Situation Calculus

Dr. Neil T. Dantam

CSCI-534, Colorado School of Mines

Spring 2020

Introduction

Definition: Situation Calculus

A logical representation of domains that change over time according to **actions** that my be performed.

Outcomes

- Know definitions situation calculus elements
- Know the Planning Domain Definition Language (PDDL) syntax
- Create situation calculus / PDDL representations of planning scenarios

Outline

Logical Calculi

Propositional Calculus:

- Boolean variables (propositions)
- Logical Operators ($\wedge$, $\vee$, $\neg$, $\implies$, $\iff$, $\oplus$)

Predicate Calculus: Extends the propositional calculus with:

- Objects
- Predicates
- Functions
- Quantifiers

Situation Calculus: Extends the predicate calculus to model actions that change state:

- Fluents
- Actions

Situation Calculus

Predicate Calculus + changing state:

Fluents

- Synonym for state variables of the system
- Example:
  - `closed(suitcase)`
  - `contains(suitcase, laptop)`
- From Latin *fluere* meaning "to flow."

Actions

- **Elements:**

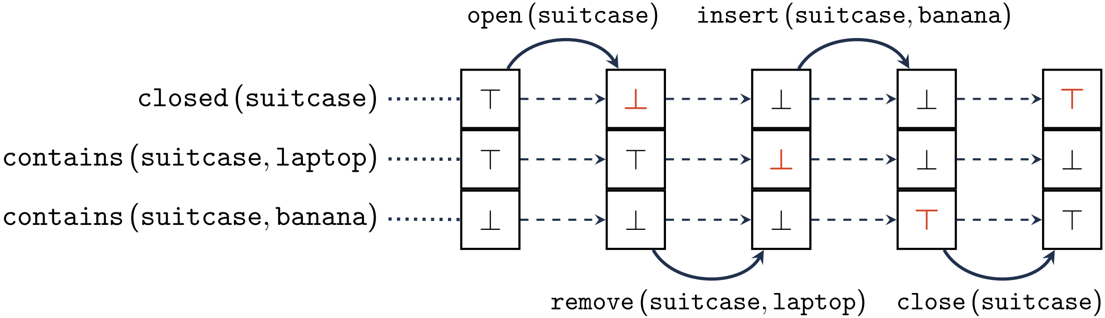
  Label: Name / arguments
  Precondition: States where the action is valid
  Effect: Result of the action

- **Example:**

  Label: `open(suitcase)`
  Precondition: `closed(suitcase)`
  Effect: $\neg$`closed(suitcase)`

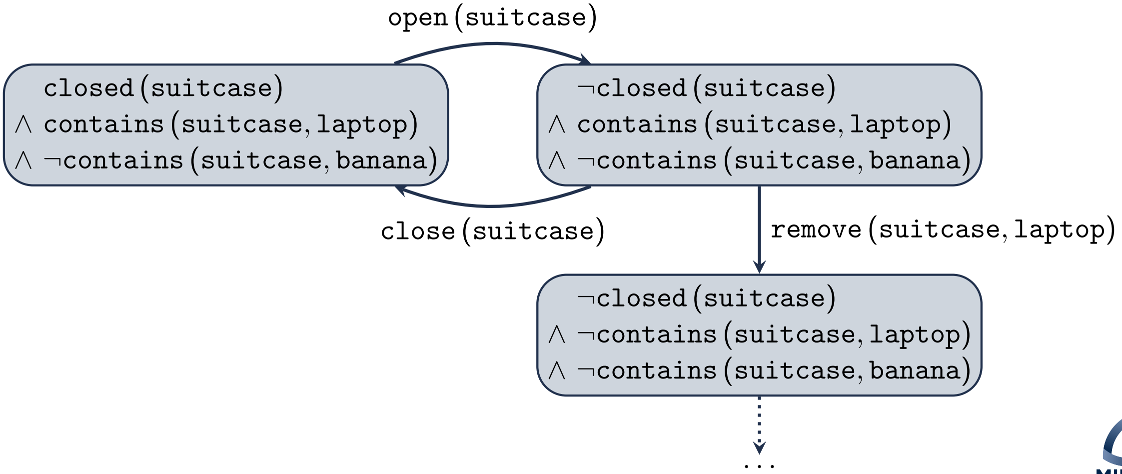
Illustration

State/Action Sequence



Illustration

Automaton



Exercise: State Space

Objects:
- $C = \{\texttt{suitcase}, \texttt{backpack}\}$
- $B = \{\texttt{laptop}, \texttt{banana}, \texttt{book}\}$

Predicate: $\texttt{contains} : C \times B \mapsto \mathbb{B}$

Fluents:

States:

Transition System

State Space: $\mathcal{Q} = f_0 \times f_1 \times \ldots \times f_m$, for each fluent f_i

Actions: $\mathcal{U} = \{a_0, \ldots, a_n\}$

Transitions: $\delta : \mathcal{Q} \times \mathcal{U} \mapsto \mathcal{Q}$,
where for $\delta(q_0, a) = q_1$,

- q_0 satisfies the precondition of a
- q_1 is the effect of a applied to q_0

Start: $s \in \mathcal{Q}$ is the initial state

Goal: $G \subseteq \mathcal{Q}$ is the set of goal states

The (Classical) Planning Problem

Given: Transition System $A = (\mathcal{Q}, \mathcal{U}, \delta, s, G)$

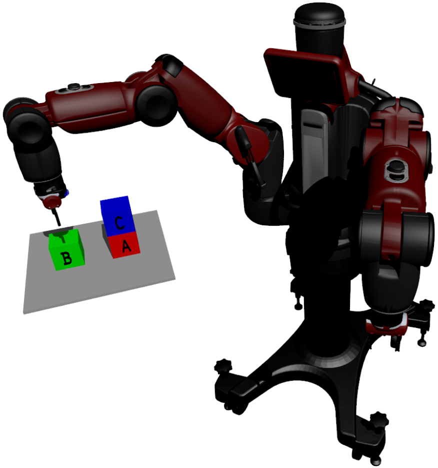
Find: A valid plan $P = (a_0, \ldots, a_n)$, such that:

- Plan begins in start state: $q_0 = s$
- Successive actions are valid transitions: $q_{i+1} = \delta(q_i, a_i)$
- Plan ends in a goal state: $q_{n_1} \in G$

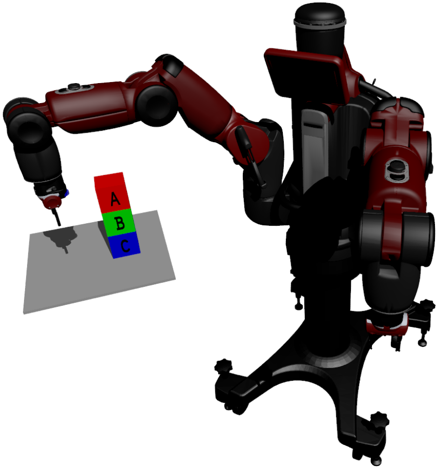
Outline

A Planning Problem

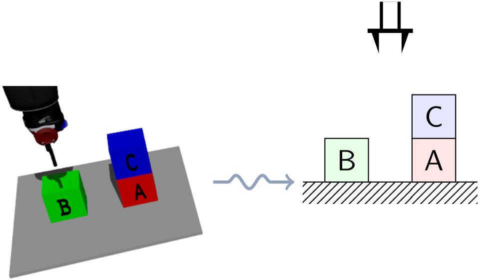
Start



Goal



First-Order Logic Description



Constants: A, B, C

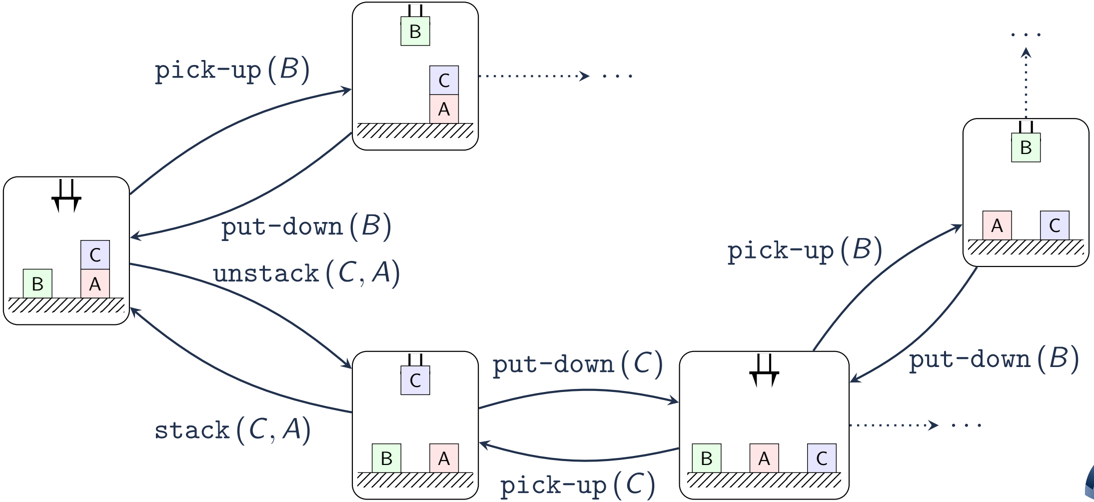
Predicates:

- $\texttt{on}(?x, ?y)$
- $\texttt{clear}(?x)$
- $\texttt{ontable}(?x)$
- $\texttt{handempty}()$

Fluents:

- $\texttt{clear}(B)$
- $\texttt{clear}(C)$
- $\texttt{ontable}(B)$
- $\texttt{ontable}(A)$
- $\texttt{handempty}()$

Task Language

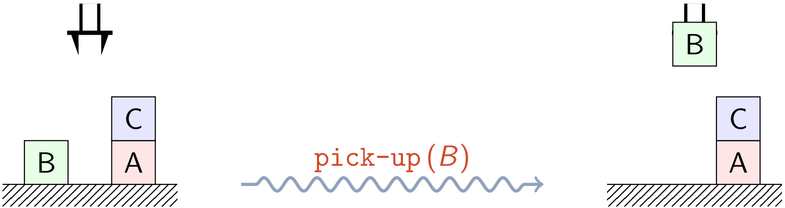


Example: Effects

$\texttt{pick-up}\,(?x)$

Precondition: $\texttt{ontable}\,(?x) \land \texttt{clear}\,(?x) \land \texttt{handempty}\,()$

Effect: $\neg\texttt{ontable}\,(?x) \land \neg\texttt{clear}\,(?x) \land \neg\texttt{handempty}\,() \land \texttt{holding}\,(?x)$



$$\begin{aligned} & \texttt{ontable}\,(B) \land \texttt{ontable}\,(A) \\ \land\ & \texttt{on}\,(C, A) \\ \land\ & \texttt{clear}\,(B) \land \texttt{clear}\,(C) \\ \land\ & \texttt{handempty}\,() \end{aligned}$$

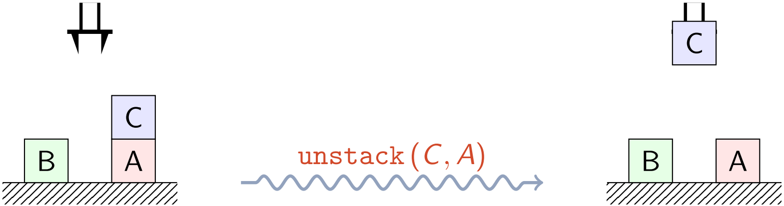
$$\begin{aligned} & \cancel{\texttt{ontable}\,(B) \land} \ \texttt{ontable}\,(A) \\ \land\ & \texttt{on}\,(C, A) \\ & \cancel{\land\ \texttt{clear}\,(B)} \land \texttt{clear}\,(C) \\ & \cancel{\land\ \texttt{handempty}}\,() \\ \land\ & \texttt{holding}\,(B) \end{aligned}$$

Exercise: Effects

$\texttt{unstack}(?x, ?y)$

Precondition: $\texttt{on}(?x, ?y) \land \texttt{clear}(?x) \land \texttt{handempty}()$

Effect: $\neg\texttt{on}(?x, ?y) \land \neg\texttt{clear}(?x) \land \neg\texttt{handempty}() \land \texttt{holding}(?x) \land \texttt{clear}(?y)$



$$
\begin{aligned}
& \texttt{ontable}(B) \land \texttt{ontable}(A) \\
\land\ & \texttt{on}(C, A) \\
\land\ & \texttt{clear}(B) \land \texttt{clear}(C) \\
\land\ & \texttt{handempty}()
\end{aligned}
$$

Outline

Logic and Planning

Blocksworld Domain

Planning Domain Definition Language (PDDL)
- Operators
- Facts

Planning Approaches
- Heuristic Search
- Constraint-Based Planning

Example: PDDL Action

$\texttt{pick-up}(?x)$

$\texttt{pick-up}(?x)$

Precondition: $\texttt{ontable}(?x)$
$\land\ \texttt{clear}(?x)$
$\land\ \texttt{handempty}()$

Effect: $\neg\texttt{ontable}(?x)$
$\land\ \neg\texttt{clear}(?x)$
$\land\ \neg\texttt{handempty}()$
$\land\ \texttt{holding}(?x)$

PDDL

```
(:action pick-up
        :parameters (?x)
        :precondition (and (ontable ?x)
                           (clear ?x)
                           (handempty))
        :effect (and (not (ontable ?x))
                     (not (clear ?x))
                     (not (handempty))
                     (holding ?x)))
```

Exercise: PDDL Action

$\texttt{unstack}\,(?x, ?y)$

$\texttt{unstack}\,(?x, ?y)$

PDDL

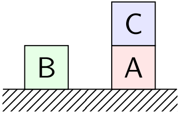
Precondition: $\texttt{on}\,(?x, ?y) \wedge \texttt{clear}\,(?x) \wedge \texttt{handempty}\,()$

Effect: $\neg\texttt{on}\,(?x, ?y) \wedge \neg\texttt{clear}\,(?x) \wedge \neg\texttt{handempty}\,() \wedge \texttt{holding}\,(?x) \wedge \texttt{clear}\,(?y)$

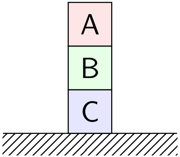
Full Operators File

Example: PDDL Facts

Start



Goal

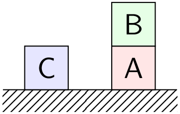


PDDL

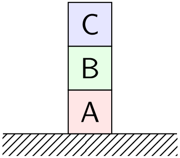
```
(define
 (problem sussman-anomaly)
  (:domain blocks)
  (:objects a b c)
  (:init (on c a)
         (ontable a)
         (ontable b)
         (clear c)
         (clear b)
         (handempty))
  (:goal (and (on b c)
              (on a b))))
```

Exercise: PDDL Facts

Start



Goal



PDDL

Outline

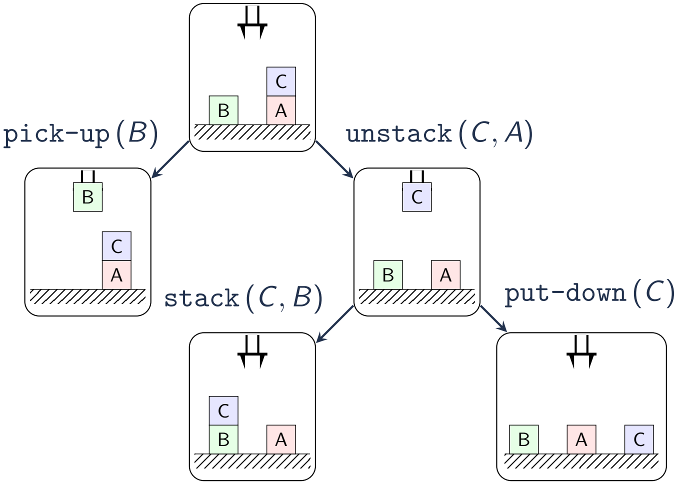
Logic and Planning

Blocksworld Domain

Planning Domain Definition Language (PDDL)
- Operators
- Facts

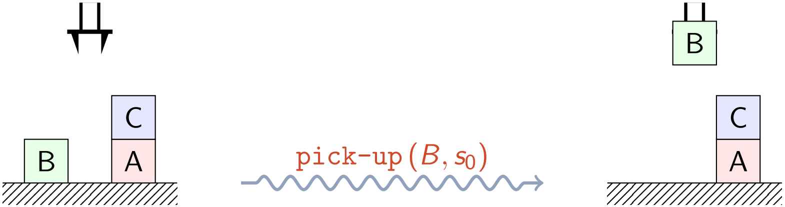
Planning Approaches
- Heuristic Search
- Constraint-Based Planning

Heuristic Search



Constraint-Based Planning

aka SATPlan



$$\texttt{pick-up}\,(B, s_0) \implies \overbrace{\texttt{ontable}\,(?x, s_0) \wedge \texttt{clear}\,(?x, s_0) \wedge \texttt{handempty}\,(s_0)}^{\text{precondition at step } i}$$

$$\wedge\ \underbrace{\neg\texttt{ontable}\,(?x, s_1) \wedge \neg\texttt{clear}\,(?x, s_1) \wedge \neg\texttt{handempty}\,(s_1) \wedge \texttt{holding}\,(?x, s_1)}_{\text{effect at step } i+1}$$

Summary

Logic and Planning

Blocksworld Domain

Planning Domain Definition Language (PDDL)
- Operators
- Facts

Planning Approaches
- Heuristic Search
- Constraint-Based Planning

References

Textbook: Russell & Norvig.

- Ch 10.1 Definition of Classical Planning

Textbook: Lavalle

- Ch 2.4 Using Logic to Formulate Discrete-Planning