General Game Playing in Common Lisp
steve@stevelosh.com
bitbucket.org
github.com
irc.freenode.net

sjl
Goal
Game AI
General$^1$ Game Playing

[1] for some value of "General"
Discrete
Finite
Playable
Winnable
Simultaneous Moves
Perfect Information
Deterministic
Reasoning
Playing
Intelligence
Game
Description
Language
;;; Initial State & Roles

(\text{role x})
(\text{role o})

(\text{init (control x)})
(\text{init (cell 1 1 blank)})
(\text{init (cell 1 2 blank)})
(\text{init (cell 1 3 blank)})
(\text{init (cell 2 1 blank)})
(\text{init (cell 2 2 blank)})
(\text{init (cell 2 3 blank)})
(\text{init (cell 3 1 blank)})
(\text{init (cell 3 2 blank)})
(\text{init (cell 3 3 blank)})

;;; Useful Rules

(<= (row ?n ?mark)
     (true (cell ?n 1 ?mark))
     (true (cell ?n 2 ?mark))
     (true (cell ?n 3 ?mark)))

(<= (column ?n ?mark)
     (true (cell 1 ?n ?mark))
     (true (cell 2 ?n ?mark))
     (true (cell 3 ?n ?mark)))

(<= (diagonal 1 ?mark)
     (true (cell 1 1 ?mark))
     (true (cell 2 2 ?mark))
     (true (cell 3 3 ?mark)))

(<= (diagonal 2 ?mark)
     (true (cell 1 3 ?mark))
     (true (cell 2 2 ?mark))
     (true (cell 3 1 ?mark)))

(<= (line ?mark) (row ?n ?mark))
(<= (line ?mark) (column ?n ?mark))
(<= (line ?mark) (diagonal ?n ?mark))

(<= open
     (true (cell ?row ?col blank)))

;;; Terminal

(<= terminal (line x))
(<= terminal (line o))
(<= terminal (not open))

;;; Goal Values

(<= (goal ?player 100)
     (line ?player))

(<= (goal ?player 0)
     (line ?other)
     (distinct ?player ?other))

(<= (goal ?player 50)
     (not (line x))
     (not (line o))
     (not open))

;;; Legal Moves

(<= (legal ?player (mark ?row ?col))
     (true (cell ?row ?col blank))
     (true (control ?player)))

(<= (legal ?player noop)
     (true (control ?other))
     (distinct ?player ?other))

;;; State Transitions

(<= (next (control x)) (true (control o)))
(<= (next (control o)) (true (control x)))

(<= (next (cell ?row ?col ?player))
     (true (cell ?row ?col blank))
     (distinct ?player blank))

(<= (next (cell ?row ?col ?player))
     (true (cell ?row ?col blank))
     (does ?player (mark ?row ?col)))

(<= (next (cell ?row ?col blank))
     (true (cell ?row ?col blank))
     (does ?player (mark ?x ?y))
     (or (distinct ?row ?x)
         (distinct ?col ?y)))
(true ... )
(true (control x))
(true (cell 1 1 x))
(true (cell 1 2 o))
(true (cell 1 3 blank))
(true (cell 2 1 blank))
(true (cell 2 2 blank))
(true (cell 2 3 blank))
(true (cell 3 1 blank))
(true (cell 3 2 blank))
(true (cell 3 3 blank))
(role ...)
(init ...)}
(role x)
(role o)

(init (control x))
(init (cell 1 1 blank))
(init (cell 1 2 blank))
(init (cell 1 3 blank))
(init (cell 2 1 blank))
(init (cell 2 2 blank))
(init (cell 2 3 blank))
(init (cell 3 1 blank))
(init (cell 3 2 blank))
(init (cell 3 3 blank))
(<= head body ... )
(<= (row ?n ?mark)
  (true (cell ?n 1 ?mark))
  (true (cell ?n 2 ?mark))
  (true (cell ?n 3 ?mark)))

(<= (line ?mark) (row ?n ?mark))
(<= (line ?mark) (column ?n ?mark))
(<= (line ?mark) (diagonal ?n ?mark))

(<= open
  (true (cell ?row ?col blank)))
(terminal)
(goal player value)
(legal player move)
(next ...)
(<= terminal (line x))
(<= terminal (line o))
(<= terminal (not open))
(terminal)
(goal player value)
(legal player move)
(next ...)
(\leq (\text{goal} \ ?\text{player} \ 100)
 (\text{line} \ ?\text{player}))

(\leq (\text{goal} \ ?\text{player} \ 0)
 (\text{line} \ ?\text{other})
 (\text{distinct} \ ?\text{player} \ ?\text{other}))

(\leq (\text{goal} \ ?\text{player} \ 50)
 (\not (\text{line} \ x))
 (\not (\text{line} \ o))
 (\not \text{open}))
(terminal)
(goal player value)
(legal player move)
(next ... )
(<= (legal ?player (mark ?row ?col))
   (true (cell ?row ?col blank)))
   (true (control ?player)))

(<= (legal ?player noop)
   (true (control ?other)))
   (distinct ?player ?other))
(terminal)
(goal player value)
(legal player move)
(next ... )
;; Control flips each turn.

(<= (next (control x))
    (true (control o)))

(<= (next (control o))
    (true (control x)))
Any cell that's already marked stays marked next turn.

(<= (next (cell ?row ?col ?mark))
   (true (cell ?row ?col ?mark))
   (distinct ?mark blank))
;; If a player chooses to mark a blank cell, that cell will have their mark next turn.

(<= (next (cell ?row ?col ?player))
 (true (cell ?row ?col blank))
 (does ?player (mark ?row ?col)))
;; All currently-blank cells that
;; WEREN'T marked stay blank.

(<= (next (cell ?row ?col blank))
 (true (cell ?row ?col blank))
 (does ?player (mark ?x ?y))
 (or (distinct ?row ?x)
     (distinct ?col ?y))))
cl-ggp

https://sjl.bitbucket.io/cl-ggp/
cl-ggp

Works\(^1\) with SBCL, CCL, ABCL, and ECL

[1] for some value of "Works"
cl-ggp

cl-ggp is a tiny framework for writing general game players in Common Lisp.

The cl-ggp system handles the GGP protocol for you and nothing else. If you plan on doing your own GDL reasoning, this is all you need.

The cl-ggp.reasoner system contains a simple Prolog-based reasoner using the Temperance logic programming library. It's useful as a starting point for when writing players.

- License: MIT
- Documentation: https://sjl.bitbucket.io/cl-ggp/
- Mercurial: https://bitbucket.org/sjl/cl-ggp/
- Git: https://github.com/sjl/cl-ggp/

Table of Contents

Installation
Overview
Main API Reference
Reasoner API Reference
Changelog

Made with Lisp and love by Steve Losh in Reykjavík, Iceland.
cd local-projects

git clone https://github.com/sjl/cl-ggp.git

git clone https://github.com/sjl/temperance.git
cl-ggp
cl-ggp.reasoner
(make-reasoner <rules>)

(initial-state <reasoner>)
(legal-moves-for <reasoner> <state> <role>)
(terminalp <reasoner> <state>)
(goal-value-for <reasoner> <state> <role>)
(next-state <reasoner> <state> <moves>)
(defparameter *reasoner* *)

*REASONER*
CL-USER> (ggp.reasoner:initial-state *reasoner*)

((GGP-RULES::CONTROL  GGP-RULES::X)
 (GGP-RULES::CELL 1 1  GGP-RULES::BLANK)
 (GGP-RULES::CELL 1 2  GGP-RULES::BLANK)
 (GGP-RULES::CELL 1 3  GGP-RULES::BLANK)
 (GGP-RULES::CELL 2 1  GGP-RULES::BLANK)
 (GGP-RULES::CELL 2 2  GGP-RULES::BLANK)
 (GGP-RULES::CELL 2 3  GGP-RULES::BLANK)
 (GGP-RULES::CELL 3 1  GGP-RULES::BLANK)
 (GGP-RULES::CELL 3 2  GGP-RULES::BLANK)
 (GGP-RULES::CELL 3 3  GGP-RULES::BLANK))
CL-USER> (ggp.reasoner:legal-moves-for
  *reasoner*
  (ggp.reasoner:initial-state *reasoner*)
  'ggp-rules::x)

((GGP-RULES::MARK 1 1)
 (GGP-RULES::MARK 1 2)
 (GGP-RULES::MARK 1 3)
 (GGP-RULES::MARK 2 1)
 (GGP-RULES::MARK 2 2)
 (GGP-RULES::MARK 2 3)
 (GGP-RULES::MARK 3 1)
 (GGP-RULES::MARK 3 2)
 (GGP-RULES::MARK 3 3))
CL-USER> (ggp.reasoner:legal-moves-for
   *reasoner*
   (ggp.reasoner:initial-state *reasoner*)
   'ggp-rules::o)

(PPG-RULES::NOOP)
Reasoning
Playing
Intelligence
GGP Game/Network Protocol
**Games you've never seen before.** Play over 100 different games against humans or intelligent computers.

Would you like to start a [Two-Player Free-For-All](#) match? **Yes!**

You can include human players, computer players, and random players.

### Listing of the 50 most recently played matches, of which 2 are ongoing:

<table>
<thead>
<tr>
<th>Time Ago</th>
<th>Match 1</th>
<th>Match 2</th>
<th>Game 1</th>
<th>Game 2</th>
<th>View</th>
</tr>
</thead>
<tbody>
<tr>
<td>1m ago</td>
<td>QFWFQ</td>
<td>ggtest1</td>
<td>Gomoku (Swap2 15x15)</td>
<td>View</td>
<td></td>
</tr>
<tr>
<td>7m ago</td>
<td>Alloy_0_10_1</td>
<td>100</td>
<td>Maze</td>
<td>View</td>
<td></td>
</tr>
<tr>
<td>15m ago</td>
<td>QFWFQ</td>
<td>Alloy_0_9</td>
<td>Iterated Ultimatu...</td>
<td>View</td>
<td></td>
</tr>
<tr>
<td>26m ago</td>
<td>Alloy_0_9</td>
<td>QFWFQ</td>
<td>Iterated Tinfoil ...</td>
<td>View</td>
<td></td>
</tr>
<tr>
<td>38m ago</td>
<td>Alloy_0_9</td>
<td>0</td>
<td>Hidato (37 hexes)</td>
<td>View</td>
<td></td>
</tr>
<tr>
<td>40m ago</td>
<td>Alloy_0_10_1</td>
<td>ggtest1</td>
<td>Reversi</td>
<td>View</td>
<td></td>
</tr>
<tr>
<td>52m ago</td>
<td>QFWFQ</td>
<td>Alloy_0_9</td>
<td>Quarto</td>
<td>View</td>
<td></td>
</tr>
<tr>
<td>57m ago</td>
<td>Alloy_0_10_1</td>
<td>ggtest1</td>
<td>Two-Player Free-F...</td>
<td>View</td>
<td></td>
</tr>
<tr>
<td>1h ago</td>
<td>Alloy_0_10_1</td>
<td>ggtest1</td>
<td>9-Board Tic-Tac-T...</td>
<td>View</td>
<td></td>
</tr>
<tr>
<td>1h ago</td>
<td>Alloy_0_10_1</td>
<td>0</td>
<td>Untwisty Complex 2</td>
<td>View</td>
<td></td>
</tr>
<tr>
<td>1h ago</td>
<td>Alloy_0_10_1</td>
<td>32</td>
<td>Iterated Tinfoil ...</td>
<td>View</td>
<td></td>
</tr>
<tr>
<td>2h ago</td>
<td>QFWFQ</td>
<td>Alloy_0_9</td>
<td>Amazons (8x8)</td>
<td>View</td>
<td></td>
</tr>
<tr>
<td>match name</td>
<td>start &amp; play clock</td>
<td>start time</td>
<td>players</td>
<td>goal values</td>
<td>actions</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------</td>
<td>------------------</td>
<td>------------------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>ru_ai17_finals.6x6.24</td>
<td>10, 5</td>
<td>08.03.2017 14:53:21 GMT</td>
<td>ai17 bt_jonse07 ai17 bt_bjornal1</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>ru_ai17_finals.5x5.37</td>
<td>10, 1</td>
<td>08.03.2017 14:55:55 GMT</td>
<td>ai17 bt_simon16 ai17 bt_birgir15</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ru_ai17_finals.6x6.38</td>
<td>10, 5</td>
<td>08.03.2017 14:56:05 GMT</td>
<td>ai17 bt_simon16 ai17 bt_birgir15</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>ru_ai17_finals.6x6.28</td>
<td>10, 5</td>
<td>08.03.2017 14:56:11 GMT</td>
<td>ai17 bt_sveinbjorn14</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>ru_ai17_finals.5x5.25</td>
<td>10, 1</td>
<td>08.03.2017 14:59:12 GMT</td>
<td>ai17 bt_sveinbjorn14 ai17 bt_birgir15</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>ru_ai17_finals.6x6.26</td>
<td>10, 5</td>
<td>08.03.2017 14:59:32 GMT</td>
<td>ai17 bt_sveinbjorn14 ai17 bt_birgir15</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ru_ai17_finals.6x5.42</td>
<td>10, 5</td>
<td>08.03.2017 15:02:21 GMT</td>
<td>ai17 bt_simon16 ai17 bt_sveinbjorn14</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>ru_ai17_finals.8x8.92</td>
<td>10, 10</td>
<td>08.03.2017 15:14:45 GMT</td>
<td>ai17 bt_jonse07 ai17 bt_sveinbjorn14</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ru_ai17_finals.8x8.112</td>
<td>10, 10</td>
<td>08.03.2017 15:14:45 GMT</td>
<td>ai17 bt_james15 ai17 bt_simon16</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>ru_ai17_finals.8x8.126</td>
<td>10, 10</td>
<td>08.03.2017 15:14:45 GMT</td>
<td>ai17 bt_jonse07 ai17 bt_simon16</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ru_ai17_finals.8x8.115</td>
<td>10, 10</td>
<td>08.03.2017 15:21:39 GMT</td>
<td>ai17 bt_andrij15 ai17 bt_birgir15</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ru_ai17_finals.8x8.100</td>
<td>10, 10</td>
<td>08.03.2017 15:25:03 GMT</td>
<td>ai17 bt_james15 ai17 bt_jonse07</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
There are 386 recorded matches. The following table shows matches 1 through 10.

<table>
<thead>
<tr>
<th>Match</th>
<th>Game</th>
<th>Players</th>
<th>Rewards</th>
<th>View Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1343165480614 farmingquadries ary fluxplayer</td>
<td>70 60</td>
<td>View Match</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x1343165516308 farmingquadries theturk gamer</td>
<td>70 100</td>
<td>View Match</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x1343165962187 farmingquadries lejoueur cadiaplayer</td>
<td>60 90</td>
<td>View Match</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x1343236982627 pilgrimage cadiaplayer turboturtle</td>
<td>0 0</td>
<td>View Match</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x1343237123506 pilgrimage gamer ary</td>
<td>0 10</td>
<td>View Match</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x1343237218840 pilgrimage theturk fluxplayer</td>
<td>0 10</td>
<td>View Match</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x1343237382039 pilgrimage lejoueur miniplayer</td>
<td>10 0</td>
<td>View Match</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x1343240270875 firesheep fluxplayer turboturtle</td>
<td>10 20</td>
<td>View Match</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x1343240380112 firesheep lejoueur gamer</td>
<td>30 10</td>
<td>View Match</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x1343240778760 firesheep cadiaplayer ary</td>
<td>40 20</td>
<td>View Match</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Display matches 11 through 20

Comments and complaints to gamereth@stanford.edu.
ggp-base

https://github.com/ggp-org/ggp-base
cl-ggp

https://sjl.bitbucket.io/cl-ggp/
cl-ggp
cl-ggp.reasoner
(defclass my-player (ggp:ggp-player) ...)

(player-start-game <player> <rules> <role> <deadline>)
(player-update-game <player> <moves>)
(player-select-move <player> <deadline>)
(player-stop-game <player>)
Random Player
Reasoning
Playing
Intelligence
Barack Obama

Ask Google who is the [King Of United States] and Google will inform you that it is Barack Obama, the current President of the United States. The Google Answer is pulled from Breitbart, a story they posted five days ago named All Hail King Barack Obama, Emperor Of The United States Of America!

According To Google, Barack Obama Is King Of The United ...
searchengineland.com/according-google-barack-obama-king-united-states...

Prince Henry of Prussia (1726–1802) - Wikipedia, the free ...
For the brother of King Frederick William II of Prussia, see Prince Henry of Prussia ... 1 Biography; 2 Proposal for King of United States; 3 Ancestry; 4 References ...
Biography · Proposal for King of United States · Ancestry · References
Search the Game Tree

[1] for some value of "Tree"
Depth-first search
... with minimax
... and alpha/beta pruning
... and iterative deepening
... and transposition tables
Heuristic

game state $\rightarrow$ estimated value
Monte Carlo Search
Monte Carlo Player
More?
Improved Reasoning

Faster Prolog Implementations

Propositional Networks
Improved Search

Parallelize

Monte Carlo Tree Search

Heuristics in Playouts
Relaxing Restrictions

GDL-II
General Video Game Playing
Thanks!
The GDL & GGP Protocol Spec

cl-ggp
https://sjl.bitbucket.io/cl-ggp/

ggp-base
https://github.com/ggp-org/ggp-base

When Google's Algorithms Fail, Barack Obama Becomes King of USA

Other GGP Resources
http://www.ggp.org/
http://www.general-game-playing.de/