# CS 4649/7649 RIP
# Robot Intelligence: Planning

## Classical Planning

**Sungmoon Joo**

**School of Interactive Computing**
**College of Computing**
**Georgia Institute of Technology**



# Course Info.

- **Course Website: joosm.github.io/RIP2014**
- **Course Wiki: Teaming & Collaboration. Under construction**
- **Email me(sungmoon.joo@cc.gatech.edu) to introduce yourself**
- **Issues with registration?**

# What is Planning?

**Planning:**
**"devising a plan of action to achieve one's goal" (Russel & Norvig)**

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

**Task 1: Find a sequence of actions that take you from Init. to Goal**

**Task 2: Find actions that take you from any state to Goal**

**Task 3: Decide the best action to take now in order to improve your odds of reaching Goal**

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

**…**



# DARPA's Assessment

# State of the Art




# First Representation: Predicate Logic

**Unless, some how, we can describe the world, we cannot devise a plan.**

**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**
- **$P(x_1,x_2,\dots,x_n)$: n-ary predicate**

***Proposition can be considered as a 0-nary predicate.**
***Predicate Logic ~ First Order Logic(FOL)**

## 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

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

# Situation Calculus*

**To represent and reason about dynamical worlds**

**To represent 'change', 'state (implicitly time)' is introduced.**

- **Fluents = Aspects of the world that change**

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

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

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

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

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

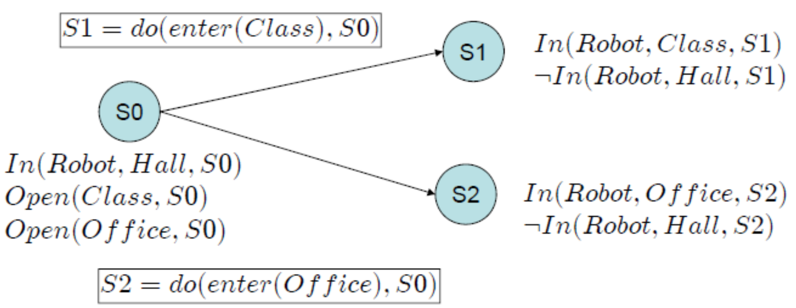
***Modern version is different from the original for clarity. Calculus = study about 'change'.**



# Situation Calculus

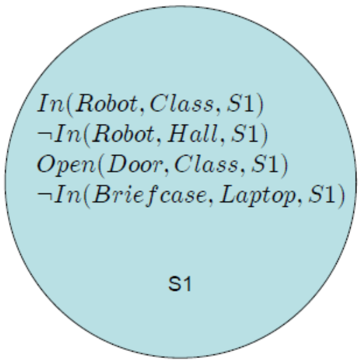
- Effect axioms describe how a world changes by an action
- Effect Axioms (Positive + Negative) : Specify the outcome of an action

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



## Motivational Detour: Implicit State Space

**How many distinct states can we describe with n fluents?**






## Implicit State Space Representation

How many distinct states can we describe with n fluents?

- One fluent: 2 states
- Two fluents: 2 x 2 = 4 states
- n fluents = $2^n$ states

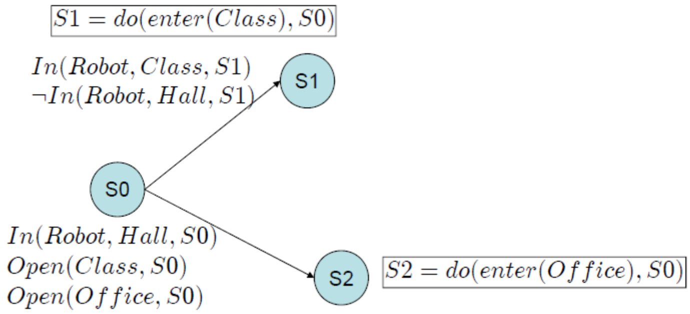
| $Open(Door, s)$ | $Hungry(Robot, s)$ |
|---|---|
| T | T |
| T | F |
| F | T |
| F | F |

Number of states is exponential in number of fluents.

Recall that fluents are themselves propositions over constants!

## Situation Calculus: A Plan

$In(Robot, rm, s) \Rightarrow Educated(Students, rm, \mathbf{do}(\mathbf{teach}(\mathbf{rm}), \mathbf{s}))$



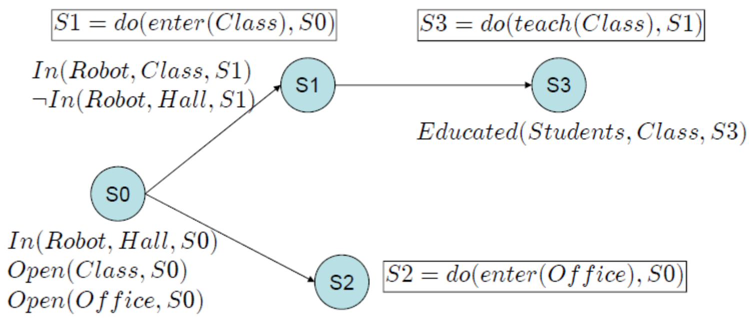

Goal: $\exists s Educated(Students, s)$

**Warning: notations are abused…**



## Situation Calculus: A Plan

$In(Robot, rm, s) \Rightarrow Educated(Students, rm, \mathbf{do}(\mathbf{teach}(\mathbf{rm}), \mathbf{s}))$




$$S3 = \mathbf{do}(teach, Class, \mathbf{do}(enter, Class, S0))$$
$$Educated(Students, S3)$$

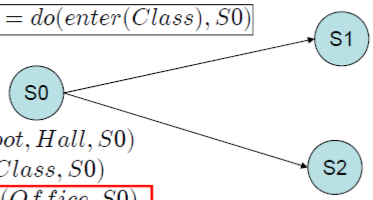
**Warning: notations are abused…**

# Situation Calculus: Problems?

- Effect Axioms (Positive + Negative)

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

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




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

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

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

**Warning: notations are abused…**



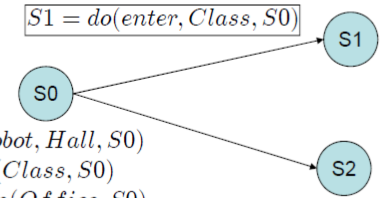
# Situation Calculus: Frame Problem*

$Room(rm)$

- Effect Axioms (Positive + Negative)

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

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




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

$In(Robot, Hall, S0)$
$Open(Class, S0)$
$\neg Open(Office, S0)$
$Color(Sky, Blue)$

?

Specifying only which conditions are changed by the actions do not allow, in logic, to conclude that all other conditions are not changed.

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

***First recognized by McCarthy & Hayes (1969)**

## Frame Axioms

Effect Axioms (Positive + Negative)

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

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

For each unchanged fluent we add:

$$Open(Office, s) \Rightarrow Open(Office, \mathbf{do}(\mathbf{enter}(\mathbf{rm}), \mathbf{s}))$$
$$\neg Open(Office, s) \Rightarrow \neg Open(Office, \mathbf{do}(\mathbf{enter}(\mathbf{rm}), \mathbf{s}))$$

$$Color(Sky, Blue, s) \Rightarrow Color(Sky, Blue, \mathbf{do}(\mathbf{enter}(\mathbf{rm}), \mathbf{s}))$$
$$\neg Color(Sky, Blue, s) \Rightarrow \neg Color(Sky, Blue, \mathbf{do}(\mathbf{enter}(\mathbf{rm}), \mathbf{s}))$$

How many in total? (for n distinct fluents and m distinct actions)

$2nm$ (Not exponential – but often not practical)

Explicitly specify that all conditions not affected by actions are not changed while executing that action



## Frame Problem

- (Representational) Frame Problem
  Significant because the real world has very many fluents
  Size of axioms: O(mn)
- Inferential Frame Problem
  Problem of projecting forward the results of a t step of actions in time O(nt)
- Origin of the Name: "Frame Problem"
  (i) Frame of Reference in Physics, which is assumed stationary wrt which motion is measured.
  (ii) Frames of a movie, in which normally most of the background stays constant while changes occur in the foreground

# Ramification Problem

**What are the ramifications of an action?**

- The robot entered the room.
- Its sensors are now in the room.
- The object it was carrying is now in the room.
- The robot is visible to you.

→ the indirect consequences of an action

**Do we really want to say all that in an effect axiom?**

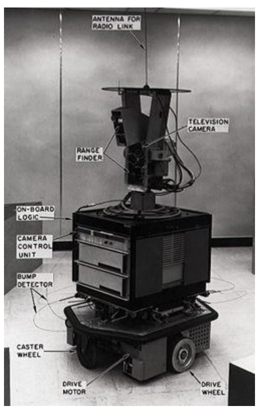
**How does this relate to the Frame Problem?**

→ Ramification Problem = The frame problem in the context of actions with indirect effects



# A solution: STRIPS

**Shakey @ Stanford**




**https://www.youtube.com/watch?v=qXdn6ynwpiI**