Robot Planning and Manipulation

Dr. Neil T. Dantam

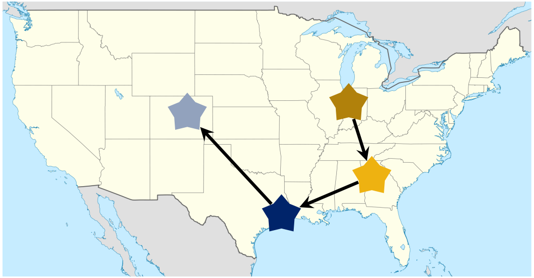
CSCI-498/598 RPM, Colorado School of Mines

Spring 2018

Outline

Dr. Neil T. Dantam

- Purdue University
  - B.S. Computer Science
  - B.S. Mechanical Engineering
- Georgia Tech, Ph.D. Robotics
- Rice, Postdoc Computer Science
- Mines, Assistant Professor, Computer Science
- Research: Robotics
  - Grammars for Robot Control
  - Robot Planning
  - Robot Manipulation



This Course

Robot Planning and Manipulation

Your Expectations

- Why are **you** here?
- What do you hope to gain from this course?

Prerequisites

1. Basic programming experience
2. Understanding of data structures and algorithms
3. Able to learn new programming languages and software frameworks (read the manual)
4. Familiar with differential calculus

Syllabus

Course Corrections

- What helps you? / What doesn't work?
- Ask about notation!
- Projects...

Outline

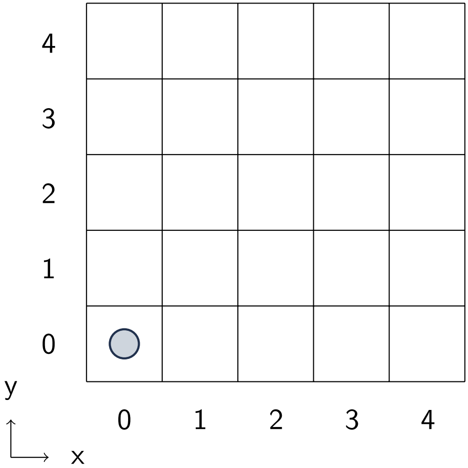
State Space

Definition: State Space

The set of values which the system can take.

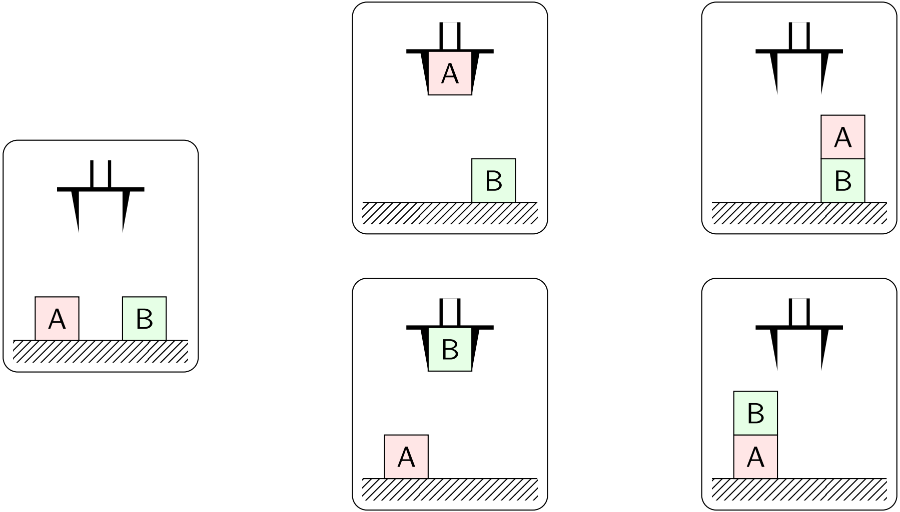
A state space may be discrete or continuous and finite or infinite.

Example: Grid State Space



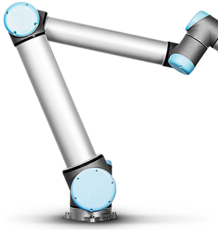
$$
\begin{aligned}
Q = \{ & (0,0),\ (1,0),\ (2,0),\ (3,0),\ (4,0) \\
& (0,1),\ (1,1),\ (2,1),\ (3,1),\ (4,1) \\
& (0,2),\ (1,2),\ (2,2),\ (3,2),\ (4,2) \\
& (0,3),\ (1,3),\ (2,3),\ (3,3),\ (4,3) \\
& (0,4),\ (1,4),\ (2,4),\ (3,4),\ (4,4)\ \}
\end{aligned}
$$

Example: Blocksworld State Space

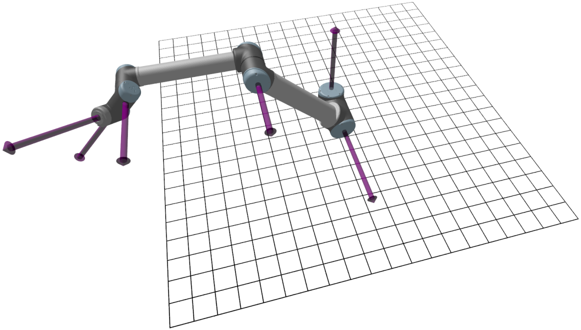


Example: Robot Arm Configuration Space

Universal Robots UR10



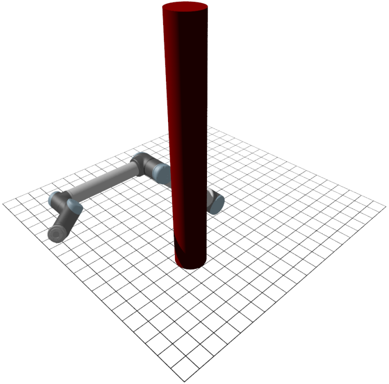
UR10 Axes



$Q \subseteq \mathbb{R}^6$

Example: Obstacles and Free Configuration Space

$$Q_{\text{free}} \subset \mathbb{R}^6$$



Transition Function

Continuous: $\frac{dx}{dt} = f(\underbrace{x}_{state}, \underbrace{u}_{input})$

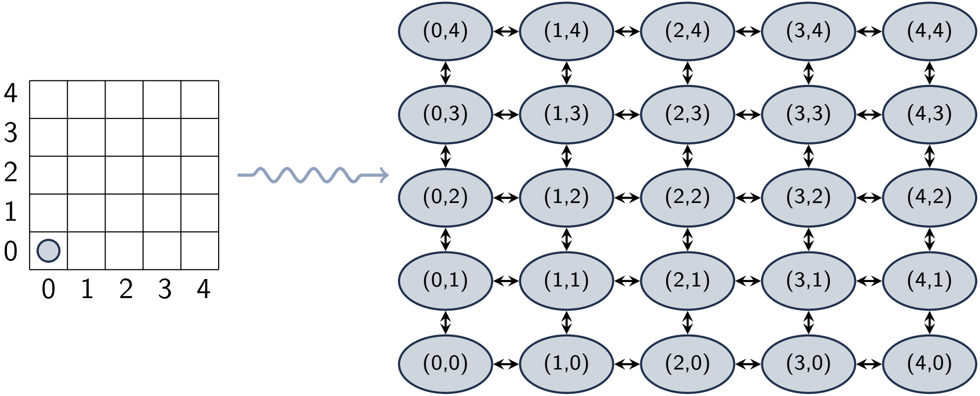
Discrete: $\delta : \overbrace{Q}^{\text{predecessor state}} \times \overbrace{U}^{\text{action}} \mapsto \overbrace{Q}^{\text{successor state}}$

$$q^{[k+1]} = \delta\left(q^{[k]},\ u^{[k]}\right)$$

Example: Grid Transition Function

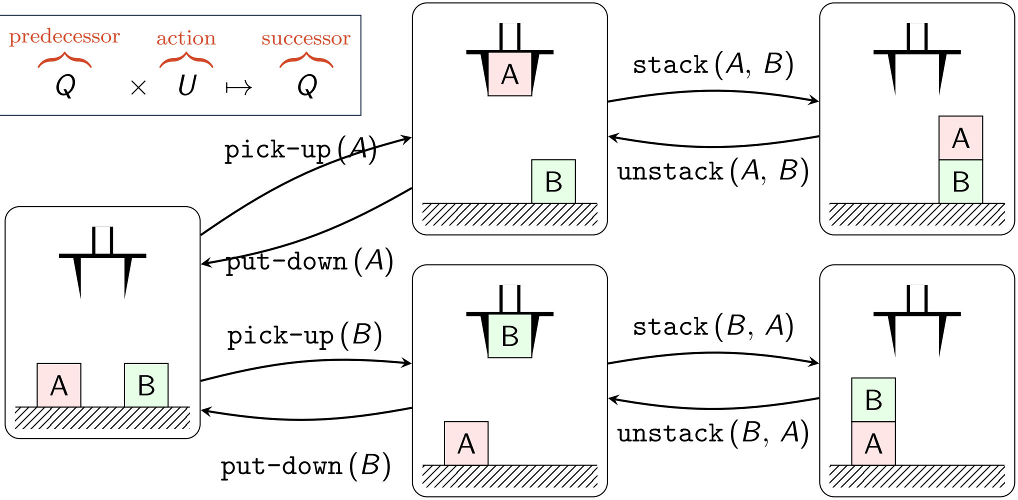
Graph Representation

$$\delta : \overbrace{Q}^{\text{predecessor}} \times \overbrace{U}^{\text{action}} \mapsto \overbrace{Q}^{\text{successor}}$$

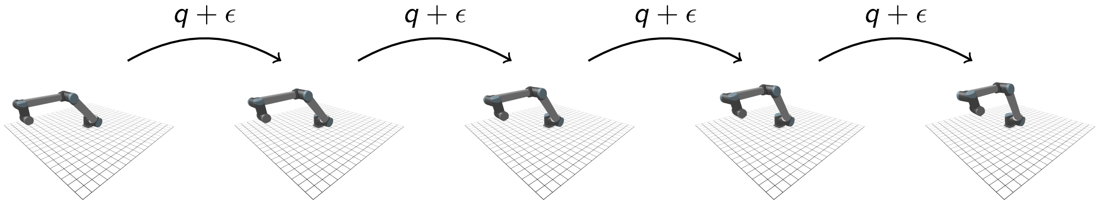


Example: Blocksworld Transition Function

$$\delta : \overbrace{Q}^{\text{predecessor}} \times \overbrace{U}^{\text{action}} \mapsto \overbrace{Q}^{\text{successor}}$$



Example: Robot Arm Transition Function



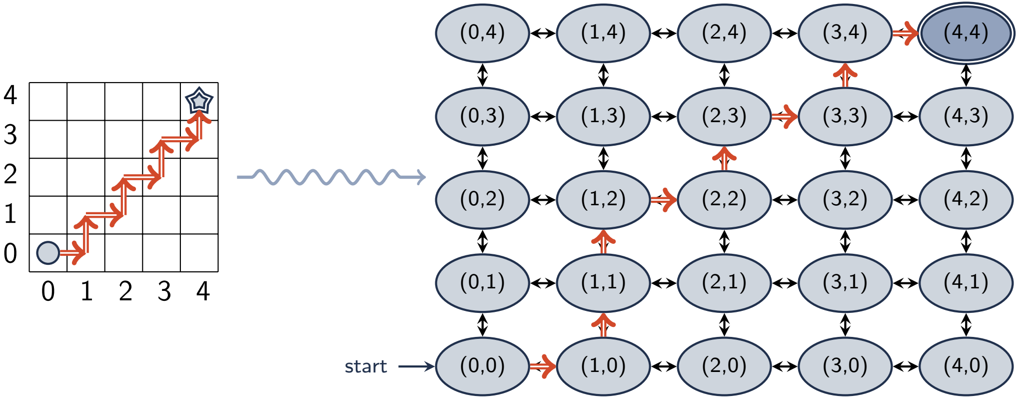
Assumption: Can connect "neighboring" configurations

The Planning Problem

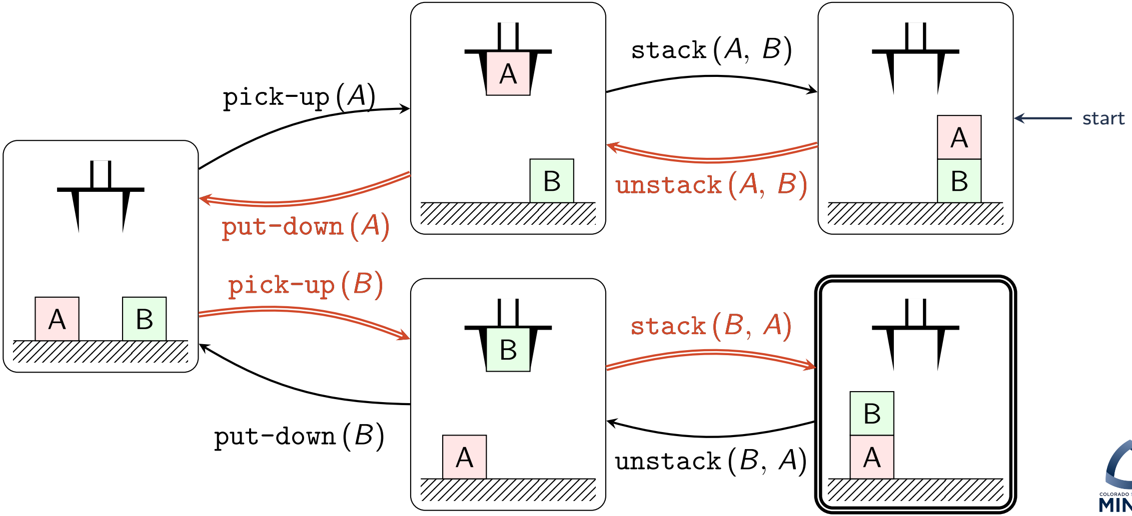
Given:
1. System model (state space, transition function, etc.)
2. Start state
3. Goal state / set

Find: Path (of states or actions) from start to goal

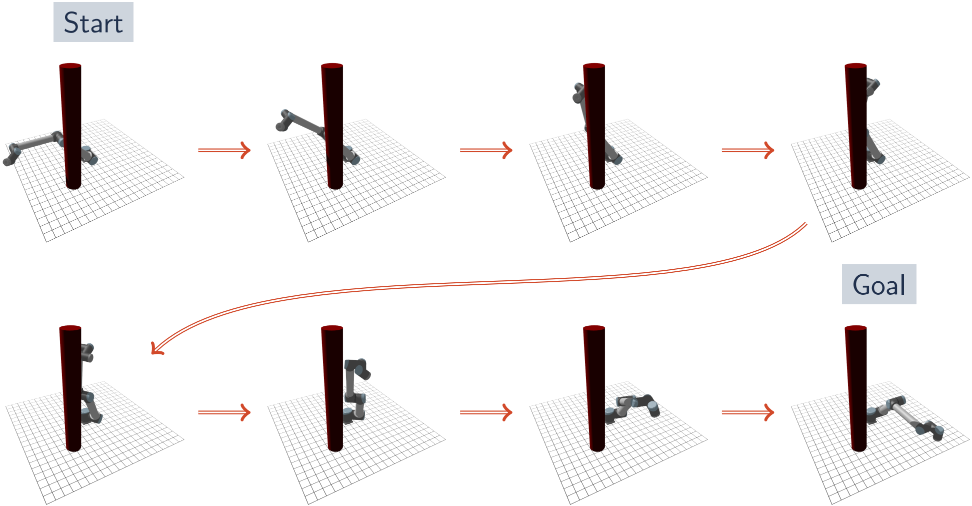
Example: Grid Plan



Example: Blocksworld Plan



Example: Motion Plan



Is planning just "fancy" search?

Given:
1. System model (state space, transition function, etc.)
2. Start state
3. Goal state / set

Find: Path (of states or actions) from start to goal

Solution:
- Search the state space,
- Beginning from the start state,
- Ending at goal state/set

Many kinds of state spaces.
Many kinds of "search."
Scalability!

Outline

Boolean Variables

(propositions)

$$
\begin{aligned}
\text{Values:} \quad & \mathbb{B} \equiv \{0, 1\} \\
& \qquad\qquad \text{true: } 1, T, \top \\
& \qquad\qquad \text{false: } 0, F, \perp \\
\text{Variables:} \quad & p \in \mathbb{B} \\
& p_1, \ldots, p_n \in \mathbb{B}^n
\end{aligned}
$$

Boolean Operators
Basic

Not

- $\neg 0 = 1$
- $\neg 1 = 0$

And

- $0 \wedge 0 = 0$
- $0 \wedge 1 = 0$
- $1 \wedge 0 = 0$
- $1 \wedge 1 = 1$

Or

- $0 \vee 0 = 0$
- $0 \vee 1 = 1$
- $1 \vee 0 = 1$
- $1 \vee 1 = 1$

Boolean Operators

Extended

Xor

$$(a \oplus b) \triangleq (a \lor b) \land \neg(a \land b)$$
$$\triangleq (a \land \neg b) \lor (\neg a \land b)$$

- $0 \oplus 0 = 0$
- $0 \oplus 1 = 1$
- $1 \oplus 0 = 1$
- $1 \oplus 1 = 0$

Implies

$$(a \implies b) \triangleq (\neg a \lor b)$$

- $(0 \implies 0) = 1$
- $(0 \implies 1) = 1$
- $(1 \implies 0) = 0$
- $(1 \implies 1) = 1$

Biconditional (iff)

$$(a \iff b) \triangleq (a \implies b) \land (b \implies a)$$
$$\triangleq \neg(a \oplus b)$$
$$\triangleq (a \land b) \lor (\neg a \land \neg b)$$

- $(0 \iff 0) = 1$
- $(0 \iff 1) = 0$
- $(1 \iff 0) = 0$
- $(1 \iff 1) = 1$

Truth Table

a	b	$(\neg a)$	$(a \wedge b)$	$(a \vee b)$	$(a \oplus b)$	$(a \implies b)$	$(a \iff b)$
0	0	1	0	0	0	1	1
0	1	1	0	1	1	1	0
1	0	0	0	1	1	0	0
1	1	0	1	1	0	1	1

Outline

Sets

- **set:** an unordered collection of object's without repetition
- $S = \{s_0, s_1, s_2, \ldots, s_n\}$
- set membership:
  - "x in S" / "x not in S"
  - $x \in S$ / $x \notin S$
- set builder notation:
  - $S = \{ \underbrace{x}_{\text{elements}} \mid \underbrace{P(x)}_{\text{property}} \}$
  - $S = \{ \underbrace{x \in A}_{\text{elements}} \mid \underbrace{P(x)}_{\text{property}} \}$

Common Sets

Booleans: $\mathbb{B} = \{0, 1\}$

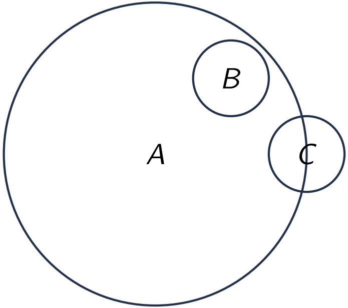
Integers: $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$

Natural Numbers: $\mathbb{N} = \{0, 1, 2, \ldots\}$

Real Numbers: $\mathbb{R}$

Real Vector: $\mathbb{R}^n$

Set Relations



Subset

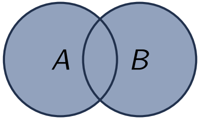
$$B \subset A$$
$$C \not\subset A$$
$$A \not\subset A$$
$$A \subseteq A$$

Superset

$$A \supset B$$
$$A \not\supset C$$
$$A \not\supset A$$
$$A \supseteq A$$

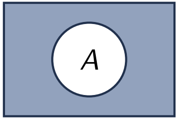
Set Operations

Union



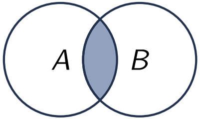
$$A \cup B = \{x \mid (x \in A) \vee (x \in B)\}$$

Complement



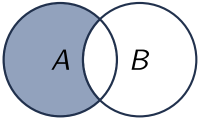
$$\overline{A} = \{x \mid x \notin A\}$$

Intersection



$$A \cap B = \{x \mid (x \in A) \wedge (x \in B)\}$$

Set Difference



$$A \setminus B = \{x \in A \mid x \notin B\}$$

Cartesian Product

- The Cartesian product, $A \times B$, is the set of all pairs of elements from A and B
- $A \times B = \{(x, y) \mid (x \in A) \wedge (y \in B)\}$
- $A = \{a_0, a_1, \ldots, a_m\}$
- $B = \{b_0, b_1, \ldots, b_n\}$
- $A \times B = \{(a_0, b_0), \ldots, (a_0, b_n), \ldots, (a_m, b_0), \ldots, (a_m, b_n)\}$

Function Notation

Function: object creating an input-output relationship (**mapping**)

Domain: the function's input

Range: the function's ouput

Notation: $\text{function} : \text{domain} \mapsto \text{range}$

Examples:

- Let $\mathbb{B}$ be the set of booleans: $\{0, 1\}$
  - $\neg : \mathbb{B} \mapsto \mathbb{B}$
  - $\wedge : \mathbb{B} \times \mathbb{B} \mapsto \mathbb{B}$
- Let $\mathbb{R}$ be the set of real numbers
  - $+ : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$
  - $\exp : \mathbb{R} \mapsto \mathbb{R}$

Sequences

Sequence: An ordered list of objects

- Example: $(1, 2, 3, 5, 8, \ldots)$

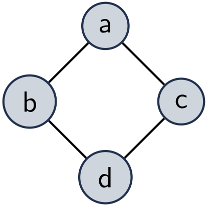
Tuple: A sequence of finite length

- **k-tuple:** An tuple of length k
- **pair:** An 2-tuple

Graph

- $G = (V, E)$
  - V: finite set of vertices
  - E: finite set of edges
    - Each edge being a **set** of two vertices
    - $E \subseteq \{\{x, y\} \mid (x \in) V \wedge (y \in V)\}$

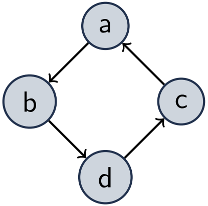
- Example:
  - $V = \{a\ ,\ b,\ c,\ d\}$
  - $E = \{\ \{a,\ b\},$
    $\{b,\ d\},$
    $\{d,\ c\},$
    $\{c,\ a\}\ \}$



Directed Graphs

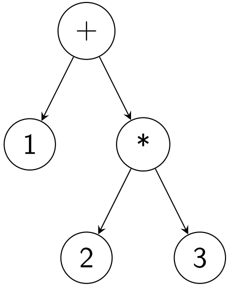
- $G = (V, E)$
  - V: finite set of vertices
  - E: finite set of edges
    - Each edge being a **pair** (sequence) of two vertices
    - $E \subseteq \{(x, y) \mid (x \in) V \wedge (y \in V)\}$

- Example:
  - $V = \{a\ ,\ b,\ c,\ d\}$
  - $E = \{\ (a,\ b),$
    $(b,\ d),$
    $(d,\ c),$
    $(c,\ a)\ \}$



Tree

- A graph that:
  - Is connected
    (a path exists between every pair of nodes)
  - No cycles
- Example:
  - $V = \{+\ ,\ 1,\ *,\ 2,\ 3\}$
  - $E = \{\ (+,\ 1),$
    $(+,\ *),$
    $(*,\ 2),$
    $(2,\ 3)\ \}$

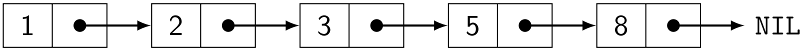


Symbolic Expressions

"S-expressions"

Sequence: $(1, 2, 3, 5, 8)$

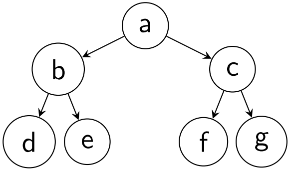
Data Structure:



S-expression: `(1 2 3 5 8)`

Trees as S-expressions

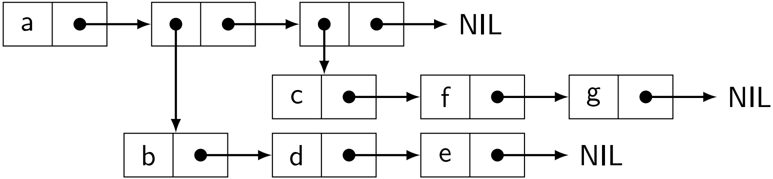
Tree:



S-expression:

(a (b d e) (c f g))

Data Structure:

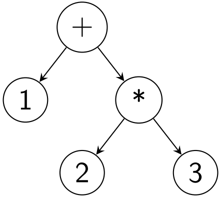


Example: Arithmetic as S-expressions

Infix:

`1 + 2 * 3`

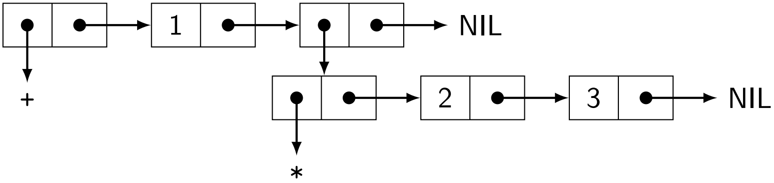
Tree:



S-Expression:

`(+ 1 (* 2 3))`

Data Structure:

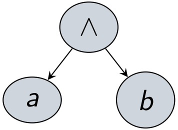


Example: Boolean Formulae

S-expression

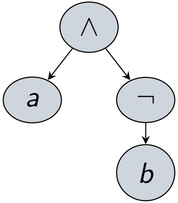
$a \wedge b$

`(and a b)`



$a \wedge \neg b$

`(and a (not b))`



$(a \vee b) \wedge (\neg a \vee b)$

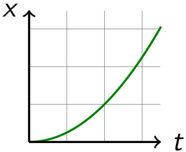
$(\neg a) \implies (b \vee c)$

Outline

Differential Calculus

Position

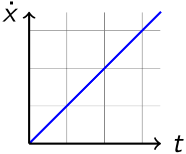
$x(t)$



$x = \frac{1}{2}t^2$

Velocity

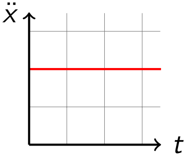
$v(t) = \frac{dx}{dt} = \dot{x}(t)$



$\dot{x} = t$

Acceleration

$a(t) = \frac{d^2x}{dt^2} = \ddot{x}(t)$



$\ddot{x} = 1$

Chain Rule

$$\frac{dy}{dt} = \frac{dy}{dx}\frac{dx}{dt}$$

$$y = f(x) \text{ and } x = g(t) \quad \rightsquigarrow \quad \underbrace{\frac{d}{dt} f(g(t))}_{dy/dt} = \underbrace{f'(g(t))}_{dy/dx} \underbrace{\frac{d}{dt} g(t)}_{dx/dt}$$

$\sin t^2$

- $f = \sin$ and $g(t) = t^2$
- $f' = \cos$ and $\dot{g}(t) = 2t$
- $\underbrace{\frac{d}{dt} \sin t^2}_{dy/dt} = \underbrace{\left(\cos t^2\right)}_{dy/dx} * \underbrace{(2t)}_{dx/dt}$

$\ln \sin t$

- $f = \ln$ and $g(t) = \sin t$
- $f'(x) = \frac{1}{x}$ and $\dot{g}(t) = \cos t$
- $\underbrace{\frac{d}{dt} \ln \sin t}_{dy/dt} = \underbrace{\left(\frac{1}{\sin t}\right)}_{dy/dx} * \underbrace{(\cos t)}_{dx/dt}$

Summary

Course Introduction

What is Planning?

Propositional Logic

Discrete Structures
- Sets
- Functions
- Structures
- Symbolic Expressions

Differential Calculus