Robot Planning and Manipulation

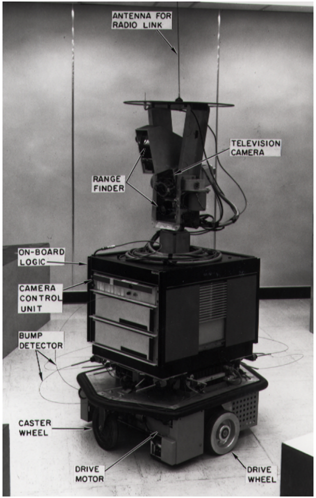
Dr. Neil T. Dantam

CSCI-534, Colorado School of Mines

Spring 2020

Shakey the Robot

1966-1972



"Intelligence"

- Sense, Plan, Act
- Stanford Research Institute Planning System (STRIPS)
  - Discrete Actions (e.g., push)
  - Preconditions
  - Effects
- Motion Planning

Outline

Thinking Machines

"The question of whether a computer can think
is no more interesting
than the question
of whether a submarine can swim."

–Edsger W. Dijkstra

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

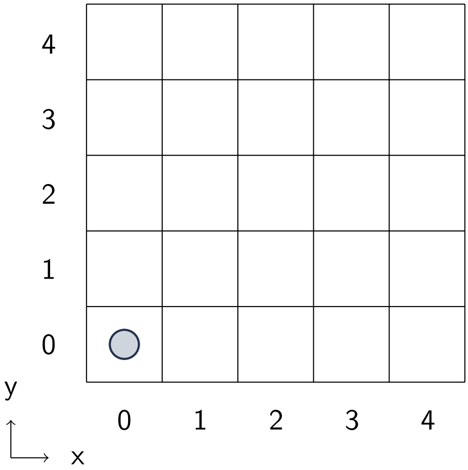
State Space

Definition (State Space)

The set of values which the system can take.

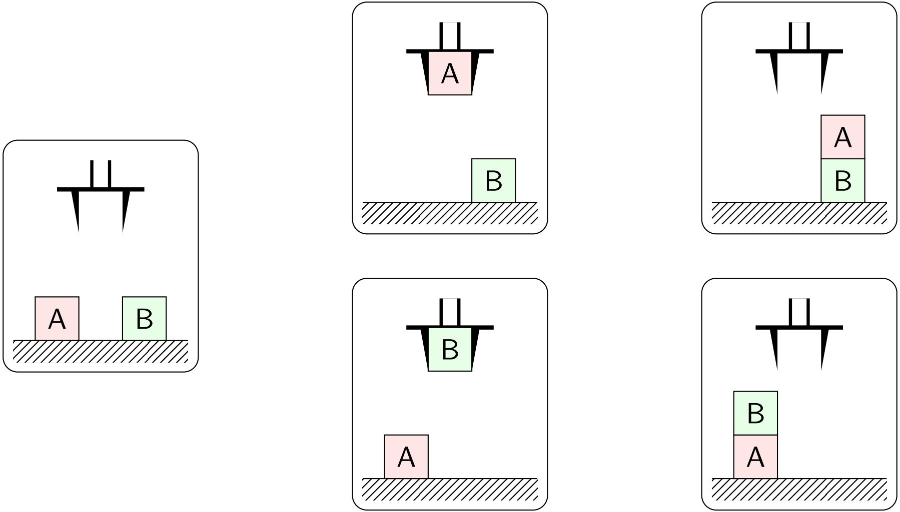
A state space may be discrete or continuous and finite or infinite.

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

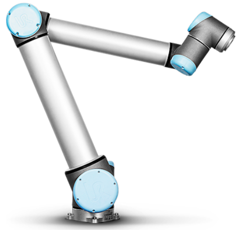


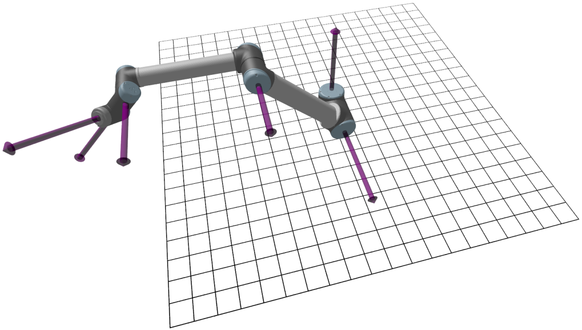
Example: Robot Arm Configuration Space

Universal Robots UR10

UR10 Axes

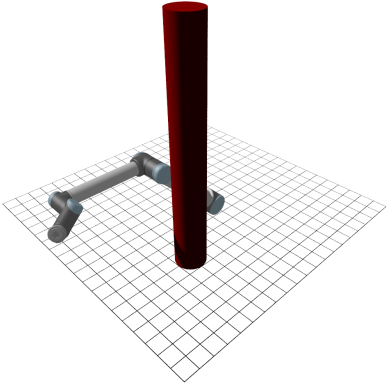
$Q \subseteq \mathbb{R}^6$





Example: Obstacles and Free Configuration Space

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



Transition Function

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

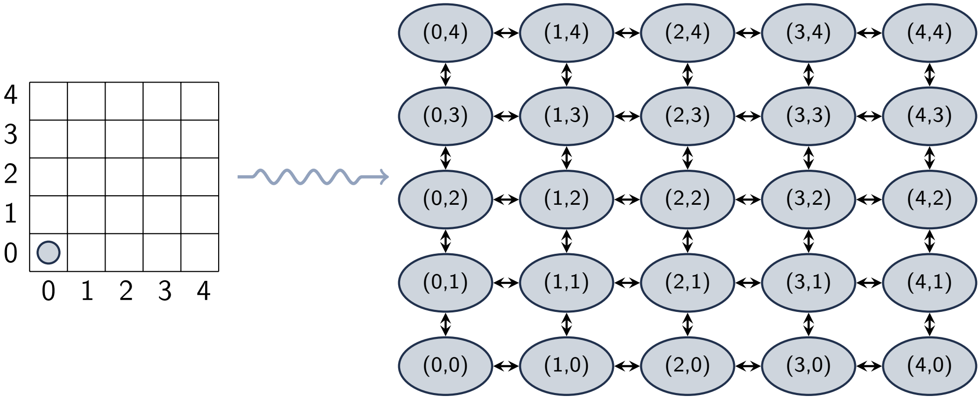
Discrete: $\delta : \overbrace{Q}^{\text{predecessor state}} \times \overbrace{U}^{\text{action}} \mapsto \overbrace{Q}^{\text{successor state}}$

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

Example: Grid Transition Function

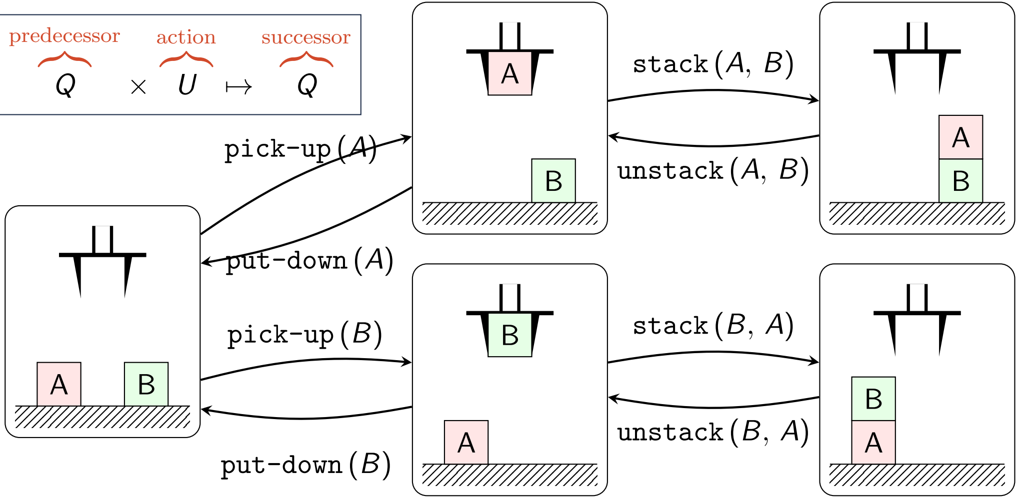
Graph Representation

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

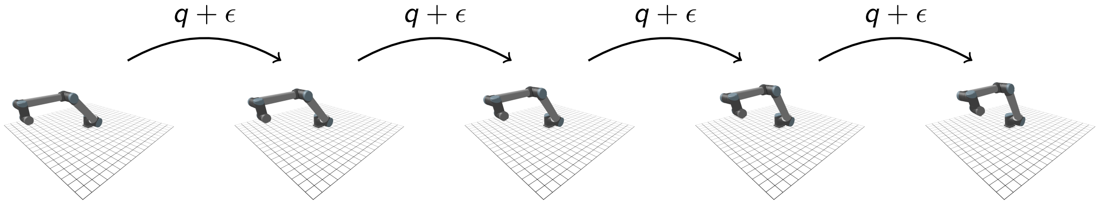


Example: Blocksworld Transition Function

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

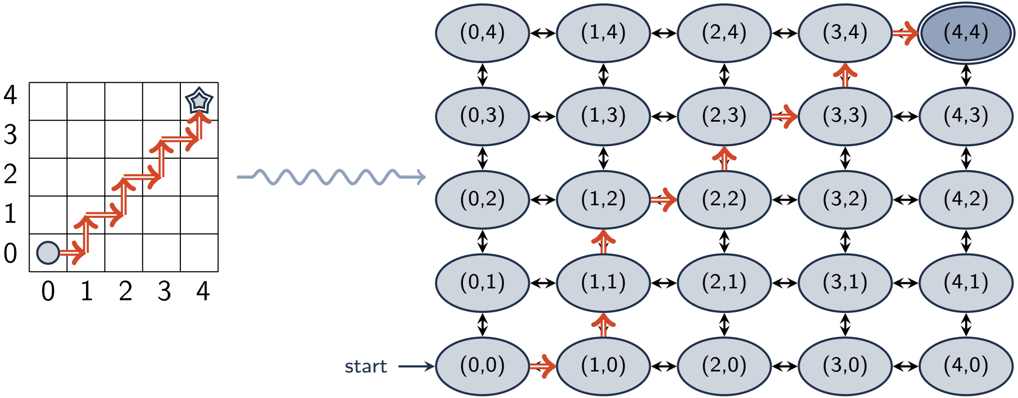


Example: Robot Arm Transition Function

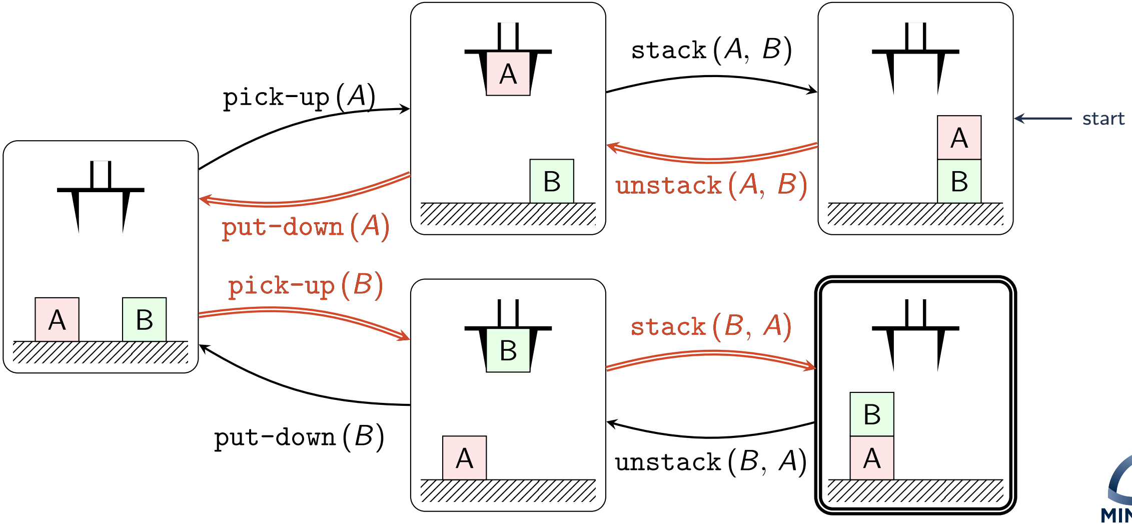


Assumption: Can connect "neighboring" configurations

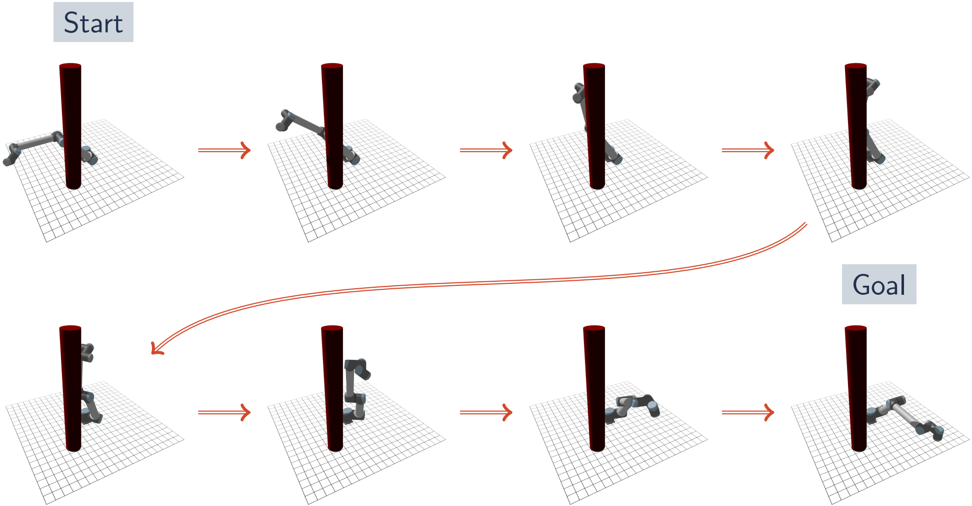
Example: Grid Plan



Example: Blocksworld Plan



Example: Motion Plan

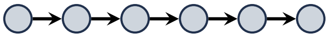


Simulation vs. Planning

Simulation

$$x^{[k+1]} = f(x^{[k]})$$

Given: Predecessor/Initial state $(x^{[k]})$
Find: Successor/Final state $(x^{[k+1]})$

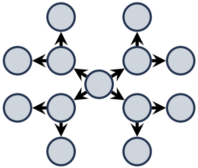


Planning

$$x^{[k+1]} = f(x^{[k]}, u^{[k]})$$

Given:
1. Predecessor/Initial state $(x^{[k]})$
2. Successor/Goal state $(x^{[k+1]})$

Find: Action / Path to reach goal $(u^{[k]})$



Different problems; simulation as "subroutine" for planning

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

Expectations

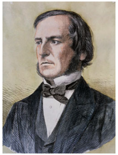
- This course is an upper/graduate level computer science course
  - You already know how to program
  - You can learn new programming languages and frameworks
- This course is *Robot Planning and Manipulation*
  - This is **NOT** a programming course (but we will program)
  - This course **IS** about the math and algorithms for autonomous robots
- Robotics is interdisciplinary
  - Discrete math and algorithms
  - Continuous equations and (differential) calculus

Syllabus

Outline

Boolean Variables

(propositions, bits)



true: $1, T, \top$

false: $0, F, \bot$

Boolean Operators

Basic

Not: $\neg$

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

And: $\wedge$

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

Or: $\vee$

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

Boolean Operators

Extended

Xor: $\oplus$

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

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

Implies: $\implies$

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

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

Biconditional (iff): $\iff$

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

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

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

Sets

Definition (Set)

An unordered collection of object's without repetition

Notation

- $S = \{s_0, s_1, s_2, \ldots, s_n\}$
- Empty Set: $\emptyset$
- set membership:

$$\underbrace{x \in S}_{x \text{ in } S} \qquad \underbrace{x \notin S}_{x \text{ not in } S}$$

Example

- $S = \{1, 2, 3\}$
- Common Sets:

Integers:	$\mathbb{Z}$
Real Numbers:	$\mathbb{R}$
Real Vector:	$\mathbb{R}^n$
Booleans:	$\mathbb{B}$

- $2 \in \mathbb{Z}$
- $\pi \notin \mathbb{Z}$

Set Builder Notation

Notation

- "The set x, such that $P(x)$":

$$S = \{ \underbrace{x}_{\text{elements}} \mid \underbrace{P(x)}_{\text{property}} \}$$

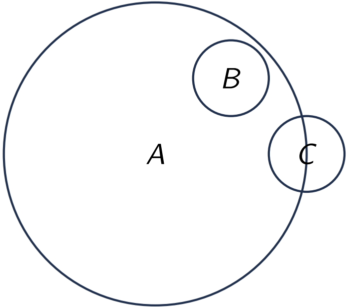
- "The set x, such that $P(x)$":

$$S = \{ \underbrace{x \in A}_{\text{elements}} \mid \underbrace{P(x)}_{\text{property}} \}$$

Example

- Even integers:
$s = \{ x \in \mathbb{Z} \mid \frac{x}{2} \in \mathbb{Z} \}$

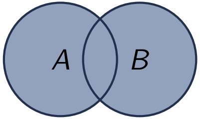
Set Relations



Subset	Superset
$B \subset A$	$A \supset B$
$C \not\subset A$	$A \not\supset C$
$A \not\subset A$	$A \not\supset A$
$A \subseteq A$	$A \supseteq A$

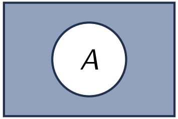
Set Operations

Union



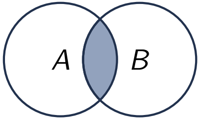
$$A \cup B = \{x \mid (x \in A) \vee (x \in B)\}$$

Complement



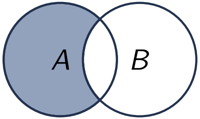
$$\overline{A} = \{x \mid x \notin A\}$$

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

Definition (Cartesian Product)

The Cartesian product, $A \times B$, is the set of all pairs of elements from A and B:

$$A \times B = \{(x, y) \mid (x \in A) \wedge (y \in B)\}$$

Example

- $A = \{a_0,\ a_1\}$
- $B = \{b_0,\ b_1,\ b_2\}$
- $A \times B = \{(a_0,\ b_0),\ (a_0,\ b_1),\ (a_0,\ b_2),\ (a_1,\ b_0),\ (a_1,\ b_1),\ (a_1,\ b_2)\}$

Function Notation

Definition

function object creating an input-output relationship (**mapping**)

domain the function's input

range the function's ouput

notation $\mathrm{function} : \mathrm{domain} \mapsto \mathrm{range}$

Illustration

$$\texttt{domain} \xrightarrow{\text{input}} \boxed{\texttt{function}} \xrightarrow{\text{output}} \texttt{range}$$

Example

- 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

Definition (Sequence)

An ordered list of objects.

Example

$(1, 2, 3, 5, 8, \ldots)$

Definition (Tuple)

A sequence of finite length.

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

Example

- 3-tuple: $(2, 4, 8)$
- pair-tuple: (a, b)

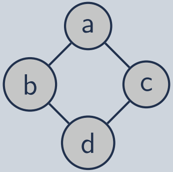
Graphs

Definition (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 (Diagram)



Example (Symbolic)

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

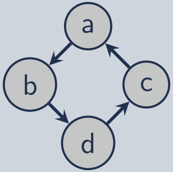
Directed Graphs

Definition (Directed Graph)

$G = (V, E)$

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

Example (Diagram)



Example (Symbolic)

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

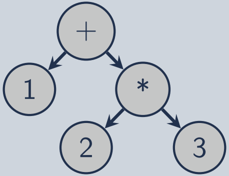
Trees

Definition (Tree)

A tree is a graph that:

- Is connected
  (a path exists between every pair of nodes)
- No cycles

Example (Diagram)



Example (Symbolic)

- $V = \{+\ ,\ 1,\ *,\ 2,\ 3\}$
- $E = \{\ (+,\ 1),\ (+,\ *),\ (*,\ 2),\ (*,\ 3)\ \}$

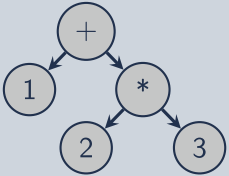
Trees and Expressions

Why and How?

Why use trees?

Abstract representation of expressions:

$1 + 2 * 3 \rightsquigarrow$



Representation

- How to represent a node in the tree?
  - Data element
  - Children
- How to represent node's children?

  Abstract: Sequence

  Concrete: Array, List

List vs. Tree

List

```
struct listnode {
    void *data;
    struct listnode *rest;
};
```

Tree

```
struct treenode {
    void *data;
    struct listnode *children;
};
```

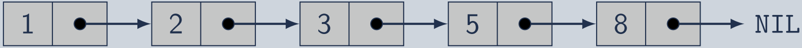
The same!
Can we use this?

Rethinking Lists as Symbolic Expressions (“S-expressions”)

Sequence

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

Data Structure – List



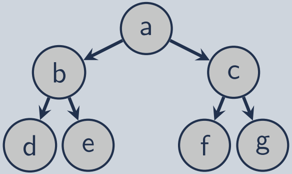
S-Expression notation

Sequence without commas:

`(1 2 3 5 8)`

Trees as S-expressions

Tree



S-Expression

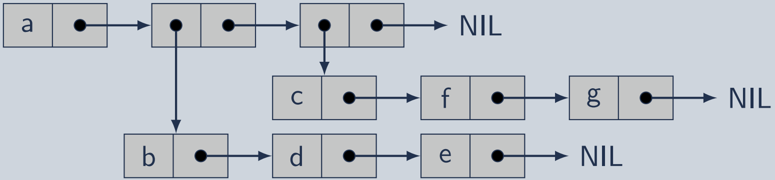
Root: First Element

Children: Rest of Elements

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

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

Data Structure

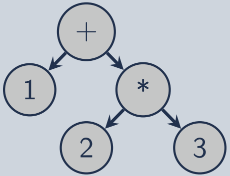


Example: Arithmetic as S-expressions

Infix

1 + 2 ∗ 3

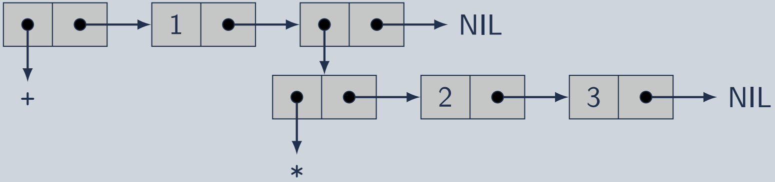
Tree



S-expression

```
(+ 1 (* 2 3))
```

Data Structure

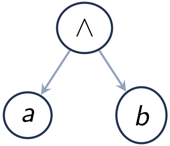


Example: Boolean Formulae

S-expression

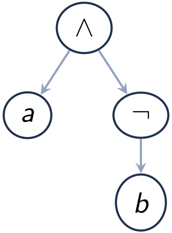
$a \wedge b$

`(and a b)`



$a \wedge \neg b$

`(and a (not b))`



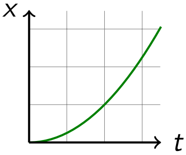
$(a \vee b) \wedge (\neg a \vee b)$

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

Differential Calculus

Position

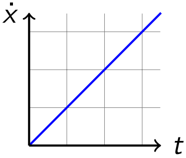
$x(t)$



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

Velocity

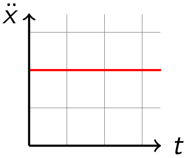
$v(t) = \frac{dx}{dt} = \dot{x}(t)$



$\dot{x} = t$

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