CS 4649/7649 RIP
Robot Intelligence: Planning

Introduction

Sungmoon Joo

Institute for Robotics & Intelligent Machines
School of Interactive Computing
Georgia Institute of Technology

What is Planning?

Given:
- States
- Actions
- Initial State and Goal State
- Constraints

Task 1 (eg. Classical Planning):
Find a sequence of actions that take you from Init. to Goal

Task 2 (eg. Policy, Controls):
Find actions that take you from any state to Goal

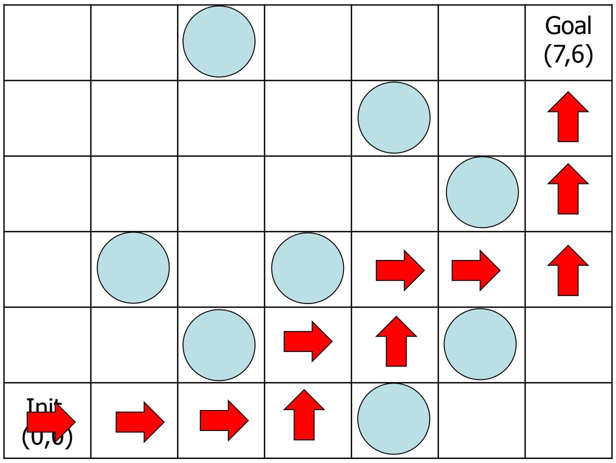
Task 3 (eg. MDP, POMDP):
Decide the best action to take now in order to improve your odds of reaching Goal

Task 4 (eg. Motion Planning):
Find a continuous path (in state space) that takes you from Init. to Goal

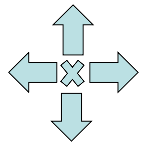
…

What is Planning?

Sequence of Actions

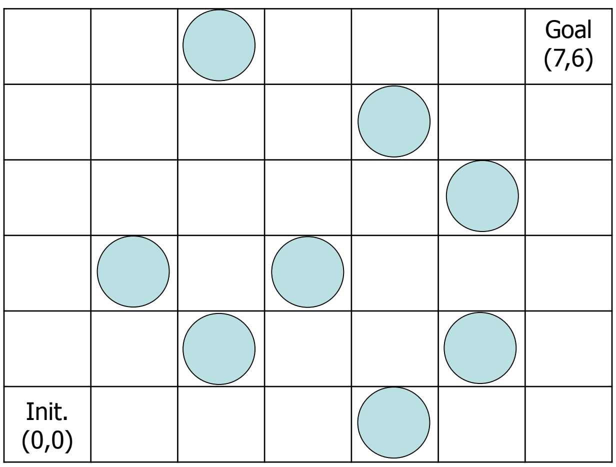


Init., Goal
States
Actions
Constraints

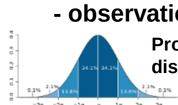


What is Planning?

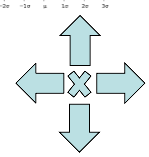
MDP,POMDP



Init., Goal
States
Actions
Constraints
Uncertainties
- action
- observation

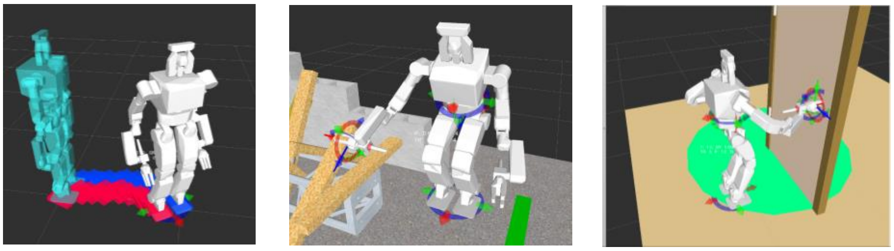


Probability distribution



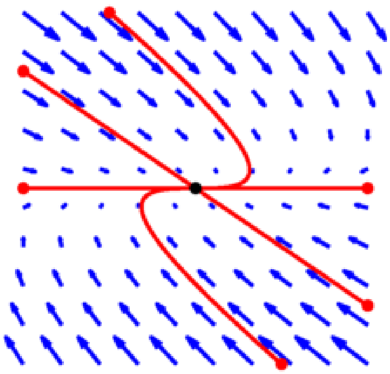
What is Planning?

Motion Planning



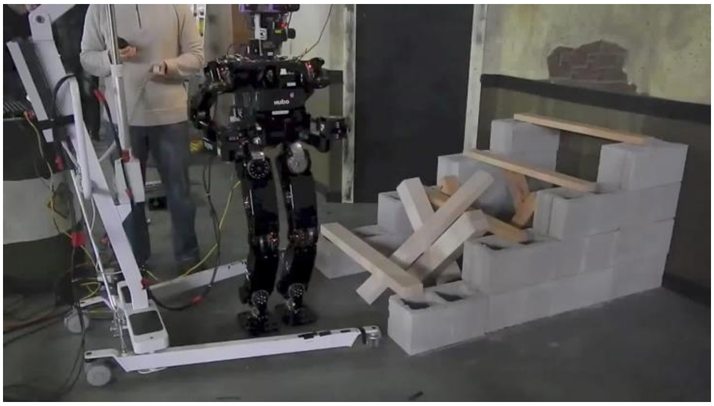
What is Planning?

Control/Policy

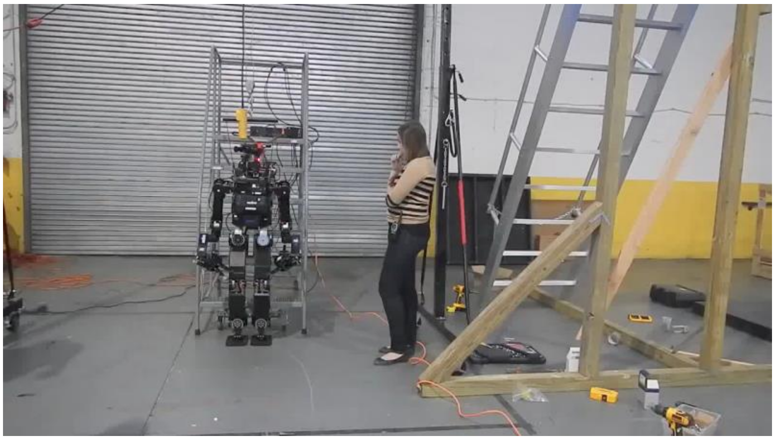


Find actions that take you from any state to Goal

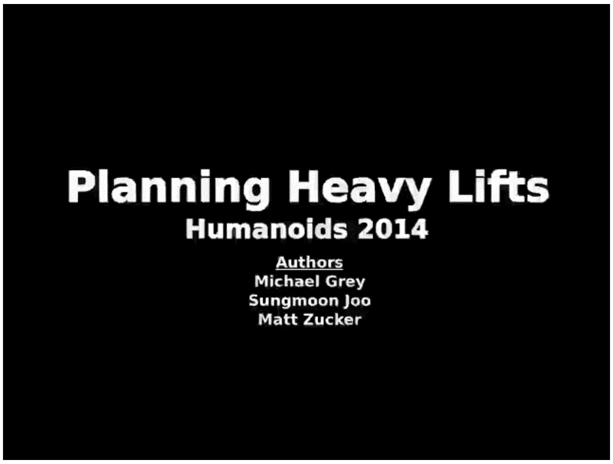
What is Planning?



What is Planning?



What is Planning?



Course Overview

Learn Tools/Frameworks:
- **Classical Planning**
- **Motion Planning**
- **MDP, POMDP**
- **Controls**

Apply to Real-world Problems:
- **Project**

Grading: 2 Group HWs + Final Project
- **Classical Planning (30%)**
- **Motion Planning (30%)**
- **Project (40%) – Grad(CS7649) / Undergrad(CS4649)**

Schedule
- **Sep. 16, Nov. 18/20 – TBD (IROS, Humanoids conference)**
- **Oct. 14, Nov. 27 – No Class (Fall Recess, Thanksgiving)**
- **Project Presentation – Dec 4. or Final exam day**

*** Course website: https://joosm.github.io/RIP2014**

Representation of Information: Predicate Logic

Motivation:

- Historically the first (McCarthy, Nilsson, Fikes)
- **More efficient than explicitly representing states**
- Only gives initial state, operators generate the rest as needed
- **Easier to analyze due to simplicity and generality**

First Representation: Predicate Logic

Statement → Predicate

(Unary) Predicate: P(x)
Introduce a (functional) symbol(P) for the predicate, and put the subject(x) as an argument to the functional symbol.

N-ary predicate is defined similarly

Examples:
- **x is happy → Happy(x): Unary predicate**
- **The suitcase contains a bomb → Contains(Suitcase, Bomb): Binary predicate**
- **X is less than Y → LessThan(x,y):Binary predicate**
- **$P(x_1,x_2,\ldots,x_n)$: n-ary predicate**

First Representation: Predicate Logic

Objects (Constants): $a, 123, house, mike, robot, suitcase$

Variables: $x, y, z \ldots$

Relations (Predicates): $LessThan, Contains, Parent, Happy$

Connectives: $\neg, \vee, \wedge, \Rightarrow$

Any expression is either true or false: $LessThan(1, 2)$

$\neg Contains(Suitcase, Bomb)$

Truth Table

Negation

A	B	$\neg A$			
F	F	T			
F	T	T			
T	F	F			
T	T	F			

First Representation: Predicate Logic

Objects (Constants): $a, 123, house, mike, robot, suitcase$

Variables: $x, y, z \ldots$

Relations (Predicates): $LessThan, Contains, Parent, Happy$

Connectives: $\neg, \vee, \wedge, \Rightarrow$

Any expression is either true or false: $LessThan(1,2)$

$$\neg Contains(Suitcase, Bomb)$$

$$Working(Robot) \wedge Has(Robot, Intelligence)$$

Truth Table

AND/Conjunction

A	B	$\neg A$	$A \wedge B$		
F	F	T	F		
F	T	T	F		
T	F	F	F		
T	T	F	T		

First Representation: Predicate Logic

Objects (Constants): $a, 123, house, mike, robot, suitcase$

Variables: $x, y, z \ldots$

Relations (Predicates): $LessThan, Contains, Parent, Happy$

Connectives: $\neg, \vee, \wedge, \Rightarrow$

Any expression is either true or false:

$$Planning(Robot) \Rightarrow Working(Robot)$$

Truth Table

Logical Implication

A	B	$\neg A$	$A \wedge B$	$A \Rightarrow B$	
F	F	T	F	T	
F	T	T	F	T	
T	F	F	F	F	
T	T	F	T	T	

First Representation: Predicate Logic

Objects (Constants): $a, 123, house, mike, robot, suitcase$

Variables: $x, y, z \ldots$

Relations (Predicates): $LessThan, Contains, Parent, Happy$

Connectives: $\neg, \vee, \wedge, \Rightarrow$

Any expression is either true or false:

$$Planning(Robot) \Rightarrow Working(Robot)$$

Equivalent

$$\neg Planning(Robot) \vee Working(Robot)$$

Truth Table

Negation+OR/Disjuction

A	B	$\neg A$	$A \wedge B$	$A \Rightarrow B$	$\neg A \vee B$
F	F	T	F	T	T
F	T	T	F	T	T
T	F	F	F	F	F
T	T	F	T	T	T

Predicate Logic: Proofs

Represent a World using Axioms(i.e. predicates) → Prove (by deduction)

- Axioms: $Contains(Suitcase, Laptop), Contains(Laptop, Slides)$
  $Contains(Slides, Movies)$

  $\forall x, y, z(Contains(x, y) \wedge Contains(y, z) \Rightarrow Contains(x, z))$

- Prove: $Contains(Suitcase, Movies)$

1) $Contains(Suitcase, Laptop) \wedge Contains(Laptop, Slides)$
2) $Contains(Suitcase, Laptop) \wedge Contains(Laptop, Slides)$
   $\Rightarrow Contains(Suitcase, Slides)$

Predicate Logic: Proofs

- Axioms: $Contains(Suitcase, Laptop), Contains(Laptop, Slides)$
  $Contains(Slides, Movies)$

  $\forall x, y, z(Contains(x, y) \wedge Contains(y, z) \Rightarrow Contains(x, z))$

- Prove: $Contains(Suitcase, Movies)$

1) $Contains(Suitcase, Laptop) \wedge Contains(Laptop, Slides)$
2) $Contains(Suitcase, Laptop) \wedge Contains(Laptop, Slides)$
   $\Rightarrow Contains(Suitcase, Slides)$
3) $Contains(Suitcase, Slides)$

Truth Table

A	B	$\neg A$	$A \wedge B$	$A \Rightarrow B$	$\neg A \vee B$
F	F	T	F	T	T
F	T	T	F	T	T
T	F	F	F	F	F
T	T	F	T	T	T

Predicate Logic: Proofs

- Axioms: $Contains(Suitcase, Laptop), Contains(Laptop, Slides)$
  $Contains(Slides, Movies)$

  $\forall x, y, z(Contains(x, y) \wedge Contains(y, z) \Rightarrow Contains(x, z))$

- Prove: $Contains(Suitcase, Movies)$

1) $Contains(Suitcase, Laptop) \wedge Contains(Laptop, Slides)$
2) $Contains(Suitcase, Laptop) \wedge Contains(Laptop, Slides) \Rightarrow Contains(Suitcase, Slides)$
3) $Contains(Suitcase, Slides)$
4) $Contains(Suitcase, Slides) \wedge Contains(Slides, Movies) \Rightarrow Contains(Suitcase, Movies)$
5) $Contains(Suitcase, Movies)$

Up to now, the world is static…

Situation Calculus*

To represent and reason about dynamical worlds

To represent 'change', 'state' is introduced.

- **Fluents** = Add state (time?) variable to relations:

$$Contains(Suitcase, Laptop, \mathbf{S0})$$
$$Working(Robot, \mathbf{S7})$$

- **Actions** are *reified* functions of constants. (They can be treated as constants themselves)

$$Put(Laptop, Suitcase) \quad Open(Car) \quad Lock(Car)$$

- The **do function**: $do(\alpha, \sigma_0) \rightarrow \sigma_1$

$$\alpha = action$$
$$\sigma = state$$

***Modern version is different from the original for clarity. Calculus = study about 'change'.**

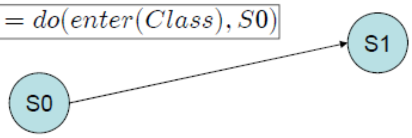
Situation Calculus

$$Room(rm)$$

- Effect Axioms (Positive + Negative)

$$In(Robot, Hall, s) \wedge Open(rm, s) \Rightarrow In(Robot, rm, \mathbf{do}(\mathbf{enter}(\mathbf{rm}), \mathbf{s}))$$
$$In(Robot, Hall, s) \wedge Open(rm, s) \Rightarrow \neg In(Robot, Hall, \mathbf{do}(\mathbf{enter}(\mathbf{rm}), \mathbf{s}))$$

$S1 = do(enter(Class), S0)$



$In(Robot, Class, S1)$
$\neg In(Robot, Hall, S1)$

$In(Robot, Hall, S0)$
$Open(Class, S0)$
$Open(Office, S0)$

Situation Calculus

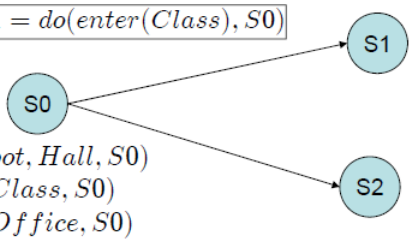
$Room(rm)$

- Effect Axioms (Positive + Negative)

$$In(Robot, Hall, s) \wedge Open(rm, s) \Rightarrow In(Robot, rm, \mathbf{do}(\mathbf{enter}(\mathbf{rm}), \mathbf{s}))$$

$$In(Robot, Hall, s) \wedge Open(rm, s) \Rightarrow \neg In(Robot, Hall, \mathbf{do}(\mathbf{enter}(\mathbf{rm}), \mathbf{s}))$$

$S1 = do(enter(Class), S0)$



$In(Robot, Class, S1)$
$\neg In(Robot, Hall, S1)$

$In(Robot, Hall, S0)$
$Open(Class, S0)$
$Open(Office, S0)$

$In(Robot, Office, S2)$
$\neg In(Robot, Hall, S2)$

$S2 = do(enter(Office), S0)$