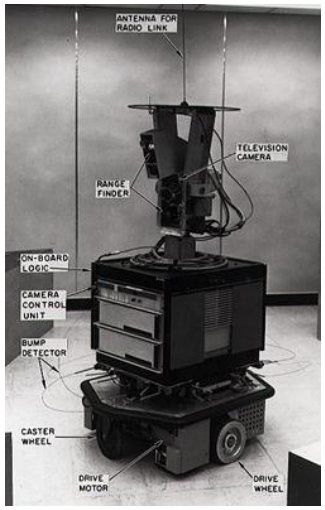
PDDL and Planners

PAH (Planning and Games)

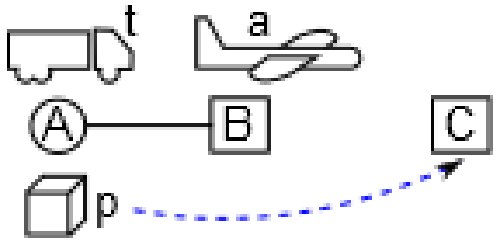
Michal Štolba
michal.stolba@agents.fel.cvut.cz

STRIPS (Stanford Research Institute Problem Solver)



- 1971 – Shakey the Robot
- <P,O,I,G>
  - P – finite set of propositional (true/false) variables
  - O – finite set of operators:
    - pre(p∈P s.t. p=true; p∈P s.t. p=false)
    - eff(p∈P s.t. p←true; p∈P s.t. p← false)
  - I – initial state (p∈P s.t. p=true, other false)
  - G – goal state (p∈P s.t. p=true; p∈P s.t. p=false)
- Set representation
  - True/false determined by the set membership
- Plan existence PSPACE-Complete

STRIPS - Example



- P = {at-t-A, at-t-B, at-a-B, at-a-C, at-p-A, at-p-B, at-p-C, in-p-t, in-p-a}
  - 2^9 = 512 states
- O – operators:
  - load-p-a-B
    - pre: at-a-B=true, at-p-B=true
    - eff: in-p-a=true, at-p-B=false
- I – initial state: at-t-A=true, …
- G – goal state: at-p-C=true

Multi-valued Planning Task (MPT or SAS+)

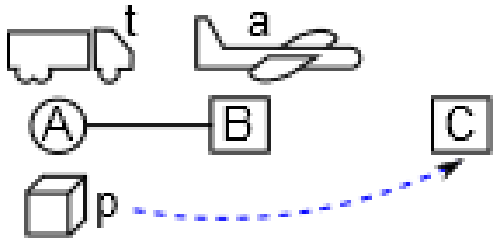
- 1995 (SAS+), 2005 (MPT – Fast Downward)
- $<V,i,g,O>$
  - V – finite set of state variables v with associated finite domain D_v
    - partial state over V is a function s over some subset of V s.t. $s(v) \in D_v$ whenever $s(v)$ is defined
    - state is a partial state s.t. s is defined for all $v \in V$
  - i – state over V called initial state
  - g – partial state over V called goal state

Multi-valued Planning Task (continued)

- $<V,i,g,O>$
  - O – finite set of operators <pre,eff>
    - pre: partial assignment (state) over V
    - eff: $<cond, v, d>$
      - $cond$: (possibly empty) partial assignment over V
      - $v \in V$ – affected variable
      - $d \in D_v$ – new value for v

- Plan existence PSPACE-complete
- Automatic conversion from STRIPS

MPT - Example



- *V* – variables and their domains:
  - truck-at ∈ {A,B}
  - plane-at ∈ {B,C}
  - package-at ∈ {A,B,C,t,a}
  - 2x2x5 = 20 states
- O – operators:
  - load-p-a-B
    - pre: plane-at=B, package-at=B
    - eff: <{}, package-at, a>

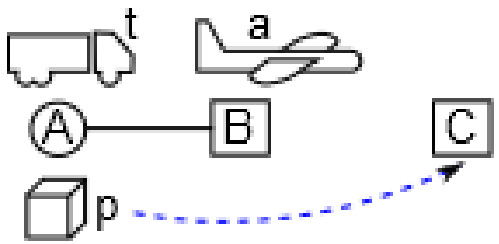
PDDL (Planning Domain Definition Language)

- General language to describe planning problems
  - Domain – definition of types, predicates, operators
  - Problem – definition of objects, initial state and goal
  - Lisp-like syntax
    - Prefix notation (+ 1 2)
    - A lot of brackets
  - Several versions (1.2, 2.1, 3.1)

Assignment 1

PDDL Excercise

- Formalize:



- Run planners
- Extend

Planners (1)

- FF (Fast Forward) not included
  - Forward-chaining heuristic state space search
  - Enforced hill-climbing / Breadth-first search
  - FF heuristic
- FD-autotune
  - Fast Downward
    - MPT, several search strategies, new heuristics
  - Automatic configuration

Planners (2)

- Lama 2008
  - Built on FD
  - Multi-heuristic search (FF, Landmarks)
  - Weighted A*
- POPF2
  - Forward-chaining partial-order temporal planner
- PROBE
  - Greedy best-first search
  - Greedy probes with highly informed heuristics

Planners (3)

- Roamer
  - Based on FD
  - Random-walk assisted Greedy BFS
    - Random walks to escape heuristic plateaus
- SGPlan6
  - Parallel decomposition and FF