



LUND
UNIVERSITY

General game playing

KRISTINA GAMM & BJARNI BIRGISSON



General game playing

- The goal: Create a generalized game player, which could play any game written in Game Description Language (GDL).
- This presentation: GDL
Working process
Results
Conclusions



General game playing

3-Puzzle

and

Tic Tac Toe

Gameplayer

Rules	State	History
-------	-------	---------

	3
2	1

robot					
right ▾					
right					
First	Back	down	do	Next	Last

Gameplayer

Rules	State	History
-------	-------	---------

		X
O		

white	black				
mark(1,1) ▾	noop ▾				
First	Back	Play	Redo	Next	Last

<http://gamemaster.stanford.edu>



LUND
UNIVERSITY

General game playing



- Game Description Language

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; 3-Puzzle
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; Components
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(role robot)

(<= (base (cell ?m ?n ?t)) (index ?m) (index ?n) (tile ?t))
(base (step 1))
(<= (base (step ?n)) (successor ?m ?n))

(index 1)
(index 2)

(tile 1)
(tile 2)
(tile 3)
(tile b)

(input robot left)
(input robot right)
(input robot up)
(input robot down)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; init
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(init (cell 1 1 b))
(init (cell 1 2 3))
(init (cell 2 1 2))
(init (cell 2 2 1))
(init (step 1))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; legal
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(<= (legal robot left) (true (cell ?m 2 b)))
```



General game playing



- Game Description Language

```
;;;;;;;;;;;;;  
;;; next  
;;;;;;;;;;;;;
```

```
(<= (next (cell 1 ?n b))  
    (does robot up)  
    (true (cell 2 ?n b)))
```

```
(<= (next (cell 2 ?n b))  
    (does robot down)  
    (true (cell 1 ?n b)))
```

```
(<= (next (cell ?m 1 b))  
    (does robot left)  
    (true (cell ?m 2 b)))
```

```
(<= (next (cell ?m 2 b))  
    (does robot right)  
    (true (cell ?m 1 b)))
```

```
(<= (next (cell 2 ?n ?x))  
    (does robot up)  
    (true (cell 2 ?n b))  
    (true (cell 1 ?n ?x)))
```

```
(<= (next (cell 1 ?n ?x))  
    (does robot down)  
    (true (cell 1 ?n b))  
    (true (cell 2 ?n ?x)))
```

```
(<= (next (cell ?m 2 ?x))  
    (does robot left)  
    (true (cell ?m 2 b))  
    (true (cell ?m 1 ?x)))
```

```
(<= (next (cell ?m 1 ?x))  
    (does robot right)  
    (true (cell ?m 1 b))  
    (true (cell ?m 2 ?x)))
```



General game playing

- Game Description Language – What do we know?

Example from 3-puzzle

Role:	<code>(role robot)</code>	Input:	<code>(input robot up)</code>
Init:	<code>(init (cell 1 1 b)) (init (cell 1 2 3)) (init (cell 2 1 2)) (init (cell 2 2 1)) (init (step 1))</code>	Distinct:	<code>(distinct ?x ?m)</code>
True:	<code>true(fact)</code>	Base:	<code>(<= (base (cell ?m ?n ?t)) (index ?m) (index ?n) (tile ?t))</code>
Next:	<code>(<= (next (cell 1 ?n b)) (does robot up) (true (cell 2 ?n b)))</code>	Terminal:	<code>(<= terminal (true (step 7)))</code>
Legal:	<code>(<= (legal robot up) (true (cell 2 ?n b)))</code>		
Does:	<code>(does robot up)</code>		
Goal:	<code>(<= (goal robot 100) (true (cell 1 1 1)) (true (cell 1 2 2)) (true (cell 2 1 3)))</code>		



General game playing

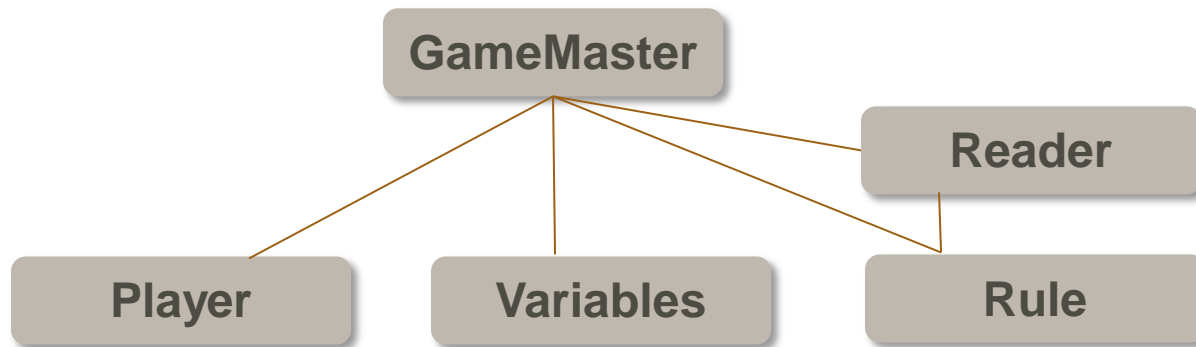
Working process

- Written in java.
- How should we treat unknown facts?
- The limitation of our program.



General game playing

Results and Architecture



General game playing

Results and Architecture

Reader

(<= (base (cell ?m ?n ?t)) (index ?m) (index ?n) (tile ?t))

Rule: head = cell ?m ?n ?t , class = base , subrules = [index ?m][index ?n][tile ?t]

GameMaster – analyse();

(<= (base (cell ?m ?n ?t)) (index ?m) (index ?n) (tile ?t))

(index 1)

(index 2)

GameMaster – escalateNext();

(<= (next (cell 1 ?n b))

(does robot up)

(true (cell 2 ?n b)))

(<= (next (cell ?m ?n ?w))

(does robot up)

(true (cell ?x ?y b))

(true (cell ?m ?n ?w))

(distinct ?y ?n))



General game playing

Results and Architecture

```
You need to be 2 to play this game.
Do you want to play against the computer? Typ y or n.
y
This is your current state in the game

cell 1 1 b
cell 1 2 b
cell 1 3 b
cell 2 1 b
cell 2 2 b
cell 2 3 b
cell 3 1 b
cell 3 2 b
cell 3 3 b
control white

Your possible moves are:
input white mark 1 1
input white mark 1 2
input white mark 1 3
input white mark 2 1
input white mark 2 2
input white mark 2 3
input white mark 3 1
input white mark 3 2
input white mark 3 3

white has to make move!
input white mark 1 1
```

```
true
This is your current state in the game

cell 1 1 x
cell 2 1 b
cell 2 2 b
cell 2 3 b
cell 3 1 b
cell 3 2 b
cell 3 3 b
cell 1 2 b
cell 1 3 b
control black

Your possible moves are:
input white noop

white has to make move!
input white noop
```



General game playing

In Conclusion

- New games provide new pitfalls such as the "or" logic
- Text comparisons may not be the way to solve this
- Extension of GDL brings ALOT more complexity





LUND
UNIVERSITY