sees:

\[
\text{next(welcoul}(M,N,himenoing)) :- \\
\text{does(poontron,dukepse}(M,N)) \& \\
\text{true(welcoul}(M,N,lorenchise))
\]
General Game Players

Logic

Possible to use logical reasoning for everything
  Computing legality of moves
  Computing consequences of actions
  Computing goal achievement
  Computing termination

Easy to convert to other representations
  Simplicity of logical formulation
  Well-known, efficient algorithms
  See, for example, Ullman: Database Systems
State Representation

\begin{center}
\begin{tabular}{ccc}
\hline
& X & \\
\hline
O & & X \\
\hline
\end{tabular}
\end{center}

\begin{align*}
\text{cell}(1,1,x) \\
\text{cell}(1,2,b) \\
\text{cell}(1,3,b) \\
\text{cell}(2,1,b) \\
\text{cell}(2,2,o) \\
\text{cell}(2,3,b) \\
\text{cell}(3,1,b) \\
\text{cell}(3,2,b) \\
\text{cell}(3,3,x) \\
\text{control(black)}
\end{align*}

Legality Computation

\begin{align*}
\text{legal}(W, \text{mark}(X,Y)) & : - \\
& \text{true(cell}(X,Y,b) \land \text{true(control(W)))}
\end{align*}

\begin{align*}
\text{legal}(\text{white}, \text{noop}) & : - \\
& \text{true(cell}(X,Y,b) \land \text{true(control(black)))}
\end{align*}

\begin{align*}
\text{legal}(\text{black}, \text{noop}) & : - \\
& \text{true(cell}(X,Y,b) \land \text{true(control(white)))}
\end{align*}

Conclusions:

\begin{align*}
\text{legal(white,noop)} & \quad \text{legal(black,mark(1,2))} \\
\text{...} & \quad \text{legal(black,mark(3,2))}
\end{align*}
**Update Computation**

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

\[
\begin{align*}
&\text{black} \\
&\text{mark}(1,3) \\
\end{align*}
\]

\[
\text{next}\left(\text{cell}(M,N,o)\right) : - \\
\text{does}\left(\text{black}, \text{mark}(M,N)\right)
\]

**Game Tree Expansion**

\[
\begin{array}{c|c|c}
X & O & X \\
\hline
O & & X \\
\hline
O & X & \\
\end{array}
\]

\[
\begin{array}{c|c|c}
X & O & X \\
\hline
O & O & X \\
\hline
O & X & X \\
\end{array}
\]

\[
\begin{array}{c|c|c}
X & O & X \\
\hline
O & O & X \\
\hline
O & X & \\
\end{array}
\]

\[
\begin{array}{c|c|c}
X & O & X \\
\hline
O & O & X \\
\hline
O & X & X \\
\end{array}
\]
Game Tree Search

Resource Limitations

Large state spaces
~5000 states in Tic-Tac-Toe
~$10^{30}$ states in Chess

Limited Resources
Memory
Time (start clock, move clock)
Partial Search

Incremental Search
 Expand Game Graph as much as time allows
 Use Minimax evaluation on non-terminal states

Evaluation function for non-terminal states?
 Proximity to goal
 Mobility for self [Clune]
 Lack of mobility for opponents

Reverse book - enlarge space of goal states
Avoid non-winnnability (assuming joint actions)
Novelty and reversibility

Reformulation

Reformulation
 Abstract problem
 Solve abstract problem
 Refine abstract solution to complete solution

Graph Analysis
 Bottlenecks
 Symmetry, e.g. *Tic-Tac-Toe*
 Independence, e.g. *Hodgepodge*
Modeling Other Players

Modeling other players
Intra-game modeling useful only for long games
Inter-game modeling possible but
identity information not always available
Must recognize style of play

Assumptions
Rationality?

Dominance Examples

```
  a  b
a  4  3
b  2  1
```

```
  a  b
a  4  1
b  3  2
```
Compilation

Partial Evaluation
  Fold game description into logical reasoning
  Recompile

Specialized Data Structures
  Make propositional net explicit
  1 bit per proposition
  Convert rules to bit operations

Performance Improvement

Performance Improvement over multiple matches
  Unfortunately, game identity not supplied
  Must recognize that it is the same game
  May be able to play matches during start clock

Performance improvement over multiple games
  Algorithms (alpha-beta and so forth)
  Useful Concepts (e.g. “pinned pieces”, “forks”)
General Game Playing Logistics

Gamemaster

Parameterized Collection of Games
  Complete or incomplete information
  One player or n players
  Competition and Cooperation

Services
  Sample Players (including human interface)
  Game Definition Support
  Game Manager

Records
  Players
  Performance Data
Game Manager

Game Descriptions
Match Records

Temporary State Data

Graphics for Spectators

Tcp/ip

Player

Game Playing Protocol

Start
Manager sends Start message to players
Start(id,role,description,startclock,playclock)

Play
Manager sends Play messages to players
Play(id, actions)
Receives plays in response

Stop
Manager sends Stop message to players
Stop(id, actions)
Misplay

Player Errors
  Illegal move
  No move before clock runs out

Manager Action
  Selects a random legal move

Rationale
  Other players must work to win
  Other players do not lose because one player errs

Variety of Games

Number of players
  single player, e.g. Maze World, Blocks World
  2-player games, e.g. Tic-Tac-Toe, Chess
  $n$ player games with allies and enemies

Nature
  simultaneous move, alternating move, other
  with and without communication

Scale
  exhaustively searchable, e.g. Tic-Tac-Toe
  not exhaustively searchable, e.g. Chess
Game Checking

Consistency
  no logical inconsistencies

Completeness
  legal moves, goals, termination, next

Playability
  at least one legal move in every state

Winnability
  there is a way for each player to win

Game Generation

Benefits
  Large source of games for general game players
  Emphasizes general game playing

Problem
  Coming up with interesting variants

Relationships
  Similar to problem of software/hardware testing
  Similar to problem of question generation in CAE
Conclusion

General Game Playing is not a game

Critique: Game playing is *frivolous*.

Reply: The techniques to solve this problem are broadly useful; GGP is effectively general problem solving in the context of other agents.

Sample Application: Enterprise Management
- Behavior: Laws, Business Rules, Contracts
- Planning under such Constraints
- Monitoring and Conformance Testing
- Business Process Management
- Automated Workflow Generation / Legislation
Critique: Human Intelligence is arguably the product of eons of evolution. We are, to some extent at least, “wired” to function well in this world. General Game Players have nothing at their disposal but mathematics.

Reply: A key indicator of intelligence is the ability to function in radically new environments.

Using What We Learn

Critique: Real intelligence requires the ability to figure out new environments.

Reply: Well, there is no question that that ability is essential. However, real intelligence also requires the ability to use theories once they are formed.

This is the domain of specification-based systems / declarative systems / and so forth; and interest in this problem dates to the beginning of the field.
John McCarthy

The main advantage we expect the advice taker to have is that its behavior will be improvable merely by making statements to it, telling it about its … environment and what is wanted from it. To make these statements will require little, if any, knowledge of the program or the previous knowledge of the advice taker.

Ed Feigenbaum

The potential use of computers by people to accomplish tasks can be “one-dimensionalized” into a spectrum representing the nature of the instruction that must be given the computer to do its job. Call it the what-to-how spectrum. At one extreme of the spectrum, the user supplies his intelligence to instruct the machine with precision exactly how to do his job step-by-step. … At the other end of the spectrum is the user with his real problem. … He aspires to communicate what he wants done … without having to lay out in detail all necessary subgoals for adequate performance.
Robert Heinlein

computer/robot

A human being should be able to change a diaper, plan an invasion, butcher a hog, conn a ship, design a building, write a sonnet, balance accounts, build a wall, set a bone, comfort the dying, take orders, give orders, cooperate, act alone, solve equations, analyze a new problem, pitch manure, program a computer, cook a tasty meal, fight efficiently, die gallantly. Specialization is for insects.
http://logic.stanford.edu/classes/games

Schedule

April 5  Introduction
12  Gamemaster Logistics
19  Tabular Game Descriptions
26  Logical Game Descriptions

May 3  Incomplete Search
10  Analysis and Reformulation
17  Opponent Modelling
24  Communication and Cooperation
31  Competition Logistics and Strategy

June 7  Final Competition