Inf2D Coursework 2

Rimvydas Rubavicius

The University of Edinburgh, School of Informatics

2022

Important Dates

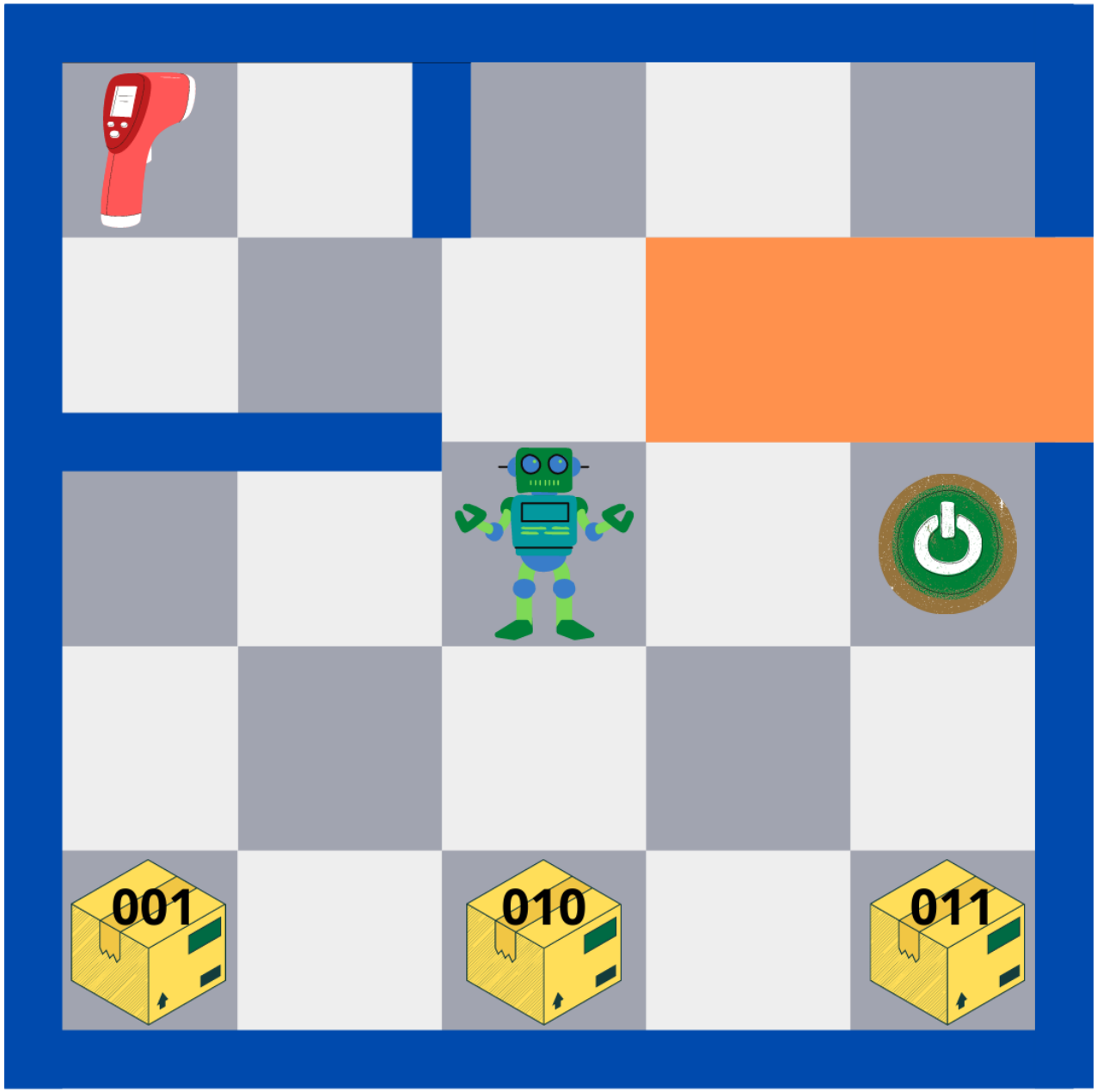
- Deadline : 3pm Thursday 31st March 2022
- Drop-in Lab Clinics: Fridays 11am–1pm, AT 6.06

Coursework Goals

- Formalize a reasonably sized planning problem
- Balance trade-offs in model design
- Actually implement and debug some PDDL

Domain Overview

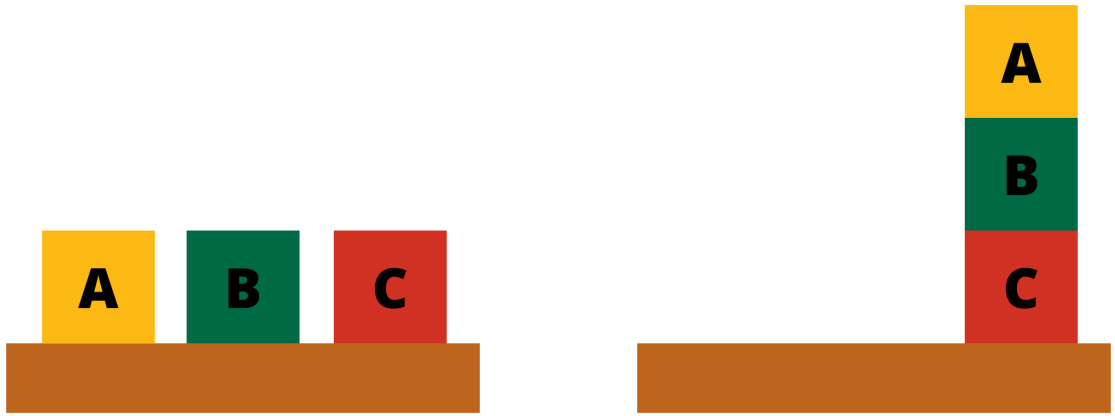
Domain Description



Coursework Tasks

- **Modelling**: formalization of the domain, given a specification.
- **Implementation**: writing your formalisation in PDDL and testing it on problem instances with the MetricFF planner.
- **Experiments**: designing an experiment to evaluate heuristics used by MetricFF planner.
- **Extensions**: extending the domain to deal with real world challenges.

Defining Initial and Goal State



Initial

$$On(A, Table) \wedge On(B, Table) \wedge On(C, Table) \wedge$$

$$\text{Block(A)} \wedge Block(B) \wedge Block(C) \wedge Clear(A) \wedge Clear(B) \wedge Clear(C)$$

Goal

$$On(A, B) \wedge On(B, C)$$

Defining Actions

Action(Move(block, from, to)):
PRECOND : On(block,from) $\wedge Clear(block) \wedge Clear(to) \wedge$
Block(block) $\wedge Block(to) \wedge$
(block $\neq$ *from*) $\wedge$ (*block* $\neq$ *to*) $\wedge$ (*from* $\neq$ *to*)
EFFECT : On(block, to) $\wedge \neg On(block, from) \wedge$
Clear (from) $\wedge \neg Clear(to)$

Implementation

Action(Move(b, x, y)):
PRECOND : On(b,x)
$\wedge Clear(b) \wedge Clear(y) \wedge$
Block(b) $\wedge Block(y) \wedge$
$(b \neq x) \wedge (b \neq y) \wedge (x \neq y)$
EFFECT : On(b, y) $\wedge \neg On(b, x) \wedge$
Clear (x) $\wedge \neg Clear(y)$

```
(:action MOVE
:parameters (
  ?b - block
  ?x - object
  ?y - block)
:precondition (and
  (On ?b ?x)
  (Clear ?b)
  (Clear ?y)
  (not (= ?b ?x))
  (not (= ?b ?y))
  (not (= ?x ?y)))
:effect (and
  (On ?b ?y)
  (Clear ?x)
  (not (On ?b ?x))
  (not (Clear ?y))))
```

Backwards Search

- $g_1 = On(A, B) \wedge On(B, C)$

Backwards Search

- $g_1 = On(A, B) \wedge On(B, C)$
- Available Actions: $Move(A, x, B)$, $Move(B, x, C)$

Backwards Search

- $g_1 = On(A, B) \wedge On(B, C)$
- Available Actions: $Move(A, x, B)$, $Move(B, x, C)$
- Choose : $Move(A, x, B)$

Backwards Search

- $g_1 = On(A, B) \wedge On(B, C)$
- Available Actions: $Move(A, x, B)$, $Move(B, x, C)$
- Choose : $Move(A, x, B)$
- $g_2 = On(A, x) \wedge Clear(A) \wedge Clear(B) \wedge Block(A) \wedge Block(B) \wedge A \neq x \wedge A \neq B \wedge x \neq B \wedge On(B, C)$

Backwards Search

- $g_1 = On(A, B) \wedge On(B, C)$
- Available Actions: $Move(A, x, B)$, $Move(B, x, C)$
- Choose : $Move(A, x, B)$
- $g_2 = On(A, x) \wedge Clear(A) \wedge Clear(B) \wedge Block(A) \wedge Block(B) \wedge A \neq x \wedge A \neq B \wedge x \neq B \wedge On(B, C)$
- Available actions : $Move(B, x', C)$, $Move(x', B, y)$, $Move(x', A, y)$, $Move(A, x', x)$

Backwards Search

- $g_1 = On(A, B) \wedge On(B, C)$
- Available Actions: $Move(A, x, B)$, $Move(B, x, C)$
- Choose : $Move(A, x, B)$
- $g_2 = On(A, x) \wedge Clear(A) \wedge Clear(B) \wedge Block(A) \wedge Block(B) \wedge A \neq x \wedge A \neq B \wedge x \neq B \wedge On(B, C)$
- Available actions : $Move(B, x', C)$, $Move(x', B, y)$, $Move(x', A, y)$, $Move(A, x', x)$
- Choose : $Move(B, x', C)$

Backwards Search

- $g_1 = On(A, B) \wedge On(B, C)$
- Available Actions: $Move(A, x, B)$, $Move(B, x, C)$
- Choose : $Move(A, x, B)$
- $g_2 = On(A, x) \wedge Clear(A) \wedge Clear(B) \wedge Block(A) \wedge Block(B) \wedge A \neq x \wedge A \neq B \wedge x \neq B \wedge On(B, C)$
- Available actions : $Move(B, x', C)$, $Move(x', B, y)$, $Move(x', A, y)$, $Move(A, x', x)$
- Choose : $Move(B, x', C)$
- $g_3 = On(A, x) \wedge Clear(A) \wedge Clear(B) \wedge Block(A) \wedge Block(B) \wedge A \neq x \wedge A \neq B \wedge x \neq B \wedge On(B, x') \wedge Clear(B) \wedge Clear(C) \wedge Block(C) \wedge B \neq x' \wedge B \neq C \wedge x' \neq C$

Backwards Search

- $g_1 = On(A, B) \wedge On(B, C)$
- Available Actions: $Move(A, x, B)$, $Move(B, x, C)$
- Choose : $Move(A, x, B)$
- $g_2 = On(A, x) \wedge Clear(A) \wedge Clear(B) \wedge Block(A) \wedge Block(B) \wedge A \neq x \wedge A \neq B \wedge x \neq B \wedge On(B, C)$
- Available actions : $Move(B, x', C)$, $Move(x', B, y)$, $Move(x', A, y)$, $Move(A, x', x)$
- Choose : $Move(B, x', C)$
- $g_3 = On(A, x) \wedge Clear(A) \wedge Clear(B) \wedge Block(A) \wedge Block(B) \wedge A \neq x \wedge A \neq B \wedge x \neq B \wedge On(B, x') \wedge Clear(B) \wedge Clear(C) \wedge Block(C) \wedge B \neq x' \wedge B \neq C \wedge x' \neq C$
- g_3 satisfies initial state by substituting $\{x = Table, x' = Table\}$

Backwards Search

- $g_1 = On(A, B) \wedge On(B, C)$
- Available Actions: $Move(A, x, B)$, $Move(B, x, C)$
- Choose : $Move(A, x, B)$
- $g_2 = On(A, x) \wedge Clear(A) \wedge Clear(B) \wedge Block(A) \wedge Block(B) \wedge A \neq x \wedge A \neq B \wedge x \neq B \wedge On(B, C)$
- Available actions : $Move(B, x', C)$, $Move(x', B, y)$, $Move(x', A, y)$, $Move(A, x', x)$
- Choose : $Move(B, x', C)$
- $g_3 = On(A, x) \wedge Clear(A) \wedge Clear(B) \wedge Block(A) \wedge Block(B) \wedge A \neq x \wedge A \neq B \wedge x \neq B \wedge On(B, x') \wedge Clear(B) \wedge Clear(C) \wedge Block(C) \wedge B \neq x' \wedge B \neq C \wedge x' \neq C$
- g_3 satisfies initial state by substituting $\{x = Table, x' = Table\}$
- Done! Final Plan: $Move(B, Table, C)$, $Move(A, Table, B)$

Additional Resources

- MetricFF planner: `https://fai.cs.uni-saarland.de/hoffmann/metric-ff.html`
- Planning wiki: `https://planning.wiki/`

Running the Planner Locally

```
[selby]s1557887: ls -l
total 1770
-rw------- 1 s1557887 people     701 Dec 10  2018 EXAMPLE-blocks-world-domain.pddl
-rw------- 1 s1557887 people     334 Dec 10  2018 EXAMPLE-blocks-world-problem.pddl
-rw------- 1 s1557887 people 1810256 Dec  2  2019 ff
[selby]s1557887: chmod +x ff
[selby]s1557887: ls -l
total 1770
-rw------- 1 s1557887 people     701 Dec 10  2018 EXAMPLE-blocks-world-domain.pddl
-rw------- 1 s1557887 people     334 Dec 10  2018 EXAMPLE-blocks-world-problem.pddl
-rwx------ 1 s1557887 people 1810256 Dec  2  2019 ff
[selby]s1557887: ./ff -o EXAMPLE-blocks-world-domain.pddl -f EXAMPLE-blocks-world-problem.pddl

ff: parsing domain file
domain 'BLOCKS-WORLD' defined
 ... done.
ff: parsing problem file
problem 'BLOCK-PROBLEM' defined
 ... done.


no metric specified. plan length assumed.

checking for cyclic := effects --- OK.

ff: search configuration is EHC, if that fails then  best-first on 1*g(s) + 5*h(s) where
    metric is  plan length

Cueing down from goal distance:    2 into depth [1]
                                   1           [1]
                                   0

ff: found legal plan as follows

step    0: MOVE B TABLE C
        1: MOVE A TABLE B


time spent:    0.00 seconds instantiating 18 easy, 0 hard action templates
               0.00 seconds reachability analysis, yielding 13 facts and 18 actions
               0.00 seconds creating final representation with 13 relevant facts, 0 relevant fluents
               0.00 seconds computing LNF
               0.00 seconds building connectivity graph
               0.00 seconds searching, evaluating 4 states, to a max depth of 1
               0.00 seconds total time
```

Submission Organization

```
Inf2d-cw2-s<matric>
├── domain-ext-1.pddl
├── domain-ext-2.pddl
├── domain-ext-3.pddl
├── domain.pddl
├── problem-1.pddl
├── problem-2.pddl
├── problem-3.pddl
├── problem-ext-1.pddl
├── problem-ext-2.pddl
├── problem-ext-3.pddl
└── report.pdf
```

Failure to comply with this specification will result in reduction of 5 marks awarded for your submission.

Compress, Validate, Submit

Compress

```
tar -cvzf Inf2d-ass2-s1234567.tar.gz
Inf2d-ass2-s1234567
```

Check your archive file!

```
tar -tf Inf2d-ass2-s1234567.tar.gz
ls -l Inf2d-ass2-s1234567.tar.gz
```

Submit via LEARN