Something for everyone:

A.I. lab assignments that span learning styles and aptitudes

Christopher League

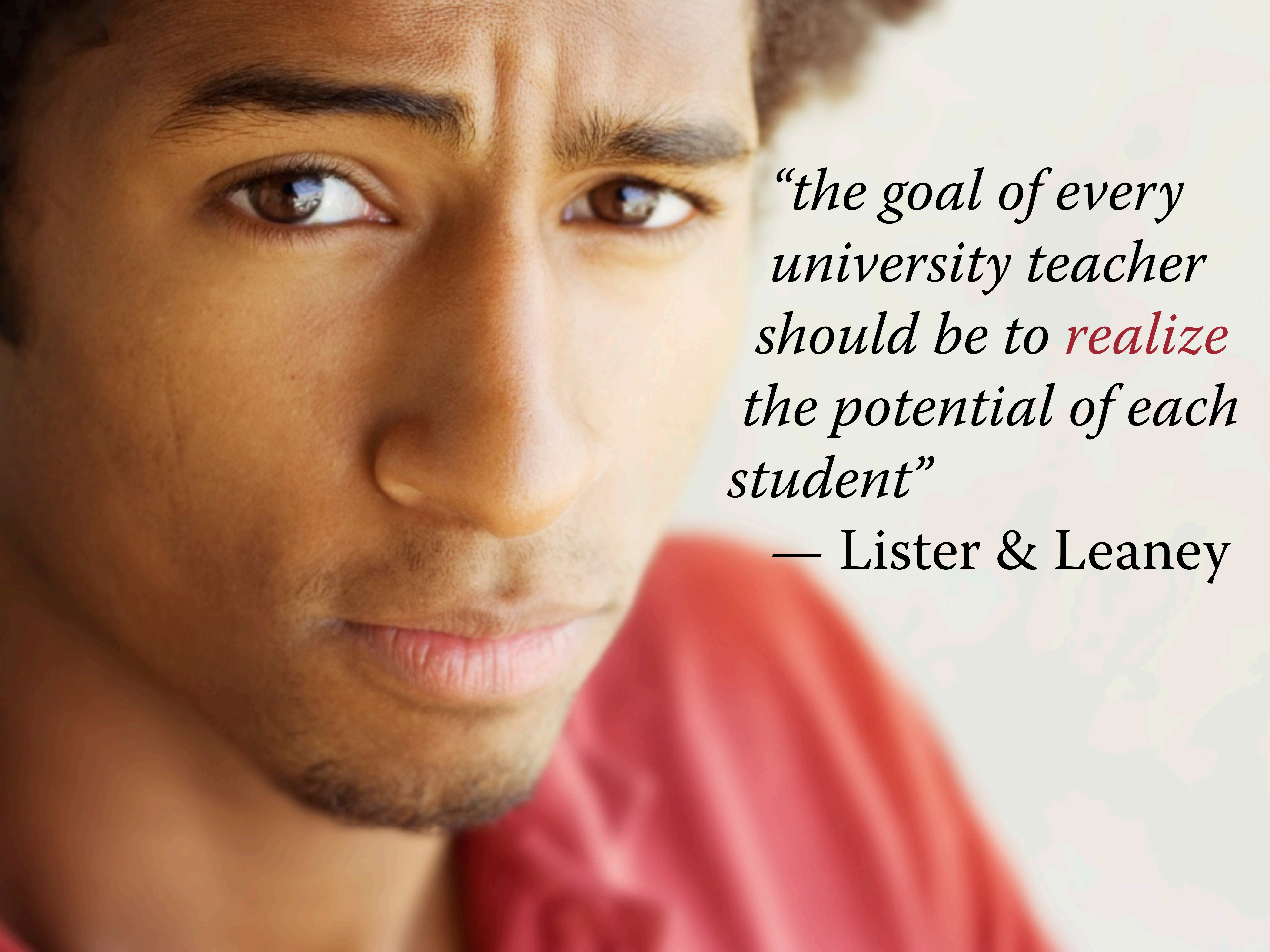
CCSC/NE
12 April 2008

LONG ISLAND UNIVERSITY

Something for everyone:

Cos θ = PF
Av = $\frac{R_1 + R_f}{R_1}$
Vi

wide range of educational backgrounds, learning styles, aptitudes, and time/energy constraints



“the goal of every university teacher should be to realize the potential of each student”

— Lister & Leaney

Bloom, *Taxonomy of educational objectives*, 1956

Bloom, *Taxonomy of educational objectives*, 1956

Evaluation

appraise · conclude · criticize
critique · defend · justify · support

Synthesis

arrange · compile · compose · create · devise
design · extend · generate · modify · plan · write

Analysis

compare · contrast · deconstruct · differentiate
distinguish · illustrate · infer · relate · separate

Application

change · compute · demonstrate · discover
operate · predict · prepare · show · solve · use

Comprehension

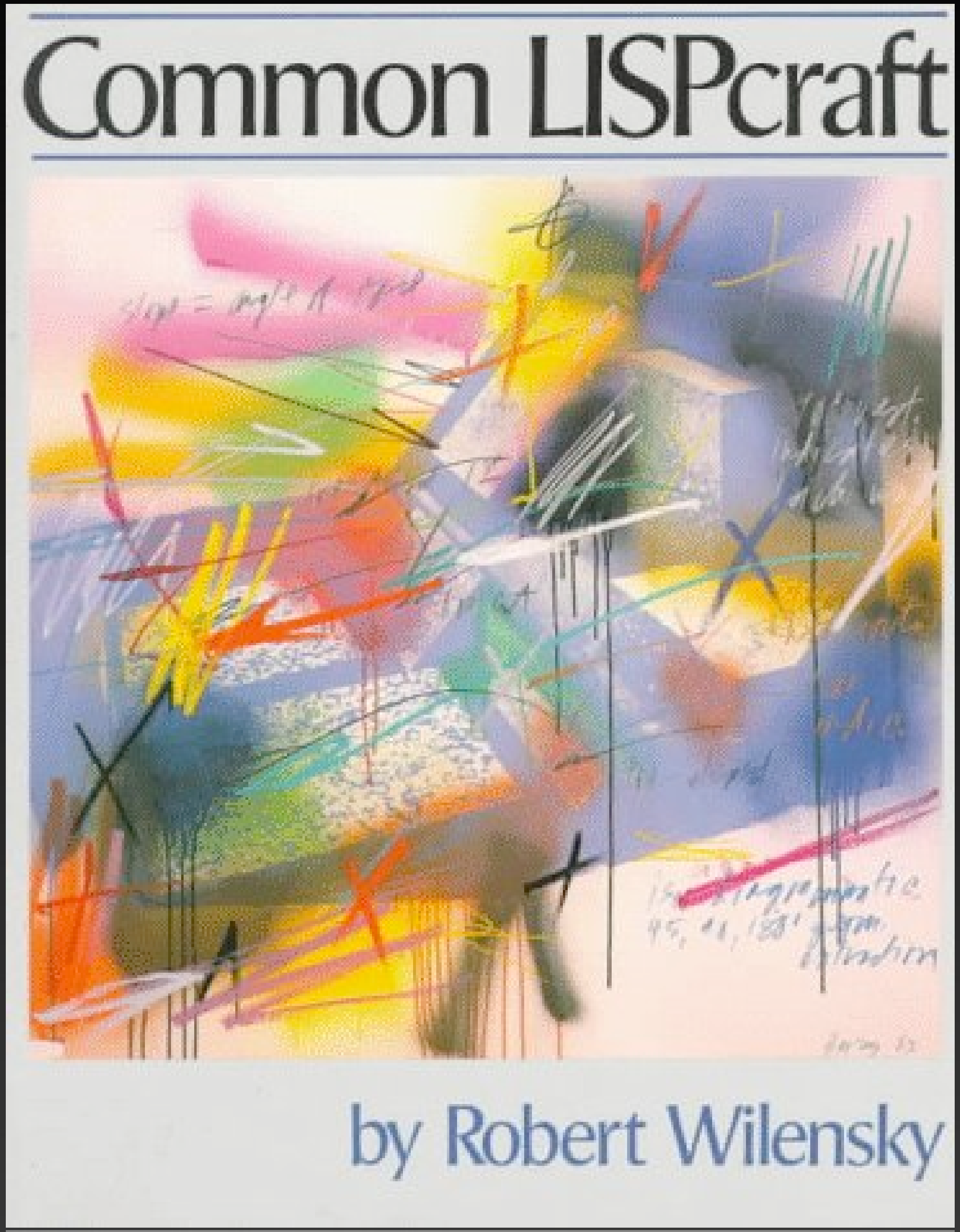
convert · estimate · explain · generalize · exemplify
infer · interpret · paraphrase · summarize · translate

Knowledge

define · describe · identify · label · list · match · name
outline · recall · recognize · reproduce · state

“IT academics place premature emphasis on the higher levels of the taxonomy”

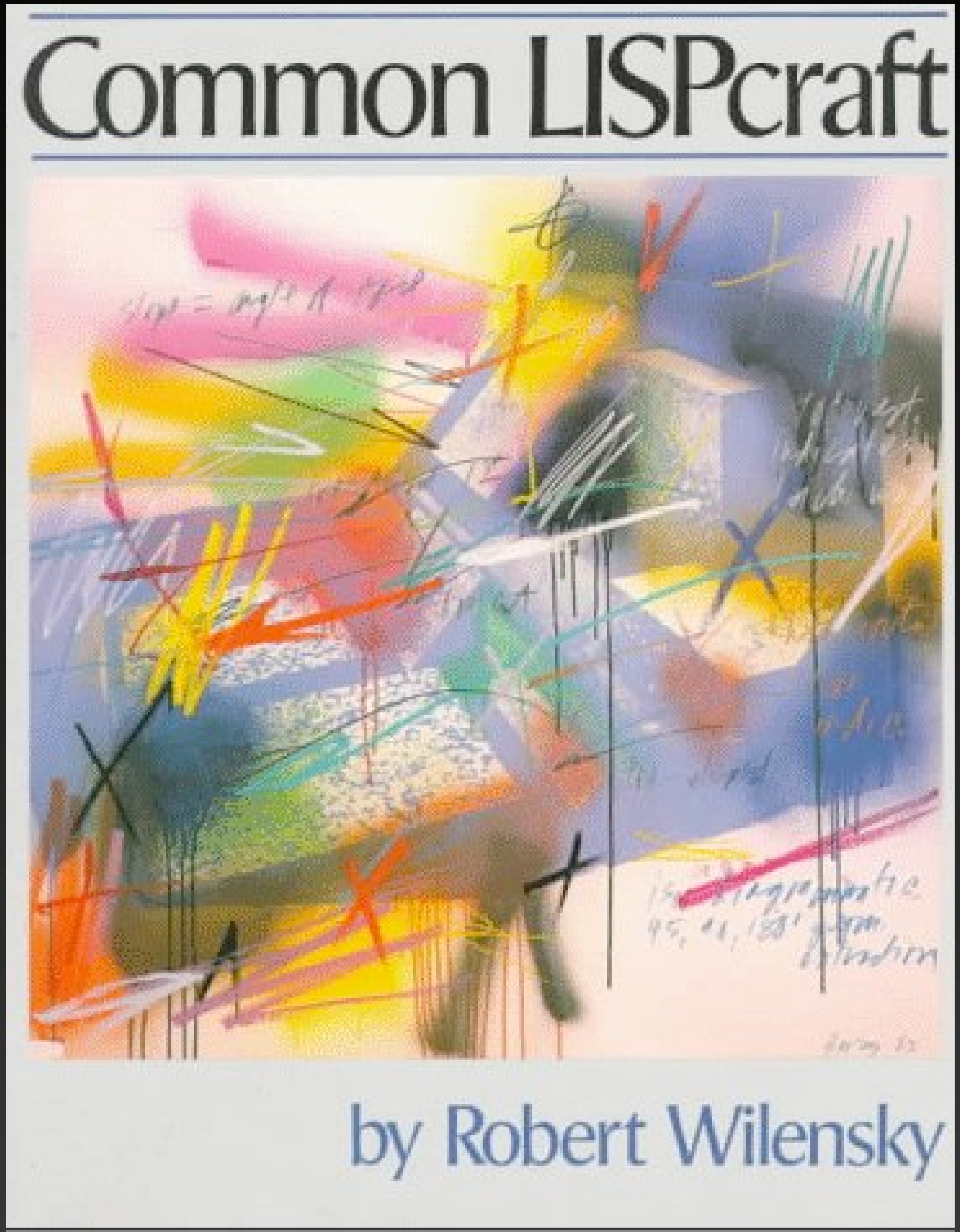
— Lister & Leaney



Common LISPcraft

by Robert Wilensky

“Implement a constraint solver…



due on Tuesday”

Evaluation

Synthesis

Analysis

Application

Comprehension

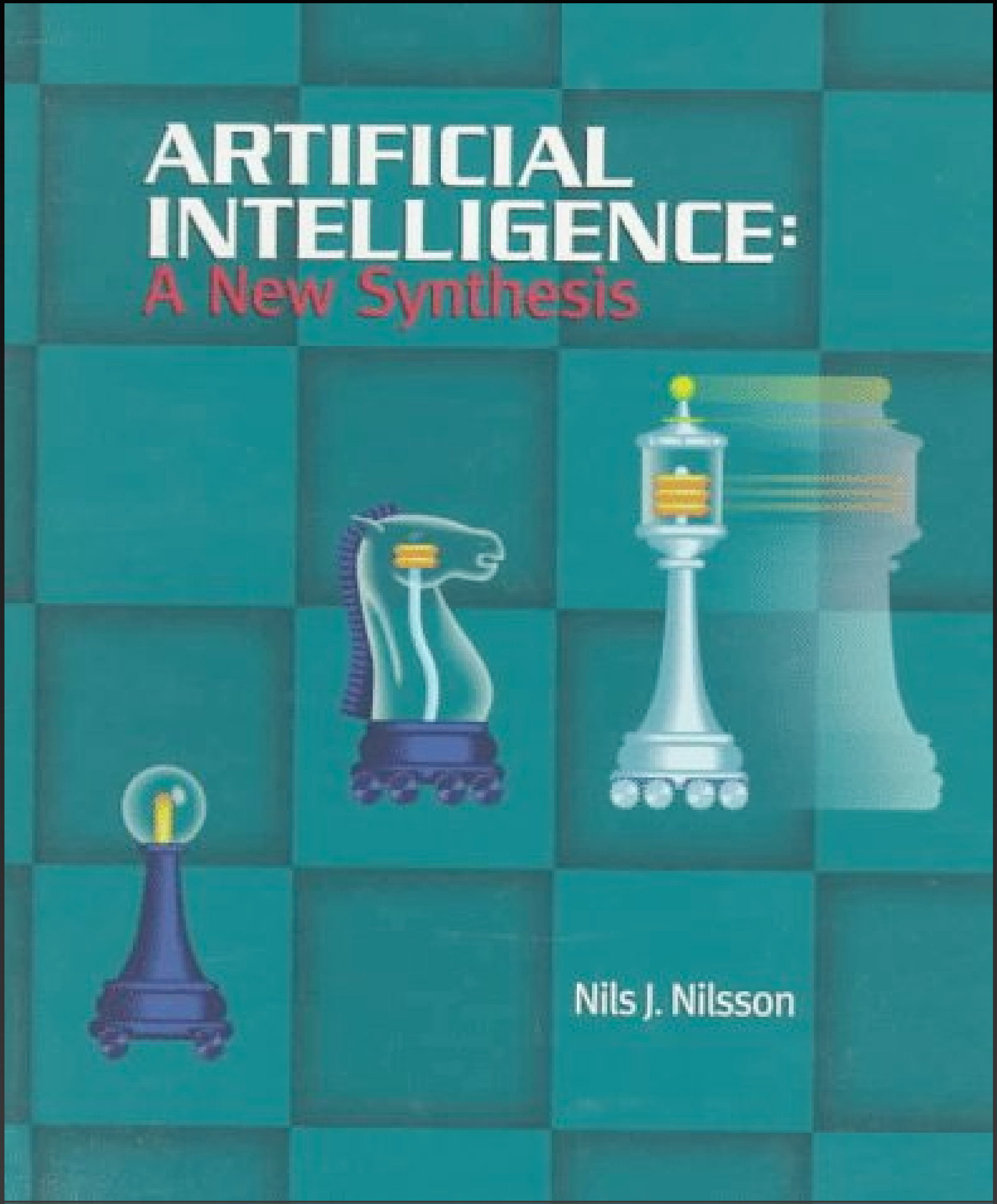
Knowledge

Workbook-style lab assignments that interleave lecture notes and software demos with a series of questions, tasks, and projects at multiple levels of Bloom's taxonomy

Topic Outline

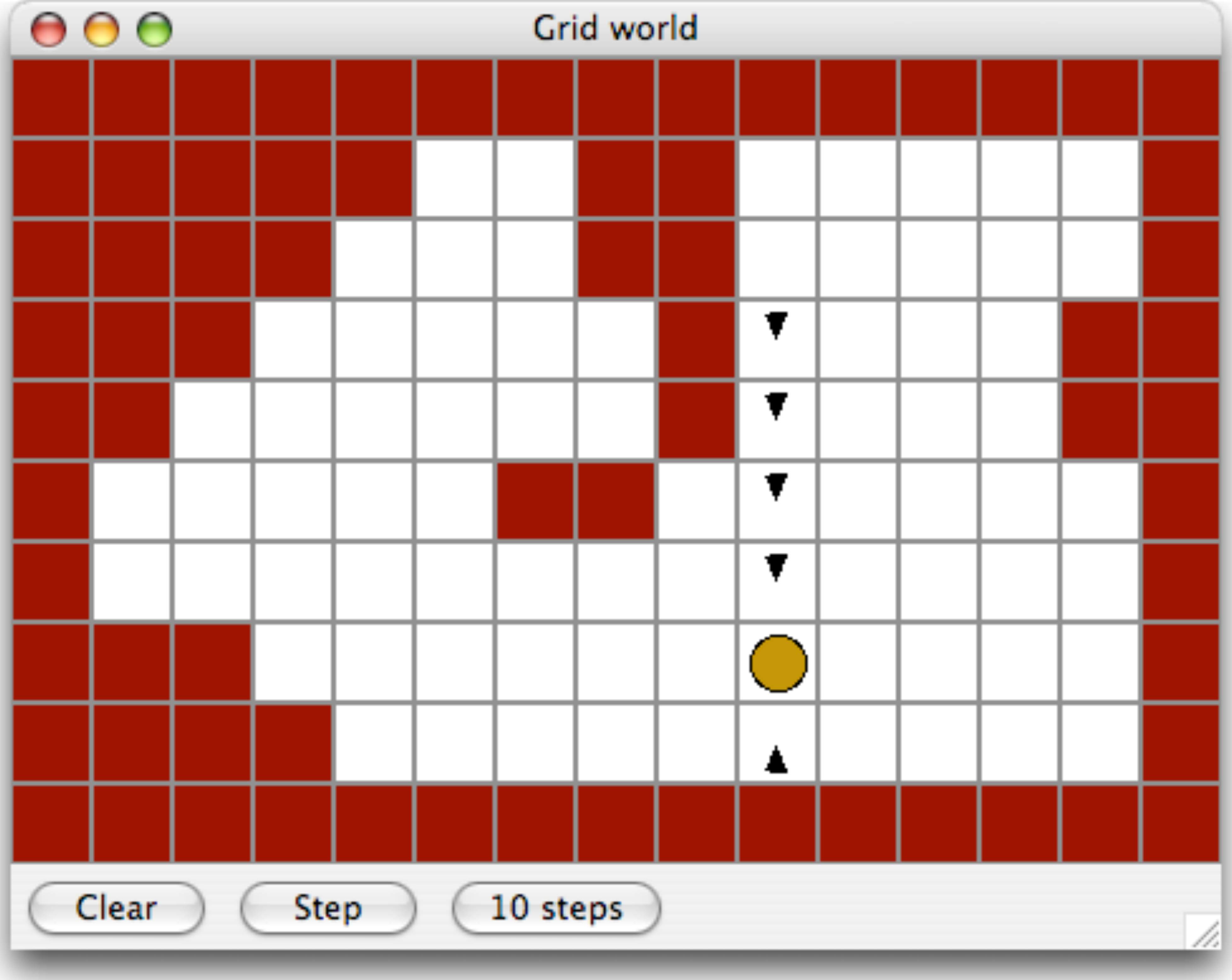
Topic Outline

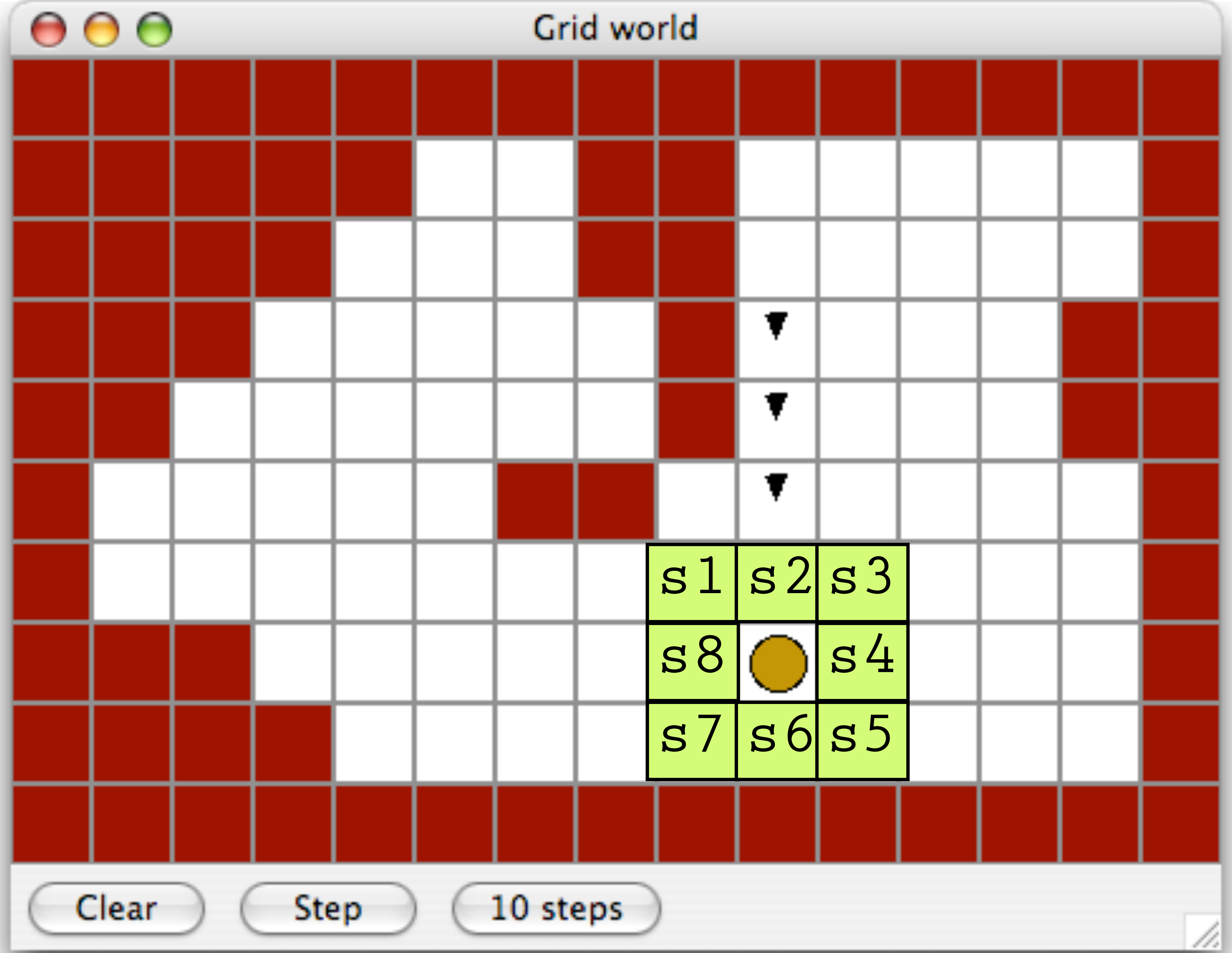
1. **Philosophical background,** strong vs. weak AI, Turing test, chat-bots
2. **Machine learning by example:** classification problems, decision trees, entropy, ID3
3. **Machine learning by evolution:** optimization problems and genetic algorithms
4. Planning using **uninformed and heuristic search:** breadth-first, depth-first, and A* algorithm
5. **Constraint propagation** and satisfaction with AC3
6. **Adversarial search** with minimax & heuristics
7. **Knowledge representation,** logic, expert systems, common sense

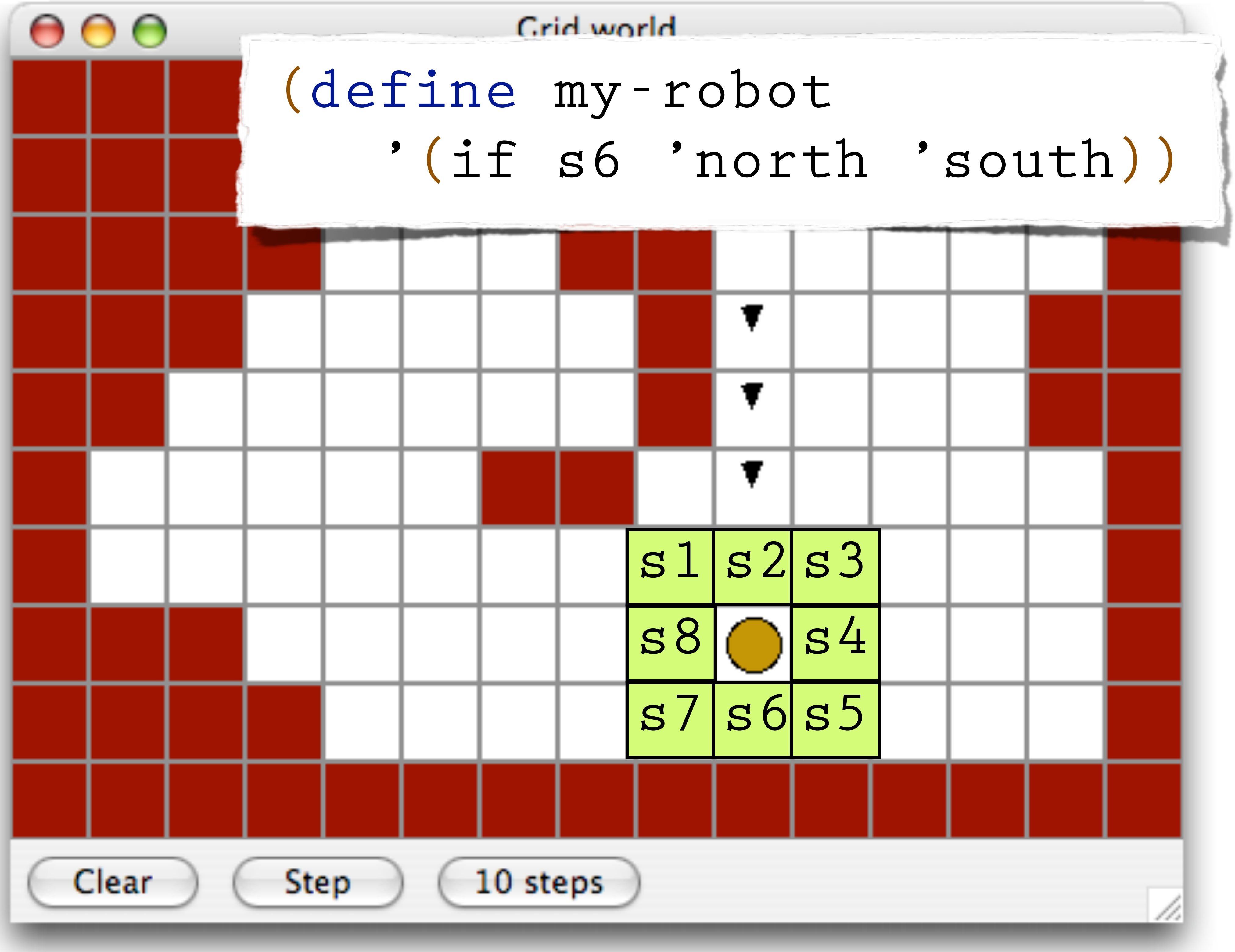


ARTIFICIAL INTELLIGENCE: A New Synthesis

Nils J. Nilsson


Grid world
Clear
Step
10 steps


Grid world
s1
s2
s3
s8
s4
s7
s6
s5
Clear
Step
10 steps

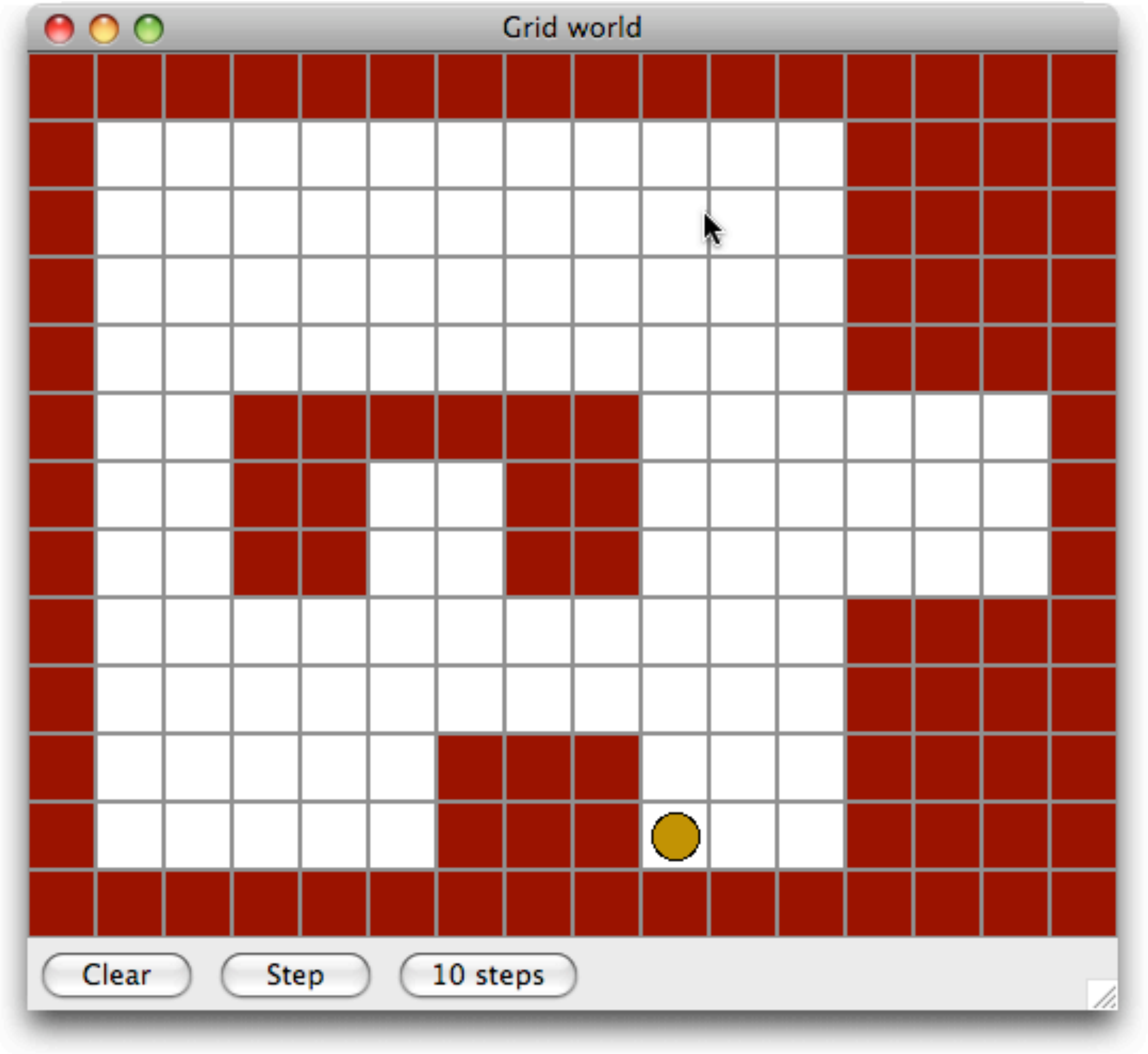


```
(define my-robot
   '(if s6 'north 'south))
```

Dr Scheme

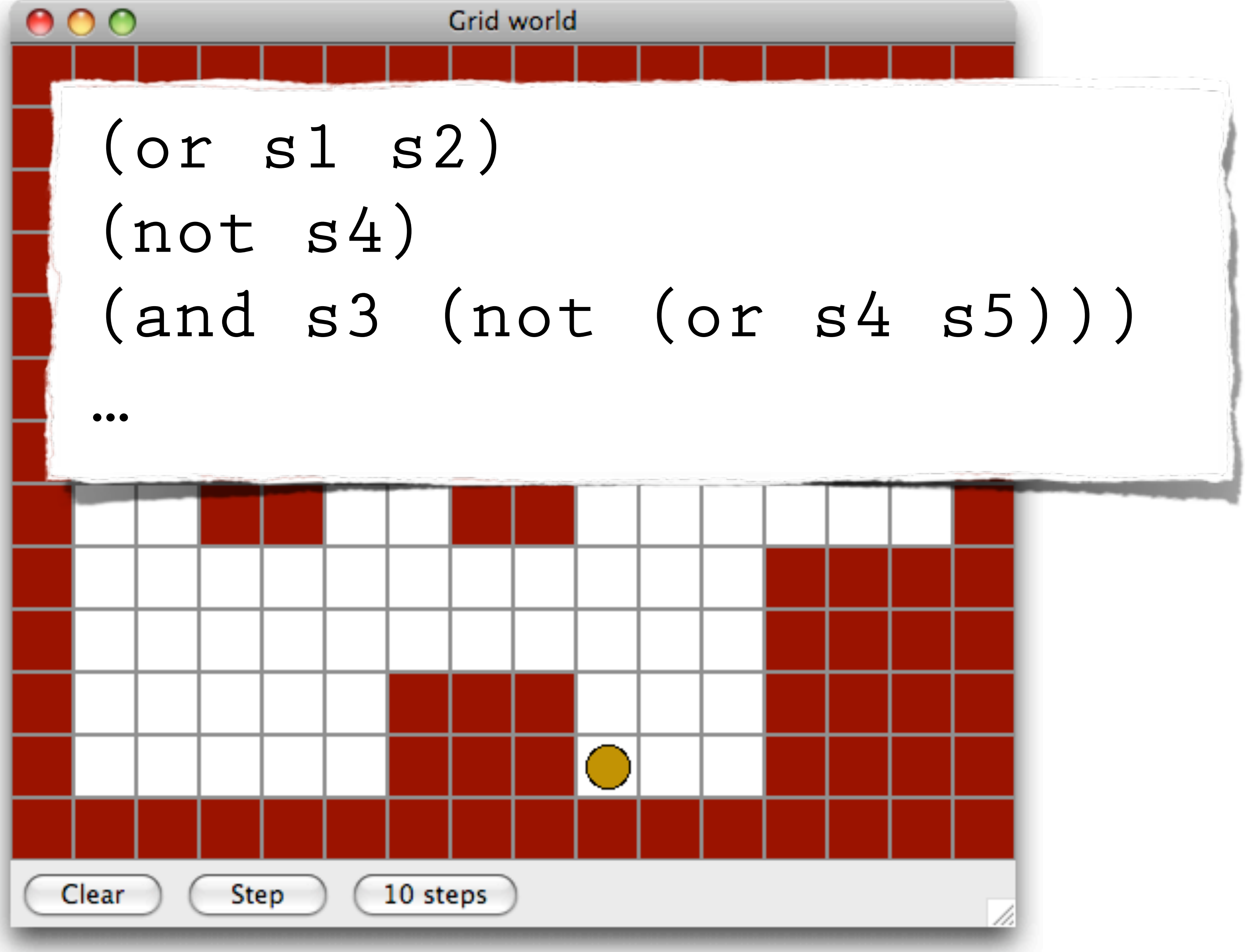
KnowledgeComprehension

- List the values of the robot's sensors at its current location on this map:



Comprehension Application

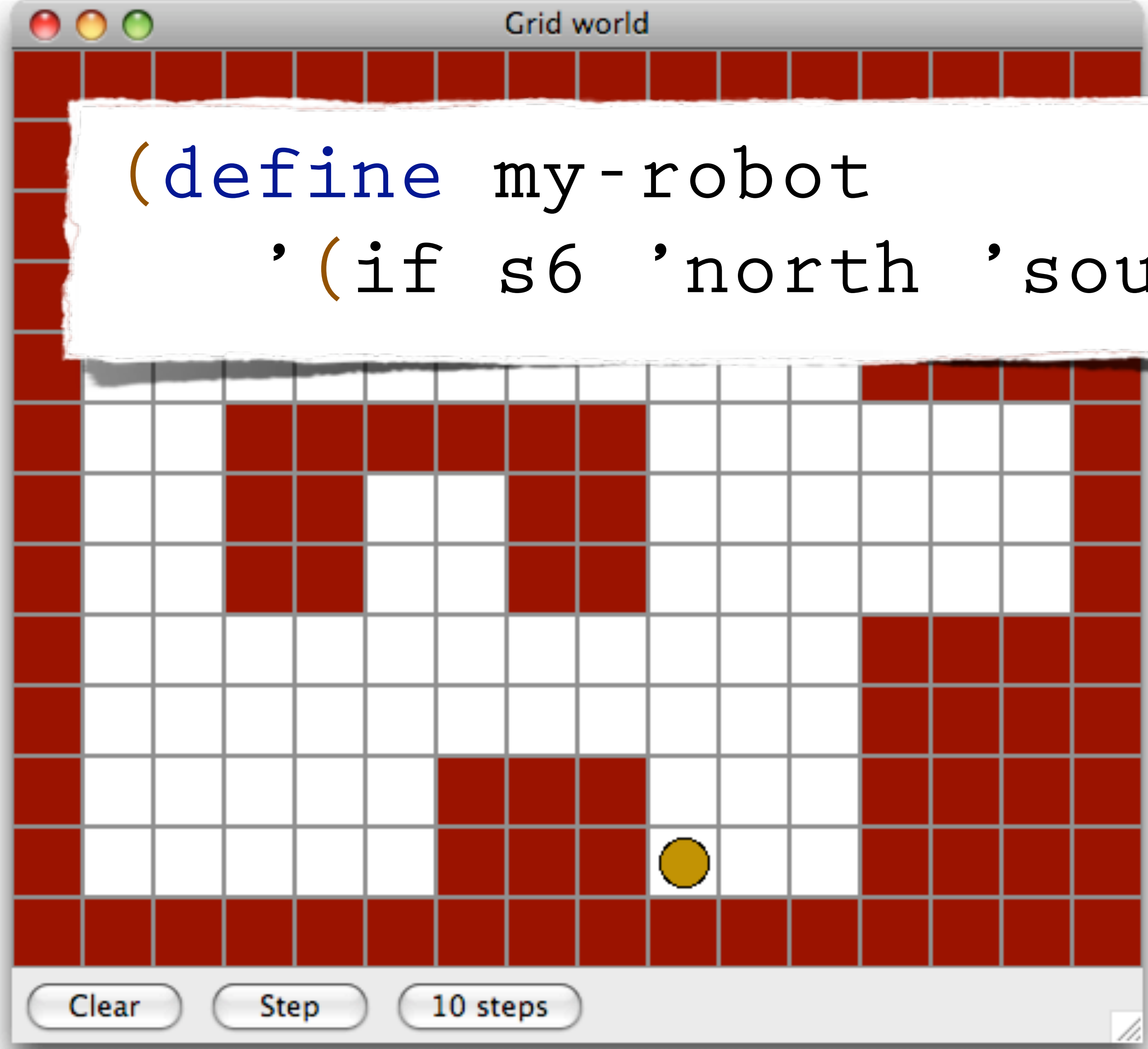
- Carefully compute the truth values of the following conditional expressions:



```
(or s1 s2)
(not s4)
(and s3 (not (or s4 s5)))
...
```

Application

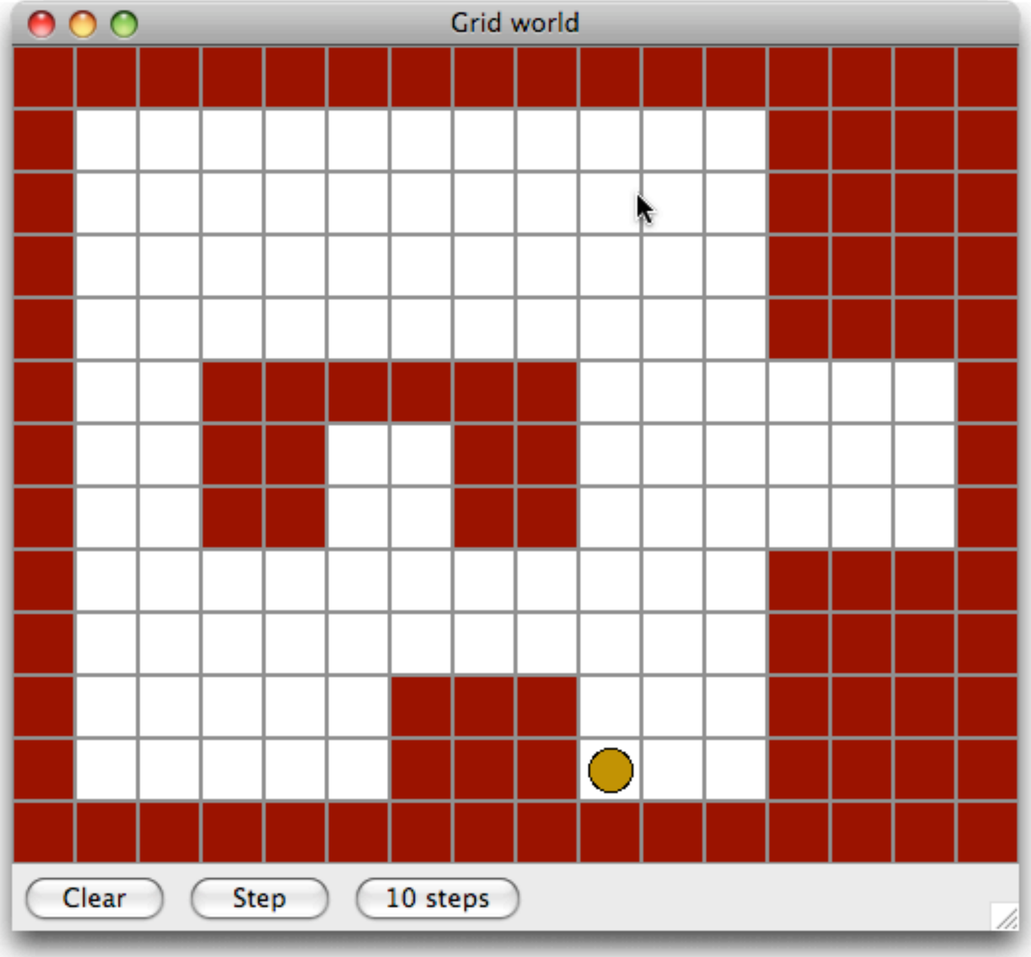
- Which direction will the robot attempt to go, if using this controller?



```
(define my-robot
    '(if s6 'north 'south))
```

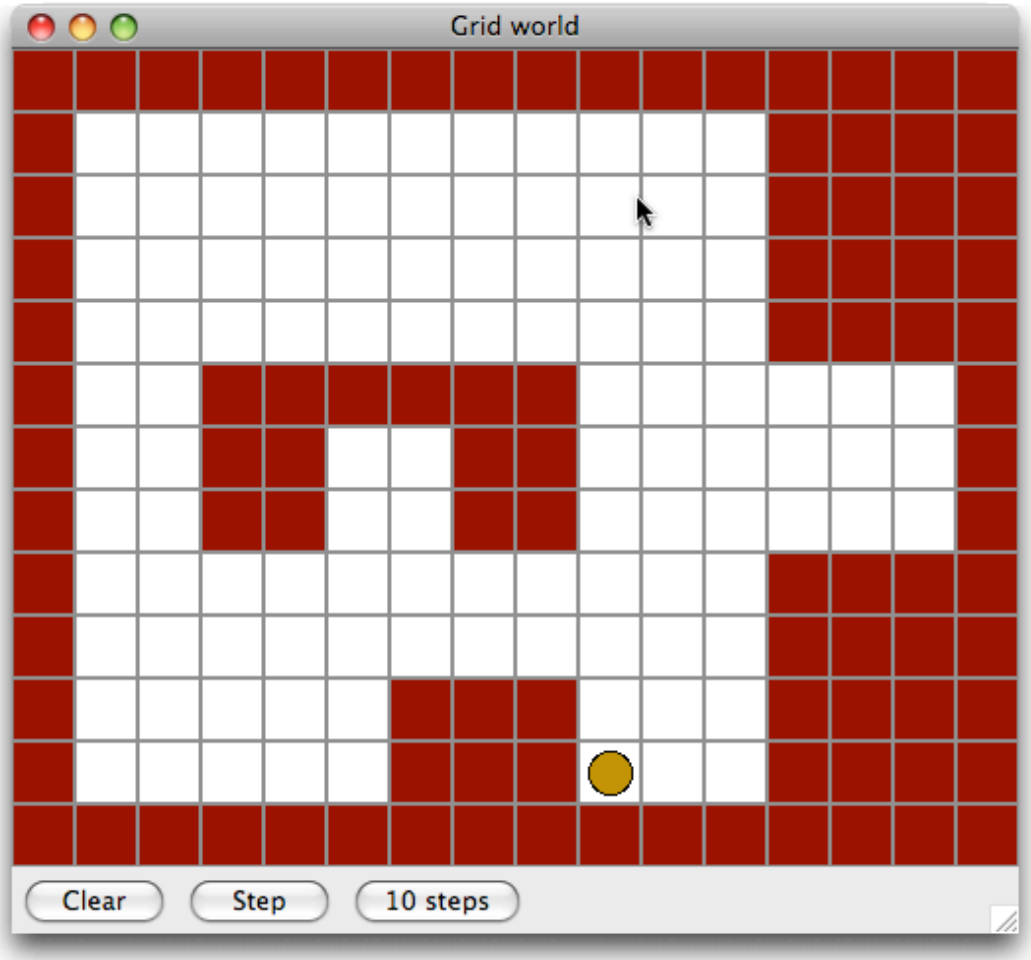
Analysis

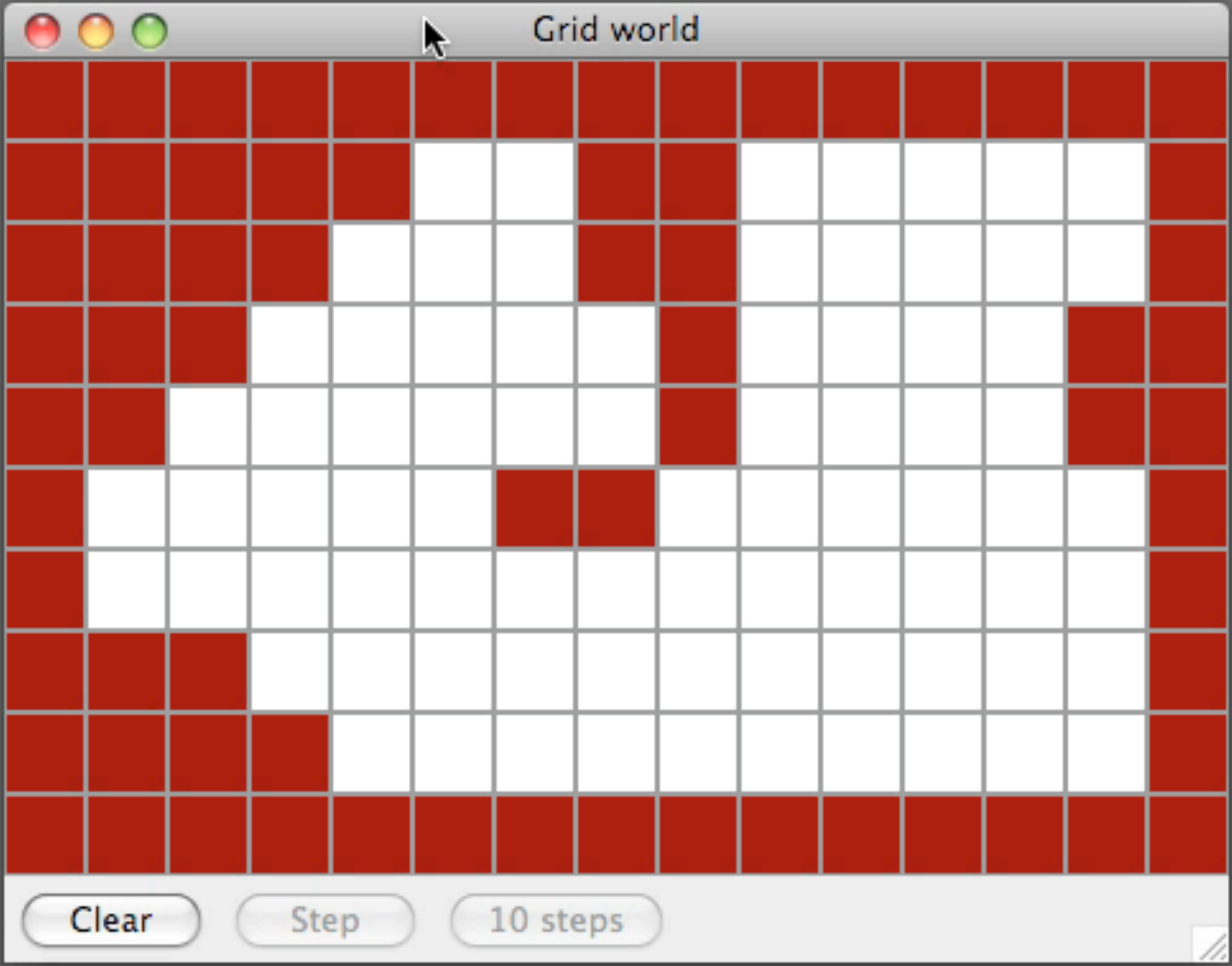
- Mark all the squares from which your robot should move north.
  - What features distinguish those squares from all the others?



Synthesis

- Compose and test your own robot controller

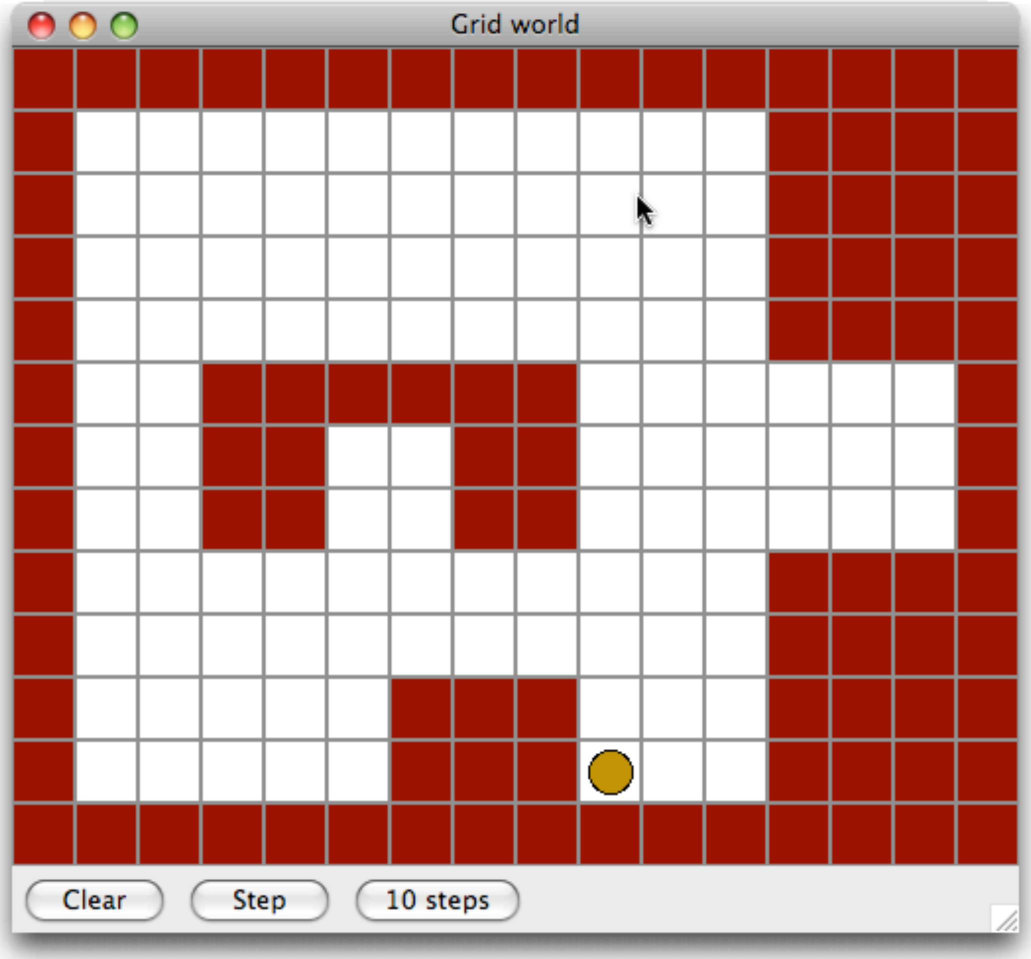



Grid world
Clear
Step
10 steps

Analysis Synthesis

- Determine how the map is specified, then design and test your own room

```
(define room-1
  '("           xxx"
    "           xxx"
    "           xxx"
    "           xxx"
    "  xxxxxx      "
    "  xx  xx      "
    "  xx  xx      "
    "           xxx"
    "           xxx"
    "     xxx   xxx"
    "     xxx   xxx"))
```

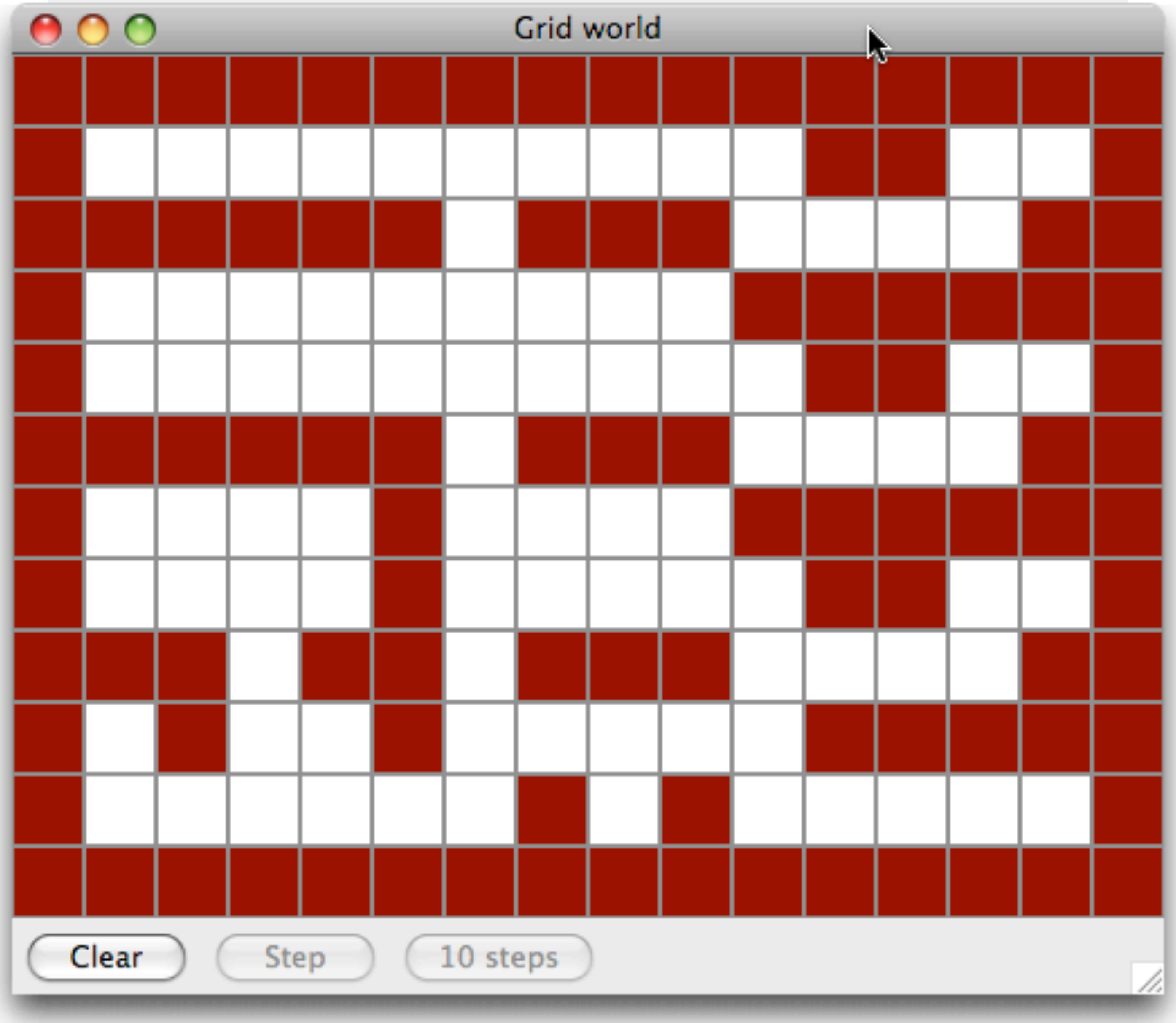


Evaluation

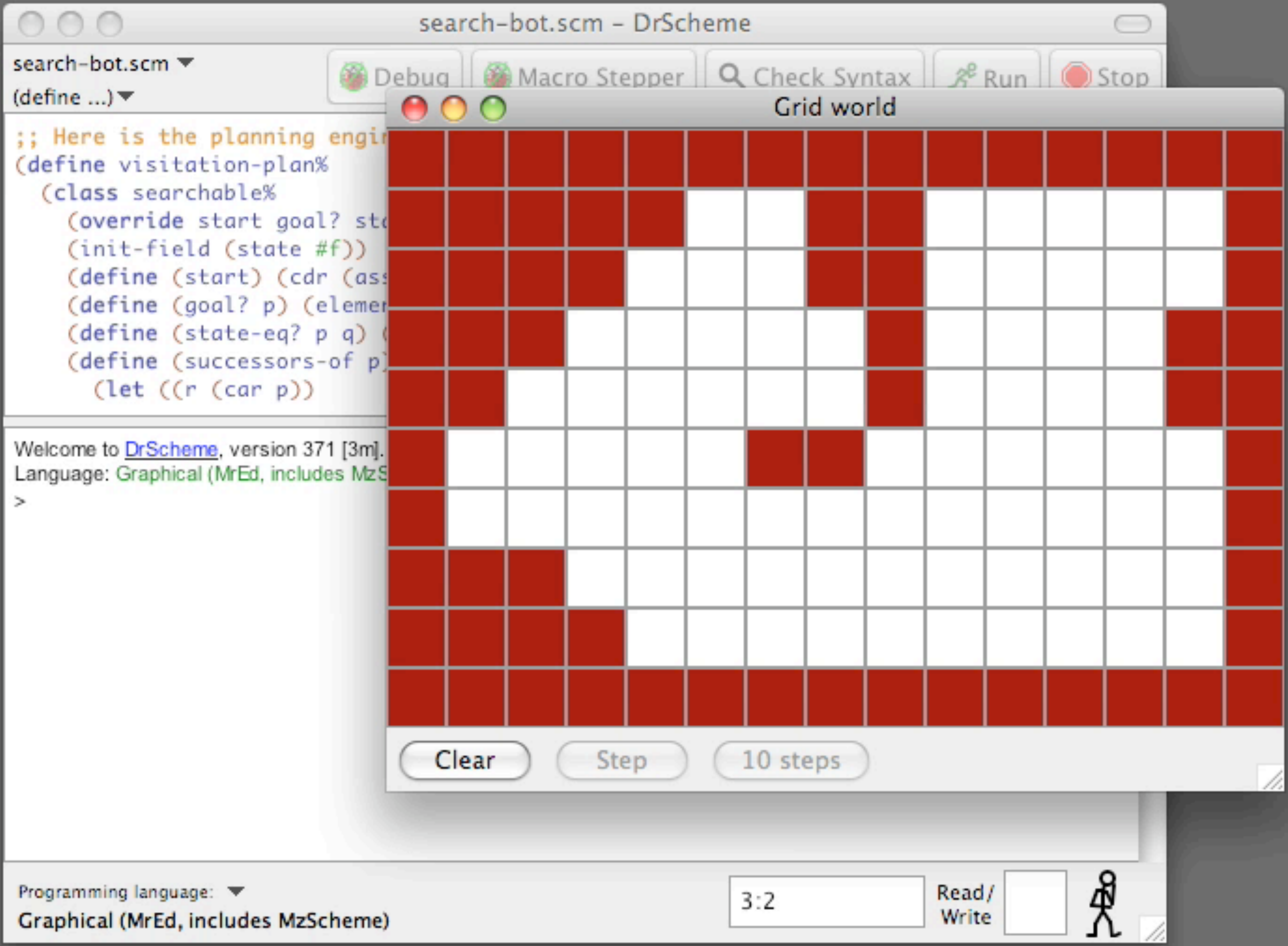
- What are some limitations of a stateless stimulus/response system?

Evaluation

- What are some limitations of a stateless stimulus/response system?



🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
🟥											🟥	🟥			🟥
🟥	🟥	🟥	🟥	🟥	🟥		🟥	🟥	🟥					🟥	🟥
🟥										🟥	🟥	🟥	🟥	🟥	🟥
🟥											🟥	🟥			🟥
🟥	🟥	🟥	🟥	🟥	🟥		🟥	🟥	🟥					🟥	🟥
🟥					🟥					🟥	🟥	🟥	🟥	🟥	🟥
🟥					🟥						🟥	🟥			🟥
🟥	🟥	🟥		🟥	🟥		🟥	🟥	🟥					🟥	🟥
🟥		🟥			🟥						🟥	🟥	🟥	🟥	🟥
🟥							🟥		🟥						🟥
🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥

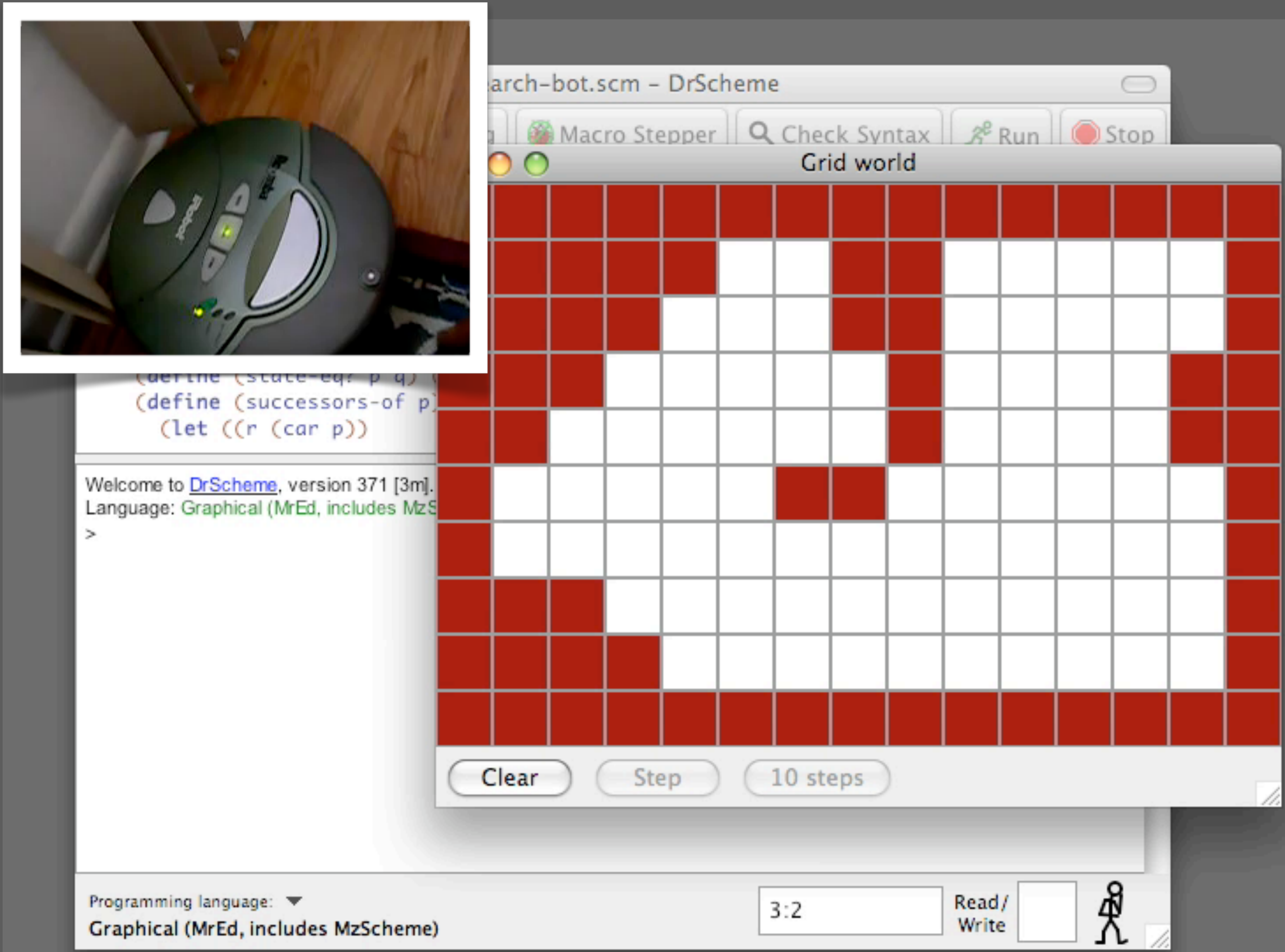


```
;; Here is the planning engi
(define visitation-plan%
  (class searchable%
    (override start goal? st
    (init-field (state #f))
    (define (start) (cdr (as
    (define (goal? p) (elemer
    (define (state-eq? p q) (
    (define (successors-of p)
      (let ((r (car p))
```

Welcome to DrScheme, version 371 [3m].
Language: Graphical (MrEd, includes MzS
>

Graphical (MrEd, includes MzScheme)

3:2



```
(define (state-eq? p q) (
(define (successors-of p)
  (let ((r (car p))
```

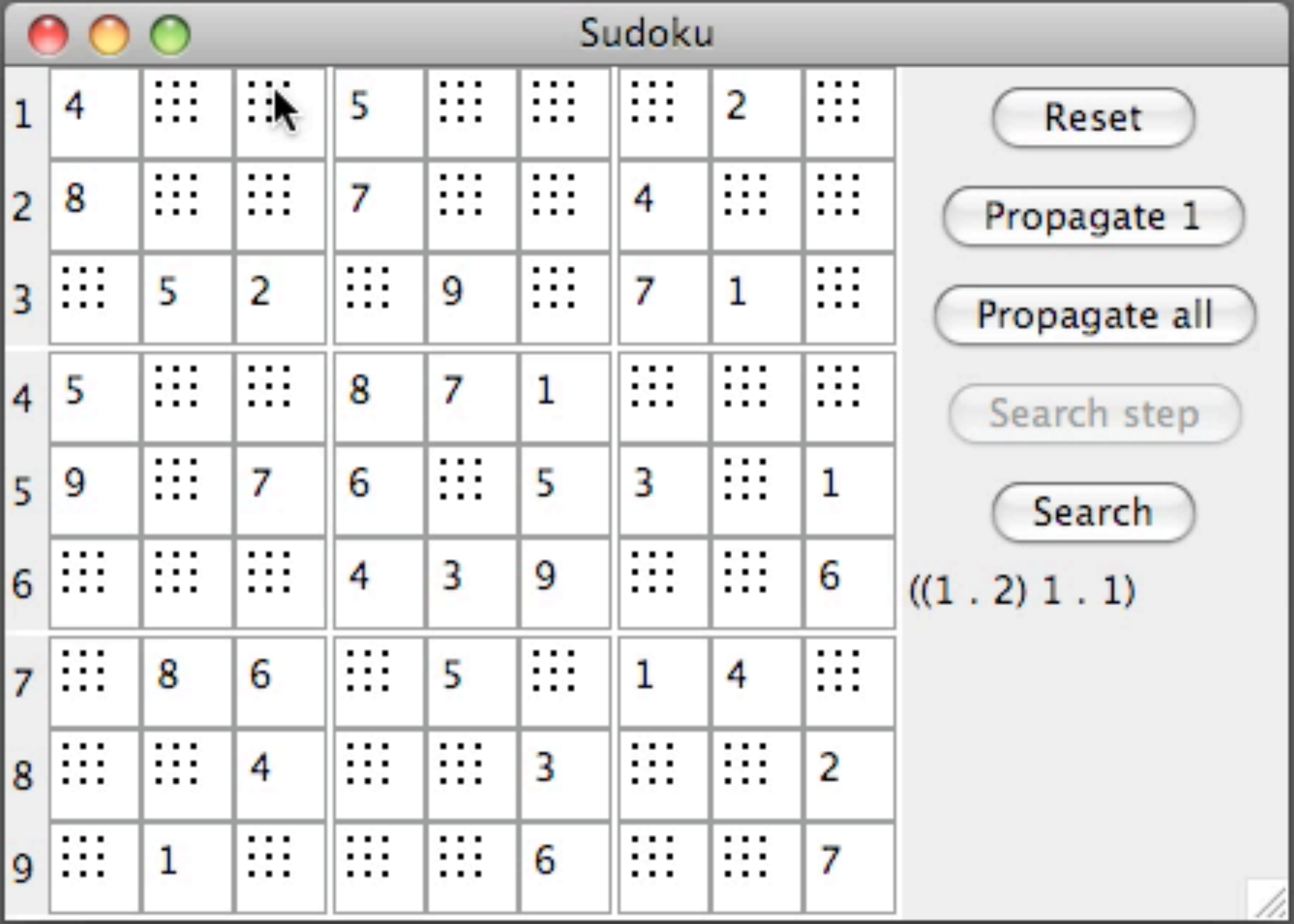
```
Welcome to DrScheme, version 371 [3m].
Language: Graphical (MrEd, includes MzS
>
```

Graphical (MrEd, includes MzScheme)

3:2

Constraint satisfaction

Constraint satisfaction



1	4	⋮⋮⋮	⋮⋮⋮	5	⋮⋮⋮	⋮⋮⋮	⋮⋮⋮	2	⋮⋮⋮
2	8	⋮⋮⋮	⋮⋮⋮	7	⋮⋮⋮	⋮⋮⋮	4	⋮⋮⋮	⋮⋮⋮
3	⋮⋮⋮	5	2	⋮⋮⋮	9	⋮⋮⋮	7	1	⋮⋮⋮
4	5	⋮⋮⋮	⋮⋮⋮	8	7	1	⋮⋮⋮	⋮⋮⋮	⋮⋮⋮
5	9	⋮⋮⋮	7	6	⋮⋮⋮	5	3	⋮⋮⋮	1
6	⋮⋮⋮	⋮⋮⋮	⋮⋮⋮	4	3	9	⋮⋮⋮	⋮⋮⋮	6
7	⋮⋮⋮	8	6	⋮⋮⋮	5	⋮⋮⋮	1	4	⋮⋮⋮
8	⋮⋮⋮	⋮⋮⋮	4	⋮⋮⋮	⋮⋮⋮	3	⋮⋮⋮	⋮⋮⋮	2
9	⋮⋮⋮	1	⋮⋮⋮	⋮⋮⋮	⋮⋮⋮	6	⋮⋮⋮	⋮⋮⋮	7

```
((1 . 2) 1 . 1)
```

1. **C’s** card has **higher** rank than **B’s** card

2. The **sum** of **C’s** card with **D’s** card is more than 8

3. **B’s** card is a **black** suit (clubs or spades)

4. **E’s** card is **not clubs**

5. **A’s** card is not the same suit as **C’s** card

KnowledgeComprehension

- Identify the unary constraints
- Identify the binary constraints

Application

- Apply the unary constraints to the hand you were dealt

- Draw a graph showing the binary relationships

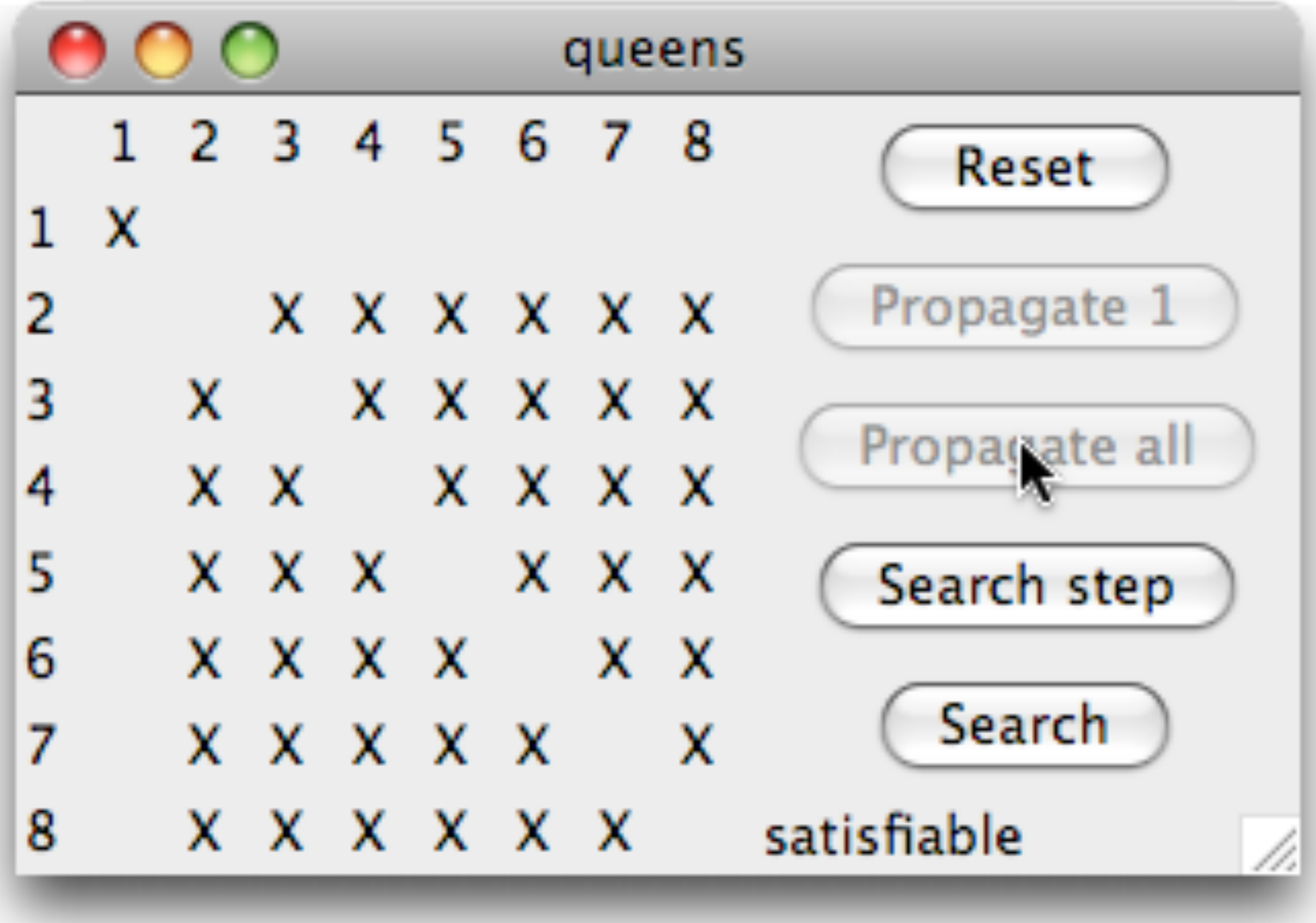
Analysis

- How many arcs are in the graph?

- When your hand changes, which arcs are added to the work list?

Synthesis

- Same process for 8-queens, but we follow through to implementation



	1	2	3	4	5	6	7	8
1	X							
2			X	X	X	X	X	X
3		X		X	X	X	X	X
4		X	X		X	X	X	X
5		X	X	X		X	X	X
6		X	X	X	X		X	X
7		X	X	X	X	X		X
8		X	X	X	X	X	X	

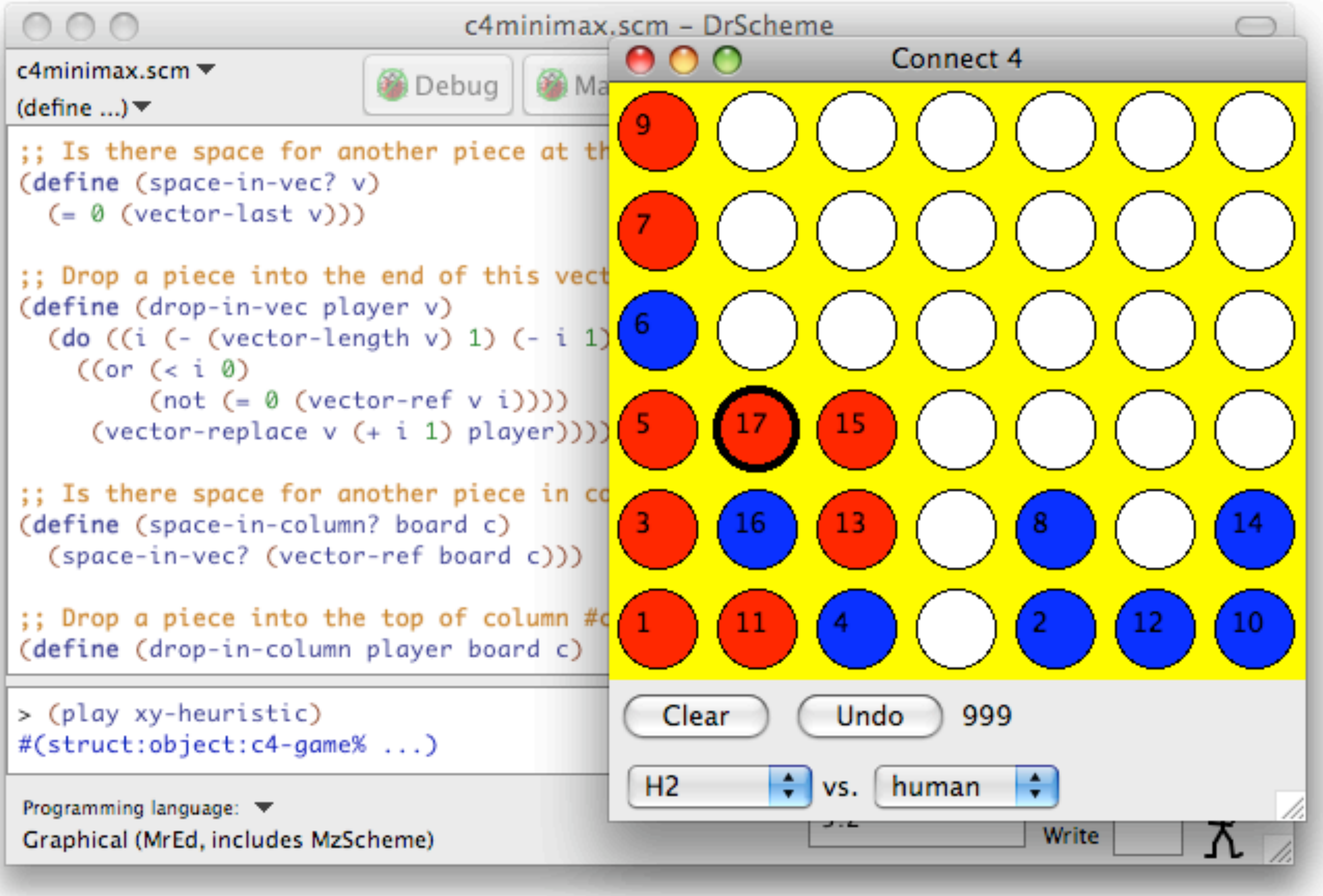
satisfiable

Topic Outline

Topic Outline

1. **Philosophical background,** strong vs. weak AI, Turing test, chat-bots
2. **Machine learning by example:** classification problems, decision trees, entropy, ID3
3. **Machine learning by evolution:** optimization problems and genetic algorithms
4. Planning using **uninformed and heuristic search:** breadth-first, depth-first, and A* algorithm
5. **Constraint propagation** and satisfaction with AC3
6. **Adversarial search** with minimax & heuristics
7. **Knowledge representation,** logic, expert systems, common sense

Connect 4



c4minimax.scm – DrScheme

```
;; Is there space for another piece at th
(define (space-in-vec? v)
  (= 0 (vector-last v)))

;; Drop a piece into the end of this vect
(define (drop-in-vec player v)
  (do ((i (- (vector-length v) 1) (- i 1)
    ((or (< i 0)
         (not (= 0 (vector-ref v i))))
     (vector-replace v (+ i 1) player))))

;; Is there space for another piece in co
(define (space-in-column? board c)
  (space-in-vec? (vector-ref board c)))

;; Drop a piece into the top of column #c
(define (drop-in-column player board c)
```

```
> (play xy-heuristic)
#(struct:object:c4-game% ...)
```

Programming language: Graphical (MrEd, includes MzScheme)

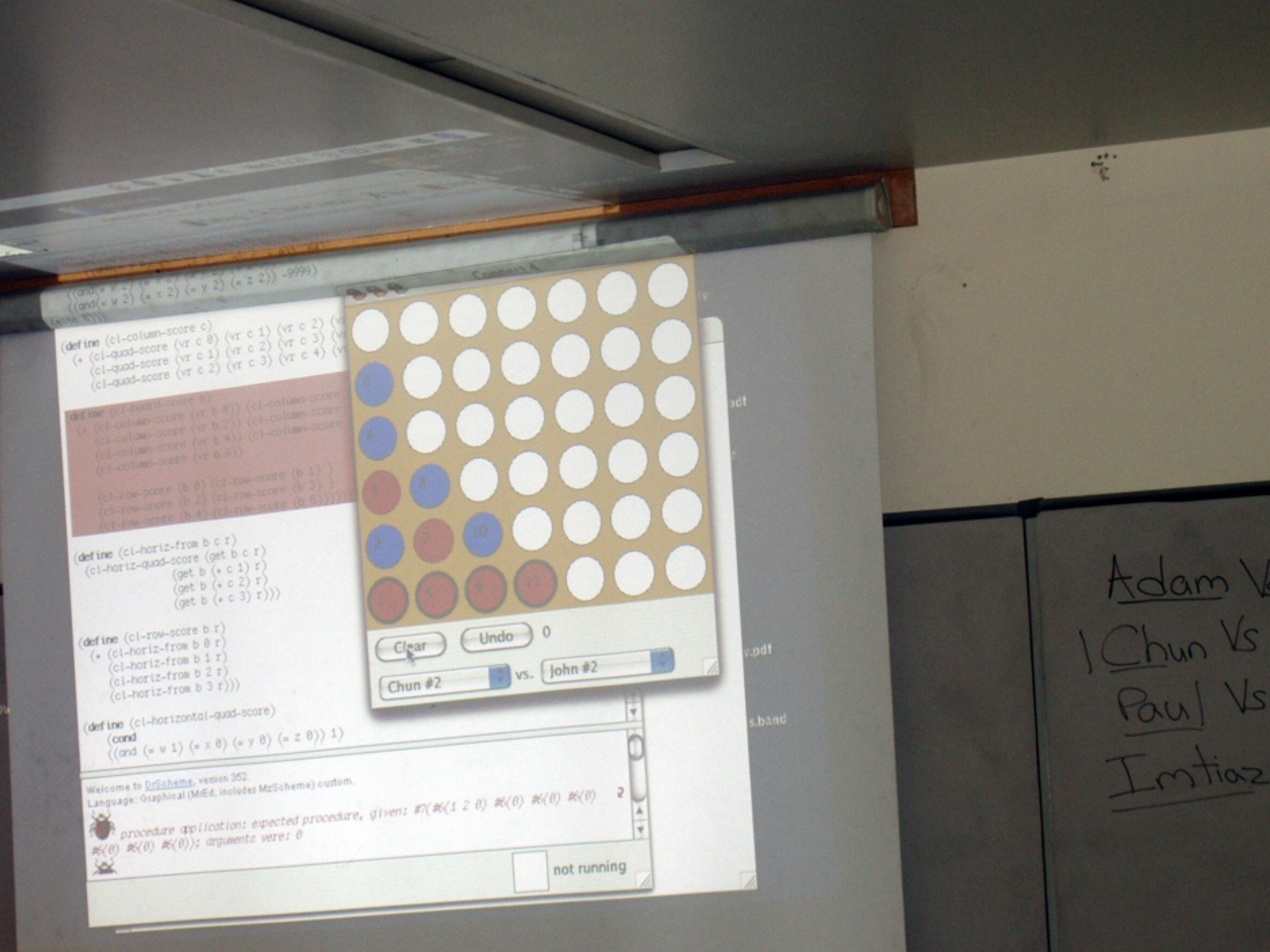
Connect 4

🟥 9	○	○	○	○	○	○
🟥 7	○	○	○	○	○	○
🟦 6	○	○	○	○	○	○
🟥 5	🟥 17	🟥 15	○	○	○	○
🟥 3	🟦 16	🟥 13	○	🟦 8	○	🟦 14
🟥 1	🟥 11	🟦 4	○	🟦 2	🟦 12	🟦 10

999

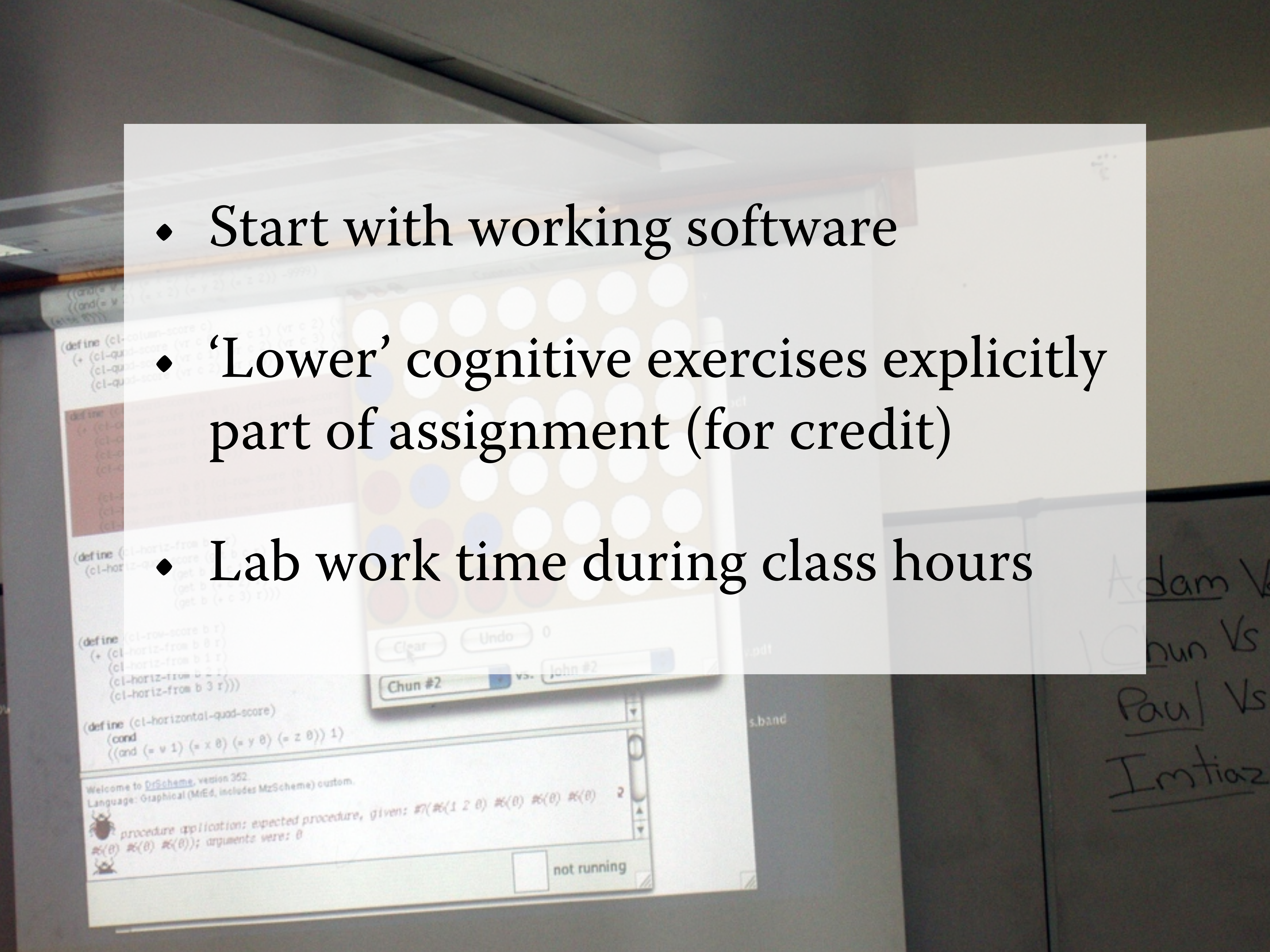
H2 vs. human

- Results: seems to work, more students submitting than usual
- “★★★★★”



```
  ((and(= w 2) (= x 2) (= y 2) (= z 2)) -9999)
 (else 0)))

(define (cl-column-score c)
  (+ (cl-quad-score (vr c 0) (vr c 1) (vr c 2) (v
     (cl-quad-score (vr c 1) (vr c 2) (vr c 3) (v
     (cl-quad-score (vr c 2) (vr c 3) (vr c 4) (v

(define (cl-board-score b)
  (+ (cl-column-score (vr b 0)) (cl-column-score
     (cl-column-score (vr b 2)) (cl-column-score
     (cl-column-score (vr b 4)) (cl-column-score
     (cl-column-score (vr b 6))

     (cl-row-score (b 0) (cl-row-score (b 1) )
     (cl-row-score (b 2) (cl-row-score (b 3) )
     (cl-row-score (b 4) (cl-row-score (b 5))))))

(define (cl-horiz-from b c r)
  (cl-horiz-quad-score (get b c r)
                       (get b (+ c 1) r)
                       (get b (+ c 2) r)
                       (get b (+ c 3) r)))

(define (cl-row-score b r)
  (+ (cl-horiz-from b 0 r)
     (cl-horiz-from b 1 r)
     (cl-horiz-from b 2 r)
     (cl-horiz-from b 3 r)))

(define (cl-horizontal-quad-score)
    (cond
    ((and (= w 1) (= x 0) (= y 0) (= z 0)) 1)
```

```
Welcome to DrScheme, version 352.
Language: Graphical (MrEd, includes MzScheme) custom.
procedure application: expected procedure, given: #7(#6(1 2 0) #6(0) #6(0) #6(0)
#6(0) #6(0) #6(0)); arguments were: 0
```

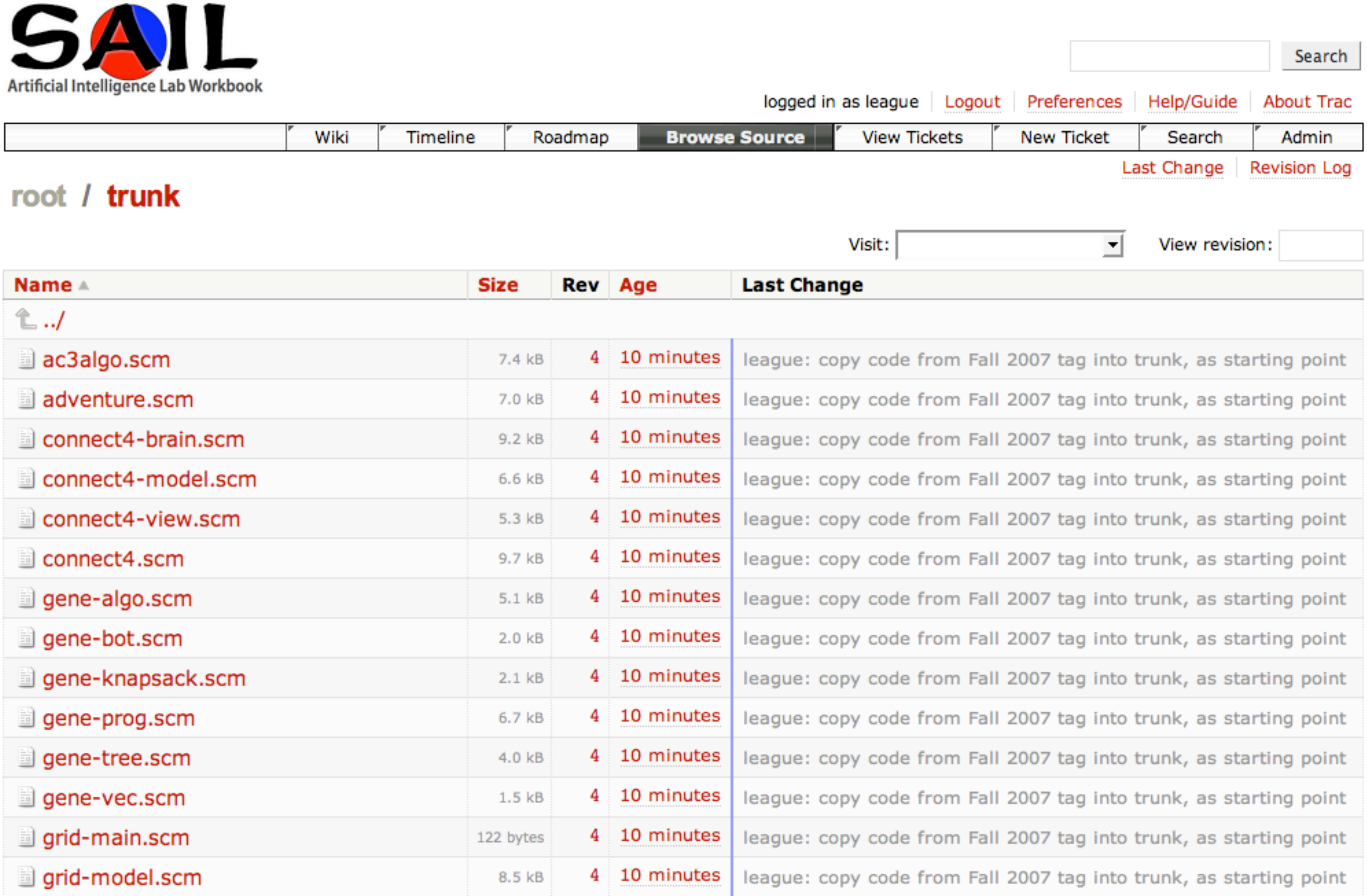


- Start with working software
- ‘Lower’ cognitive exercises explicitly part of assignment (for credit)
- Lab work time during class hours

```
(define (cl-horizontal-quad-score)
  (cond
  ((and (= v 1) (= x 0) (= y 0) (= z 0)) 1)
```

```
Welcome to DrScheme, version 352.
Language: Graphical (MrEd, includes MzScheme) custom.
procedure application: expected procedure, given: #7(#6(1 2 0) #6(0) #6(0) #6(0)
#6(0) #6(0) #6(0)); arguments were: 0
```

http://contrapunctus.net/sail/



root / trunk

Name ▲	Size	Rev	Age	Last Change
../				
ac3algo.scm	7.4 kB	4	10 minutes	league: copy code from Fall 2007 tag into trunk, as starting point
adventure.scm	7.0 kB	4	10 minutes	league: copy code from Fall 2007 tag into trunk, as starting point
connect4-brain.scm	9.2 kB	4	10 minutes	league: copy code from Fall 2007 tag into trunk, as starting point
connect4-model.scm	6.6 kB	4	10 minutes	league: copy code from Fall 2007 tag into trunk, as starting point
connect4-view.scm	5.3 kB	4	10 minutes	league: copy code from Fall 2007 tag into trunk, as starting point
connect4.scm	9.7 kB	4	10 minutes	league: copy code from Fall 2007 tag into trunk, as starting point
gene-algo.scm	5.1 kB	4	10 minutes	league: copy code from Fall 2007 tag into trunk, as starting point
gene-bot.scm	2.0 kB	4	10 minutes	league: copy code from Fall 2007 tag into trunk, as starting point
gene-knapsack.scm	2.1 kB	4	10 minutes	league: copy code from Fall 2007 tag into trunk, as starting point
gene-prog.scm	6.7 kB	4	10 minutes	league: copy code from Fall 2007 tag into trunk, as starting point
gene-tree.scm	4.0 kB	4	10 minutes	league: copy code from Fall 2007 tag into trunk, as starting point
gene-vec.scm	1.5 kB	4	10 minutes	league: copy code from Fall 2007 tag into trunk, as starting point
grid-main.scm	122 bytes	4	10 minutes	league: copy code from Fall 2007 tag into trunk, as starting point
grid-model.scm	8.5 kB	4	10 minutes	league: copy code from Fall 2007 tag into trunk, as starting point

```
http://contrapunctus.net/sail/
```

SAIL

Artificial Intelligence Lab Workbook

Christopher League

CCSC/NE

12 April 2008

LONG ISLAND UNIVERSITY