SAPIENZA
UNIVERSITÀ DI ROMA

Artificial Intelligence

2018/2019 Prof: Daniele Nardi

Exercises 1:

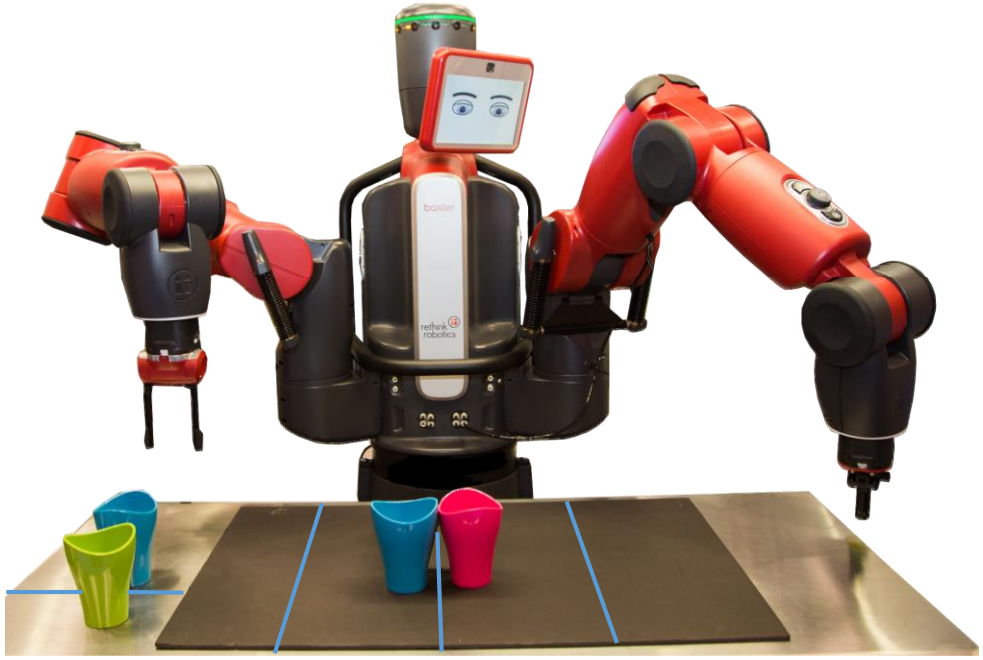
Search in the State Space*

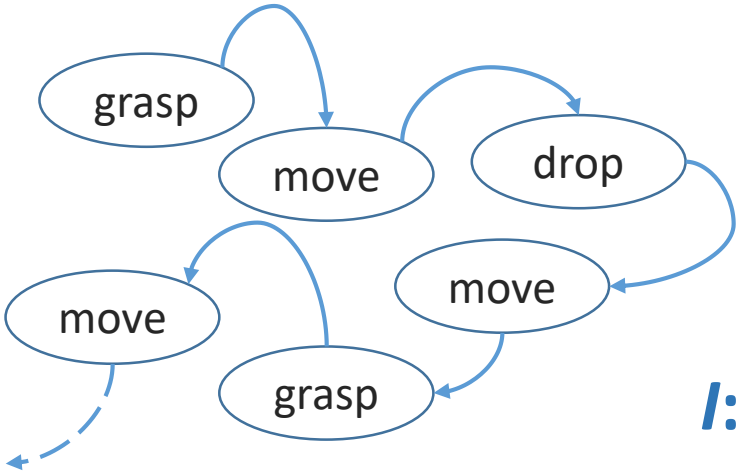
Francesco Riccio
email: riccio@diag.uniroma1.it

*The slides have been prepared using the textbook material available on the web, and the slides of the previous editions of the course by Prof. Luigia Carlucci Aiello, Prof. Daniele Nardi and Dott. Fabio Previtali.

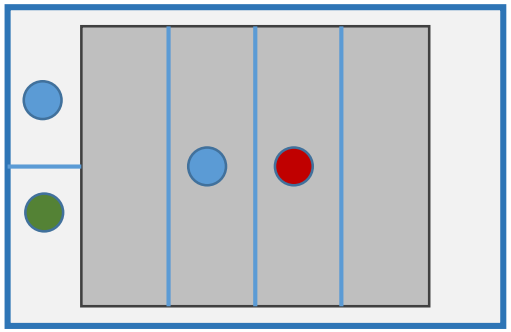
Formalization of Search Problems (Recap)

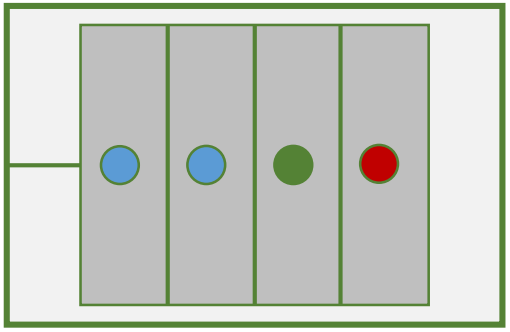
- Abstraction
- **State** representation
- **Initial** and **goal** state
- **Operator** specification:
  { grasp, drop, move }
- Constraint definition (conditions on operators, using a representation satisfying them)
- **Search** of solution



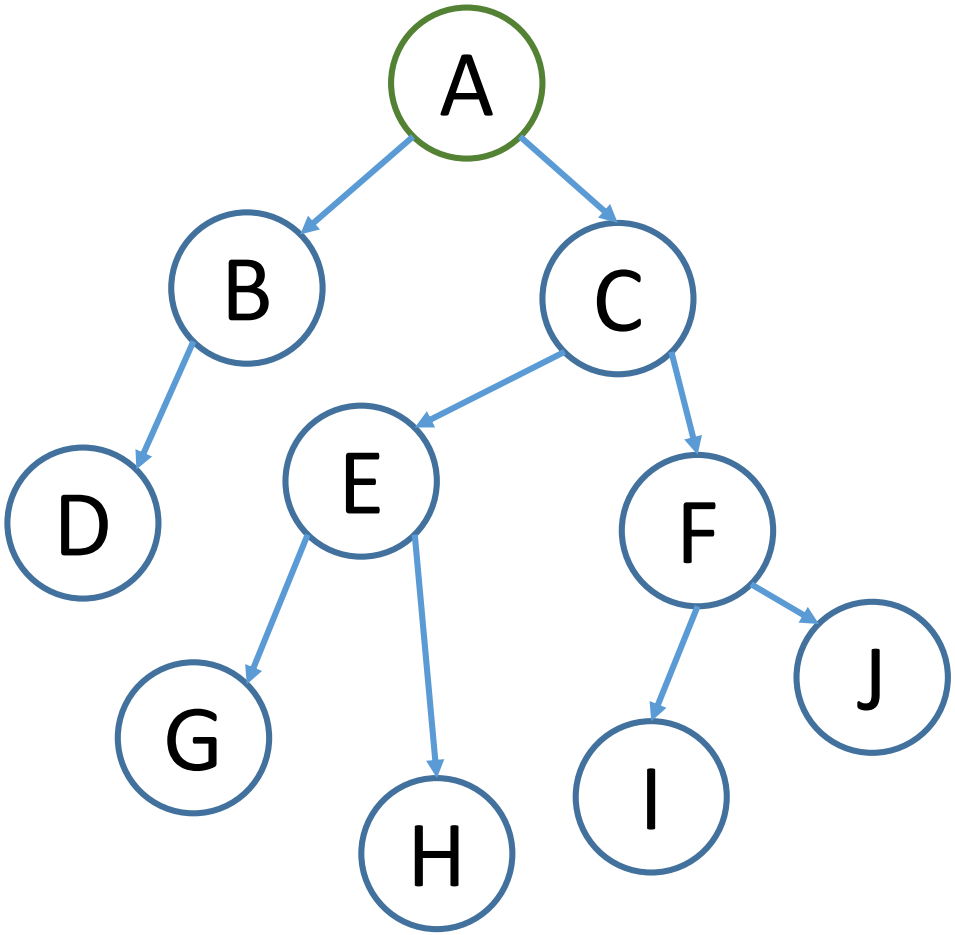


S: <c, c, c, c, c, c, c> , $c \in$ {r, g, b, e}





Breadth-first, Depth-first, Iterative deepening search



Iterative deepening search:

Level 0 : {
    expansion: {}
    visit: {A}
}
Level 1: {
    expansion: {A}
    visit: {A,B,C}
}
Level 2: {
    expansion: {A,B,C}
    visit: {A,B,D,C,E,F}
}
Level 3: {
    expansion: {A,B,C,E,F}
    visit: {A,B,D,C,E,G,H,F,I,J}
}

Depth-first search:

expansion: {A,B,C,E,F}
visit: {A,B,D,C,E,G,H,F,I,J}

Breadth-first search:

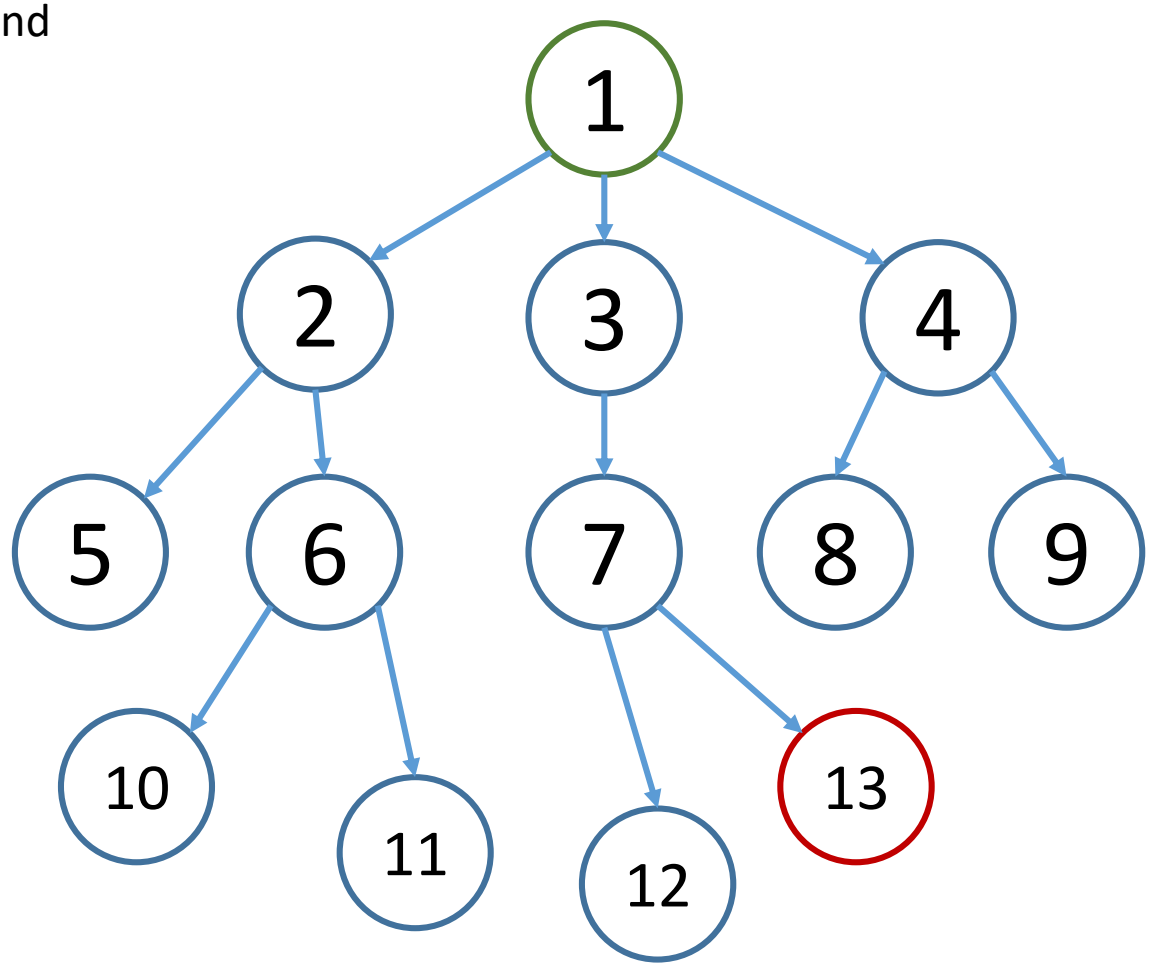
expansion: {A,B,C,E,F}
visit: {A,B,C,D,E,F,G,H,I,J}

Example: Iterative Deepening Search

IDS: write the node **expansion** and node **visit.** Goal node: **13**

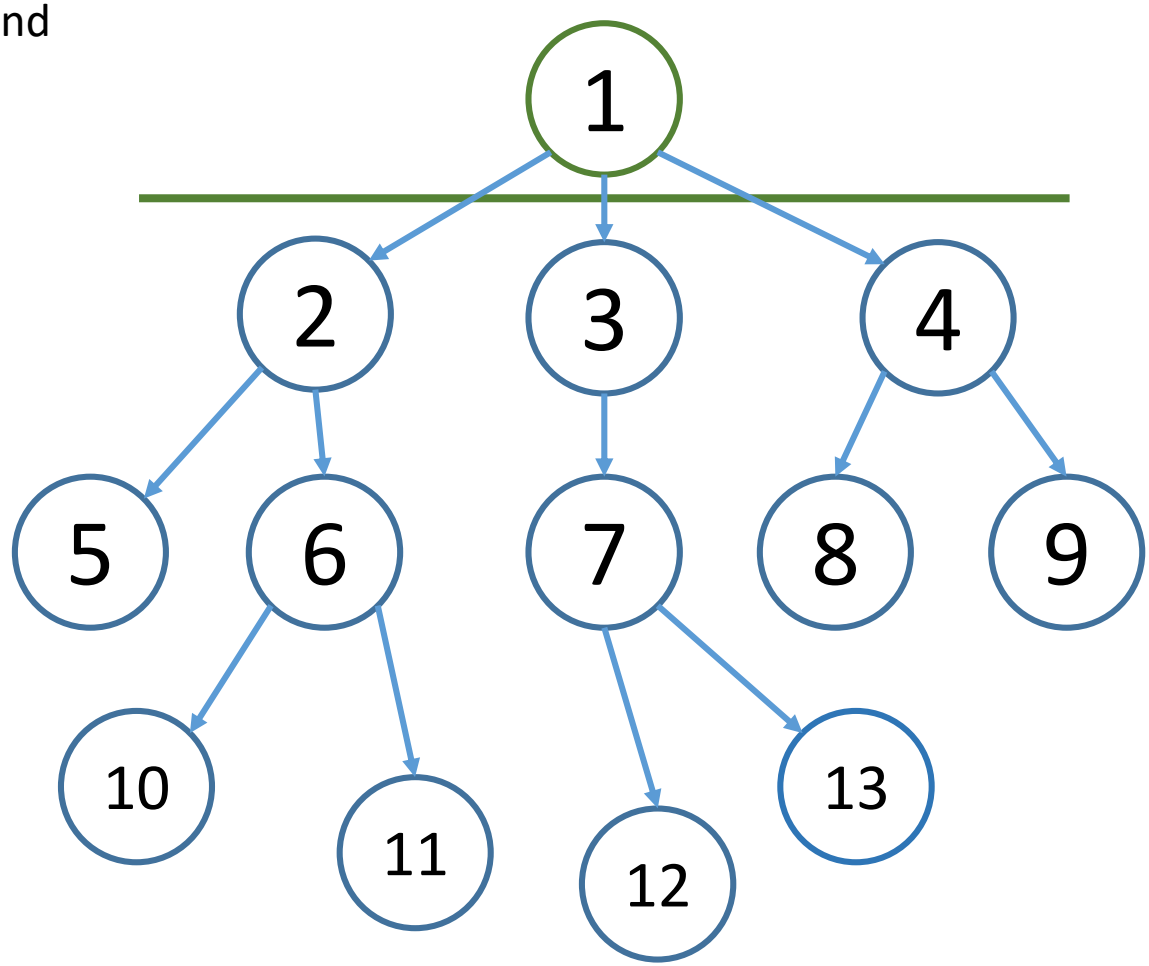
Expansion:

Visit:



Example: Iterative Deepening Search

IDS: write the node **expansion** and node **visit.** Goal node: **13**

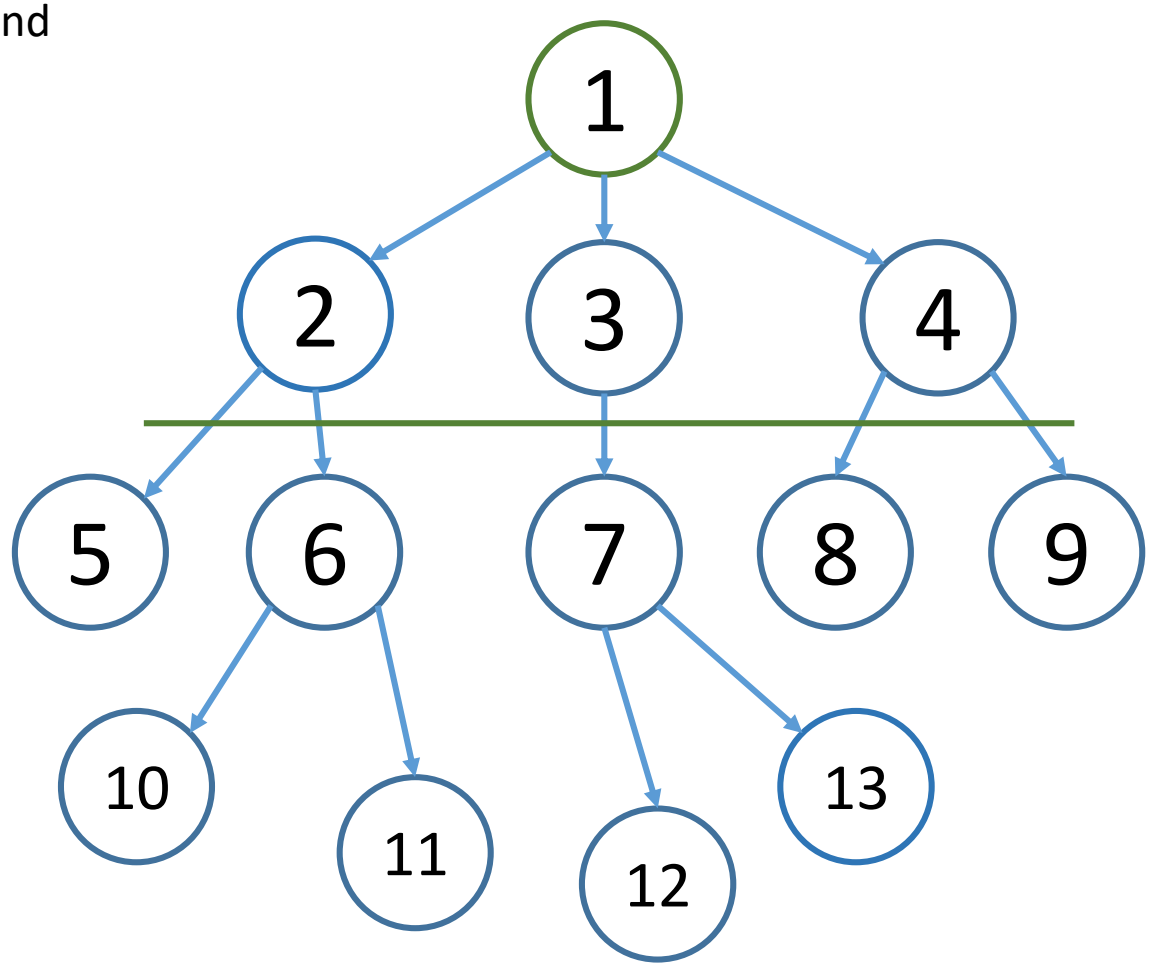


Expansion:
Level 0 : {}

Visit:
Level 0 : {1}

Example: Iterative Deepening Search

IDS: write the node **expansion** and node **visit.** Goal node: **13**

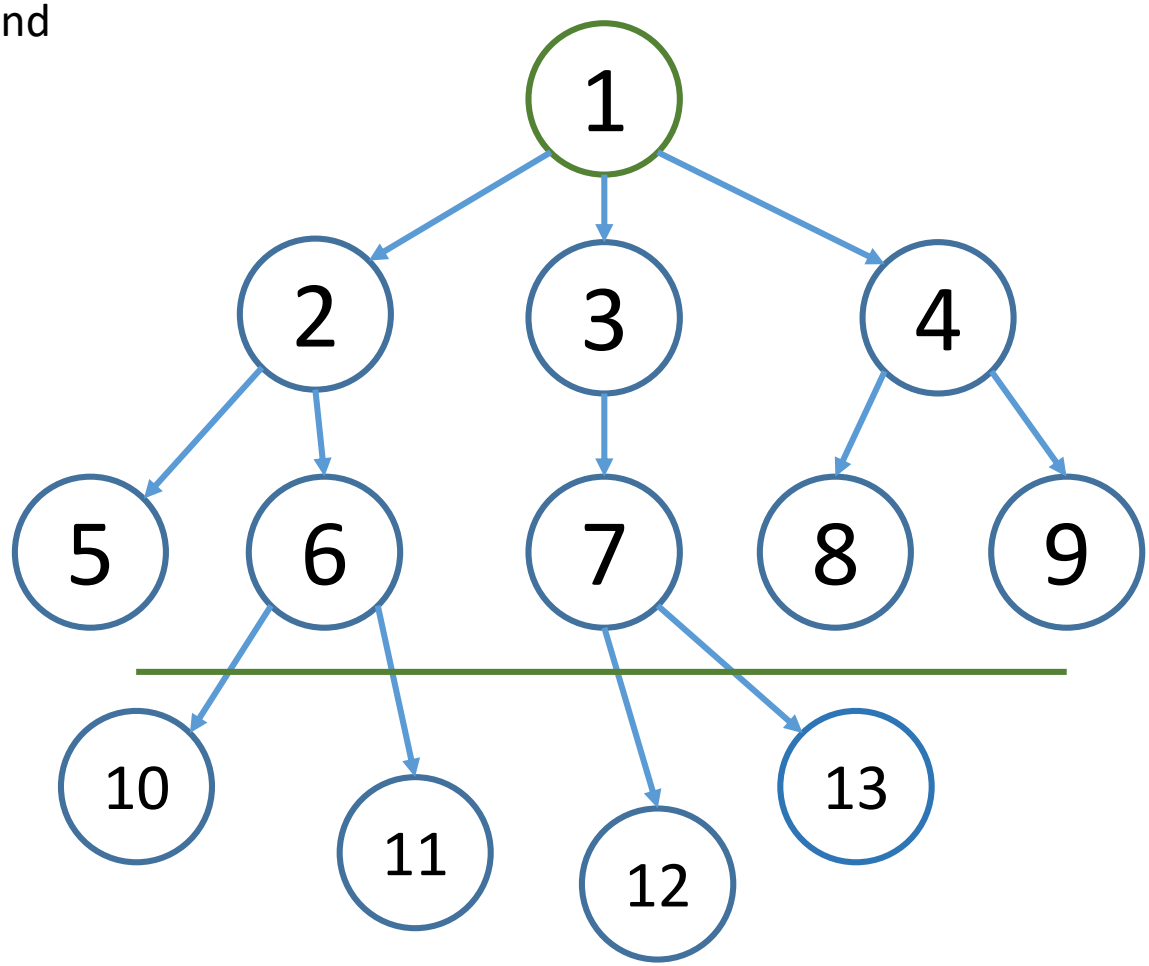


Expansion:
Level 0 : {}
Level 1 : {1}

Visit:
Level 0 : {1}
Level 1 : {1,2,3,4}

Example: Iterative Deepening Search

IDS: write the node **expansion** and node **visit.** Goal node: **13**



Expansion:
Level 0 : {}
Level 1 : {1}
Level 2 : {1,2,3,4}

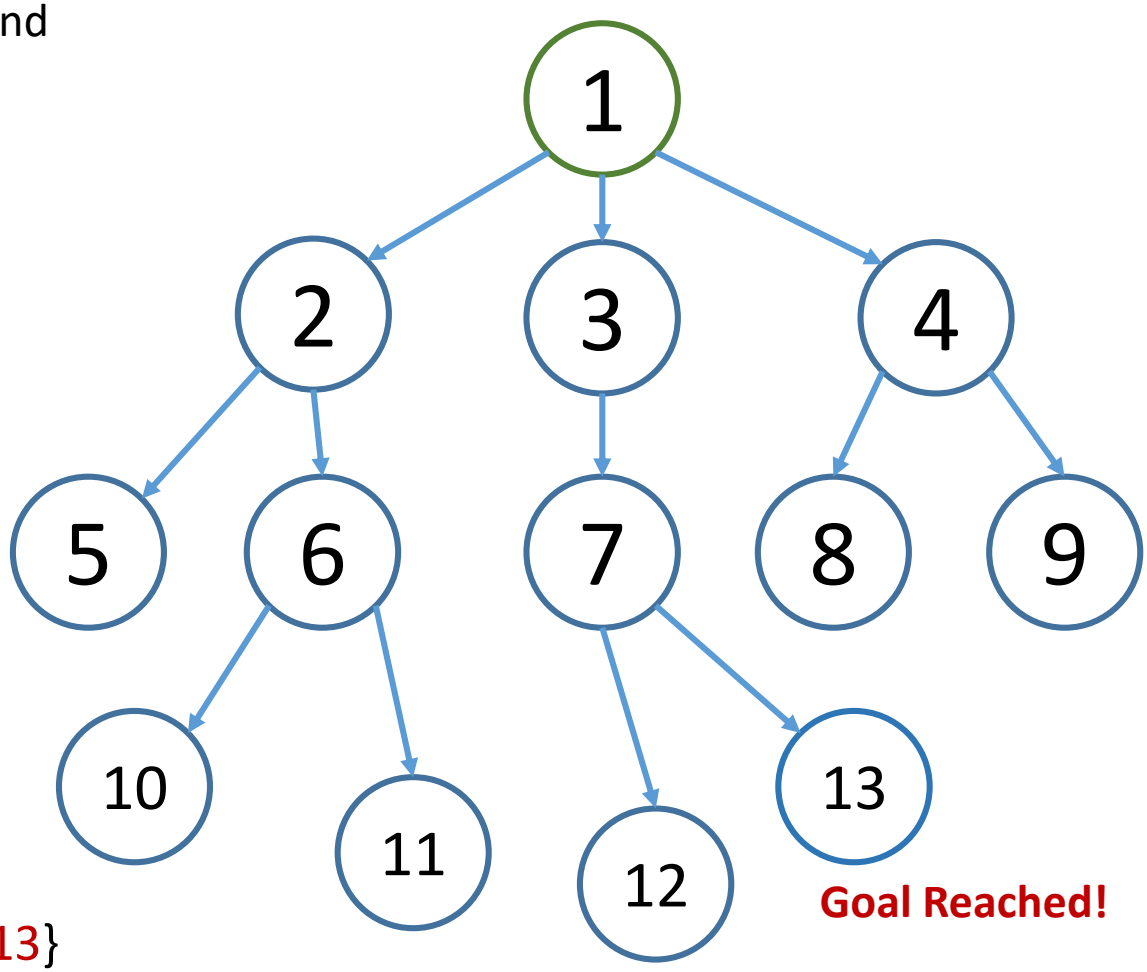
Visit:
Level 0 : {1}
Level 1 : {1,2,3,4}
Level 2 : {1,2,5,6,3,7,4,8,9}

Example: Iterative Deepening Search

IDS: write the node **expansion** and node **visit.** Goal node: **13**

Expansion:
Level 0 : {}
Level 1 : {1}
Level 2 : {1,2,3,4}
Level 3 : {1,2,6,3,7}

Visit:
Level 0 : {1}
Level 1 : {1,2,3,4}
Level 2 : {1,2,5,6,3,7,4,8,9}
Level 3 : {1,2,5,6,10,11,3,7,12,13}



Example: Fully balanced search tree

If the **branching factor is 3** and a **breadth-first search** is executed at **depth 3**. How many nodes are generated and visited?

Depth 0: ∅ expansions, 3^0 = **1** node visited, the root
Depth 1: **1** node is expanded, 3^1 = **3** nodes visited
Depth 2: **3** nodes expanded, 3^2 = **9** nodes visited
Depth 3: **9** nodes expanded, 3^3 = **27** nodes visited

Expanded nodes: 1 + 3 + 9 = **13**, **Visited** nodes: 1 + 3 + 9 + 27 = **40**

If **branching factor is 4** and an **iterative deepening search** is executed at **depth 3**. How many nodes are visited?

Depth 0: ∅ expansions, 4^0 = **1** node visited, the root
Depth 1: **1** node is expanded, $4^0 + 4^1$ = **5** nodes visited
Depth 2: **5** nodes expanded, $4^0 + 4^1 + 4^2$ = **21** nodes visited
Depth 3: **21** nodes expanded, $4^0 + 4^1 + 4^2 + 4^3$ = **85** nodes visited

Expanded nodes: 1 + 5 + 21 = **27**, **Visited** nodes: 1 + 5 + 21 + 85 = **112**

Example: Loop check

The state space of the problem is $\{s_1, s_2, s_3, s_4\}$.
Initial state s_1. **A**, **B** and **C** are the operators defined according to:

	s_1	s_2	s_3	s_4
s_1	-	A	A	A
s_2	A	-	B	B
s_3	A	B	-	C
s_4	A	B	C	-

Where the **operator** in **position** $\langle i, j \rangle$ represents a **transition** from state ***i*** to state ***j***. The set of goal states is empty. The costs of the operators are cost(A) = 10, cost(B) = 15, cost(C) = 5.

Example: Loop check

Using a search with **loop check**, i.e. avoid repeated states.
Assuming that the order of node expansion is s_1, s_2, s_3, s_4:

- Does the cost of the operators influence the order in which the nodes are visited?

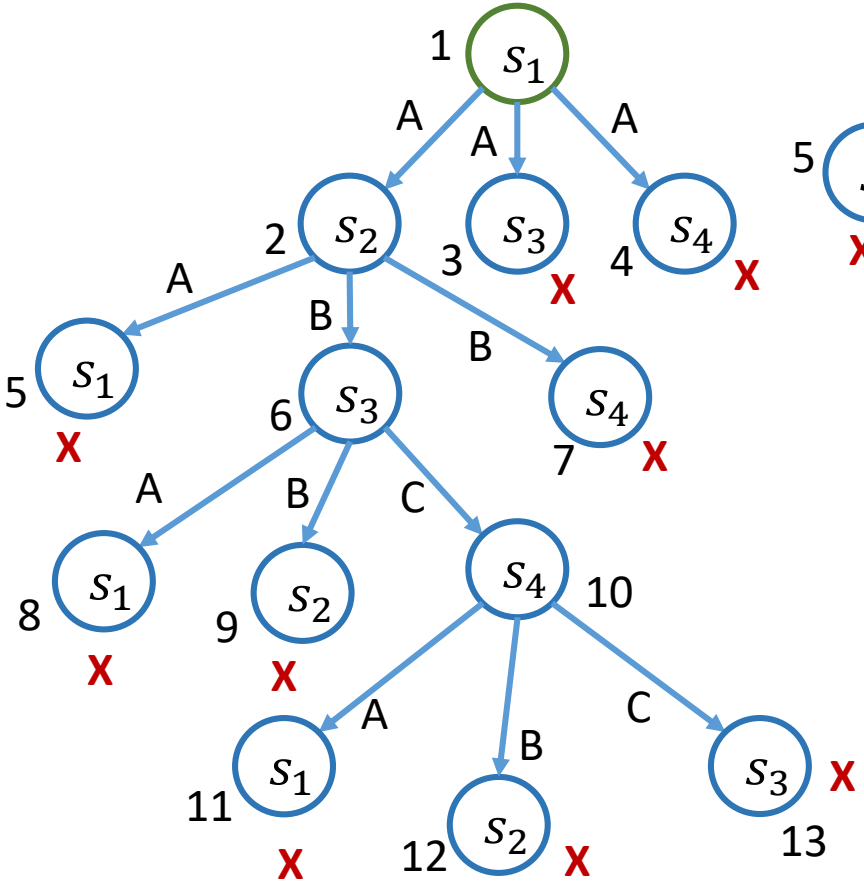
  Nodes are numbered according to the generation order. It is thus, independent from the operators' cost.

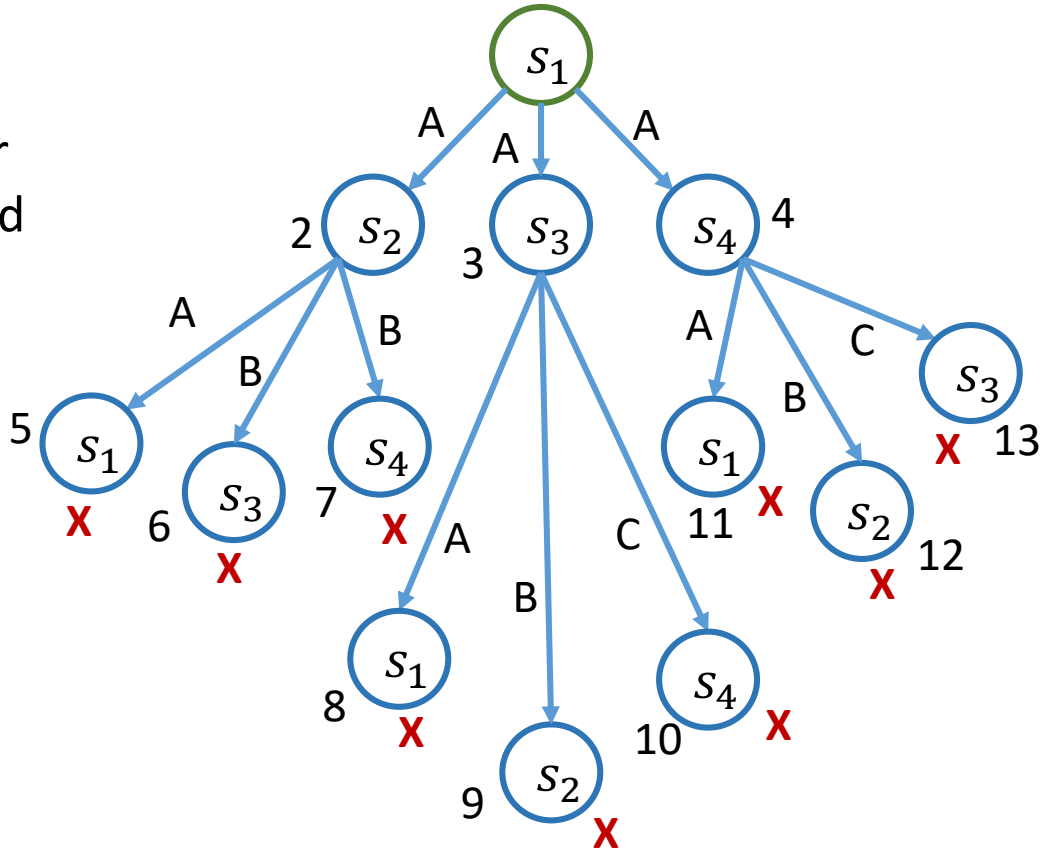
- Build the search tree using **depth-first**, specifying the order in which the nodes are generated

- Build the search tree using **breadth-first**, specifying the order in which the nodes are generated

Example: Loop check

Build the search tree using **depth-first**, specifying the order in which the nodes are generated





Build the search tree using **breadth-first**, specifying the order in which the nodes are generated

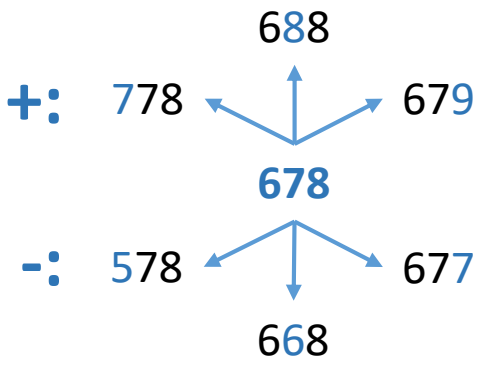
Example: Forbidden numbers

- The game is about integer numbers between 100 and 999
- Two numbers are given to express the start and goal situations
- A set of numbers is defined as the set of forbidden numbers (eg. {679, 666})
- A move changes a single digit in the number by adding or substracting one

Constraints:

- 1 cannot be added to 9 nor subtracted from 0
- Avoid forbidded numbers
- Consecutive changes of the same digit are not allowed:
  677 -> 678 -> 679

eg.



Example: Forbidden numbers

1. Characterize the **state space**
2. Specify the **operators**
3. Find a minimal sequence of moves to go from **I** = 567 to **G** = 777.
4. **Forbidden** = {666, 667}
5. Find a good **heuristics** to be used by **A***
6. **Draw** the **search tree** generated by A* to solve the problem.
7. For each node **indicate**: the number (state), f, g, and h and an integer indicating the expansion order

Example: Forbidden numbers

Characterize the **state space**

$S = D^+ \text{ x } D \text{ x } D \text{ x } M$
$D^+ = \{1, \ldots, 9\}$
$D = \{0, \ldots, 9\}$
$M = \{no, h, t, u\}$

Given $\langle N_h, N_t, N_u, M \rangle \in S$

- $N_h \in D^+$
- $N_t, N_u \in D$
- { *no* = no change, *h* = hundreds, *t* = tenths, u = units }

num$(\langle N_h, N_t, N_u, \_ \rangle)$ =
100*N_h + 10*N_t + N_u

Initial state: $\langle I_h, I_t, I_u, no \rangle$
Goal state: $\langle G_h, G_t, G_u, \_ \rangle$

$num(\langle I_h, I_t, I_u, no \rangle) = I$; and
$num(\langle G_h, G_t, G_u, \_ \rangle)$ = G

Example: Forbidden numbers

Specify the **operators**

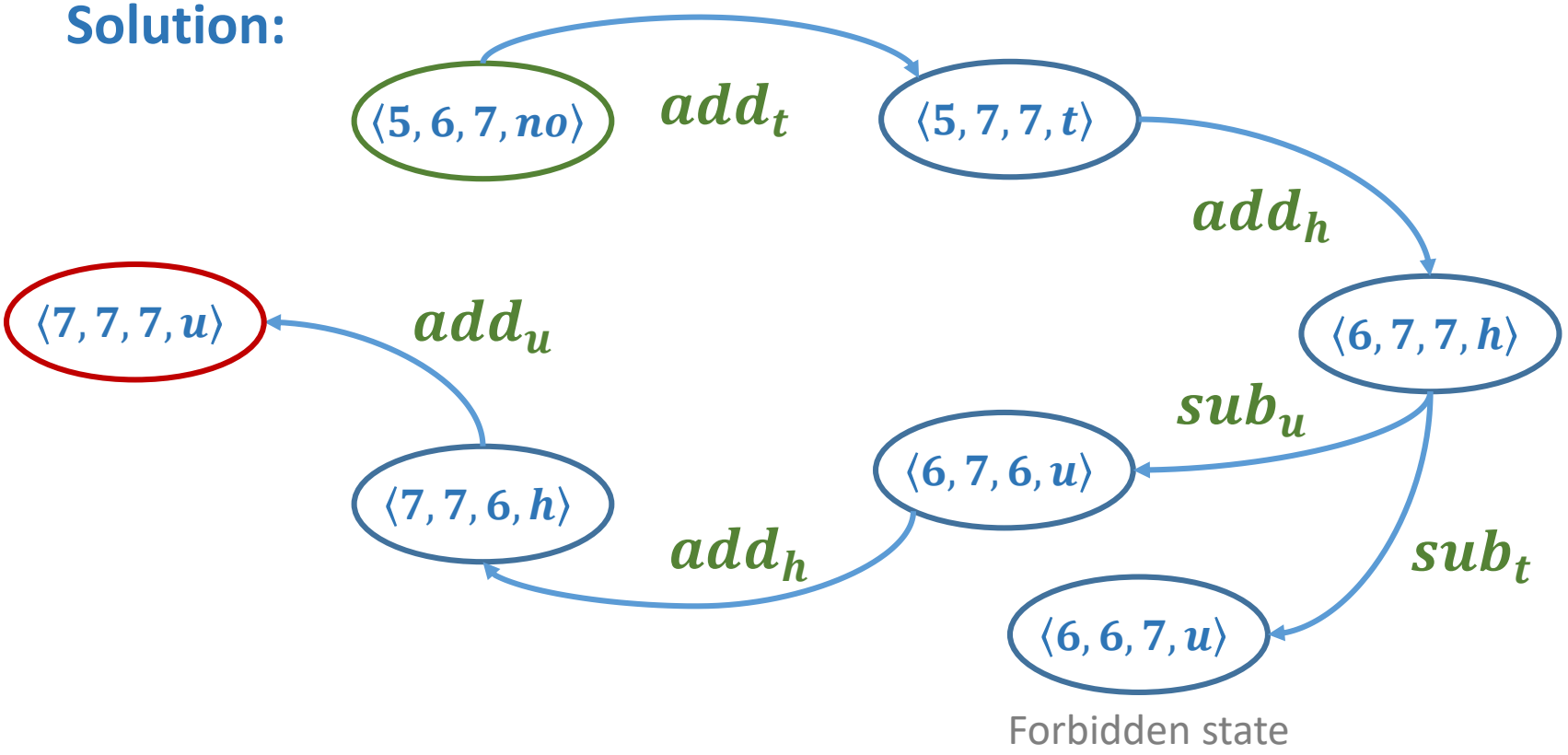
$$\langle N_h, N_t, N_u, M \rangle \in S$$

Op	Conditions	New state
add_u	$M \neq u, N_u \neq 9, num(s') \notin Forbidden,$	$s' = \langle N_h, N_t, N_u + 1, u \rangle$
sub_u	$M \neq u, N_u \neq 0, num(s') \notin Forbidden,$	$s' = \langle N_h, N_t, N_u - 1, u \rangle$
add_t	$M \neq t, N_t \neq 9, num(s') \notin Forbidden,$	$s' = \langle N_h, N_t + 1, N_u, t \rangle$
sub_t	$M \neq t, N_t \neq 0, num(s') \notin Forbidden,$	$s' = \langle N_h, N_t - 1, N_u, t \rangle$
add_h	$M \neq h, N_h \neq 9, num(s') \notin Forbidden,$	$s' = \langle N_h + 1, N_t, N_u\ h \rangle$
sub_h	$M \neq h, N_h \neq 1, num(s') \notin Forbidden,$	$s' = \langle N_h - 1, N_t, N_u, h \rangle$

Example: Forbidden numbers

- Forbidden set: {666, 667}; ***I*= 567, *G*= 777.**

Solution:



Example: Forbidden numbers

Find a good **heuristics**

Let g $= \langle G_h, G_t, G_u, \_ \rangle$ and s $= \langle S_h, S_t, S_u, \_ \rangle$

$$h(s) = |G_h - S_h| + |G_t - S_t| + |G_u - S_u|$$

It is **admissable**, relaxed problem
at least $|G_h - S_h|$ changes to the hundred digit
at least $|G_t - S_t|$ changes to the tenth digit
at least $|G_u - S_u|$ changes to the unit digit

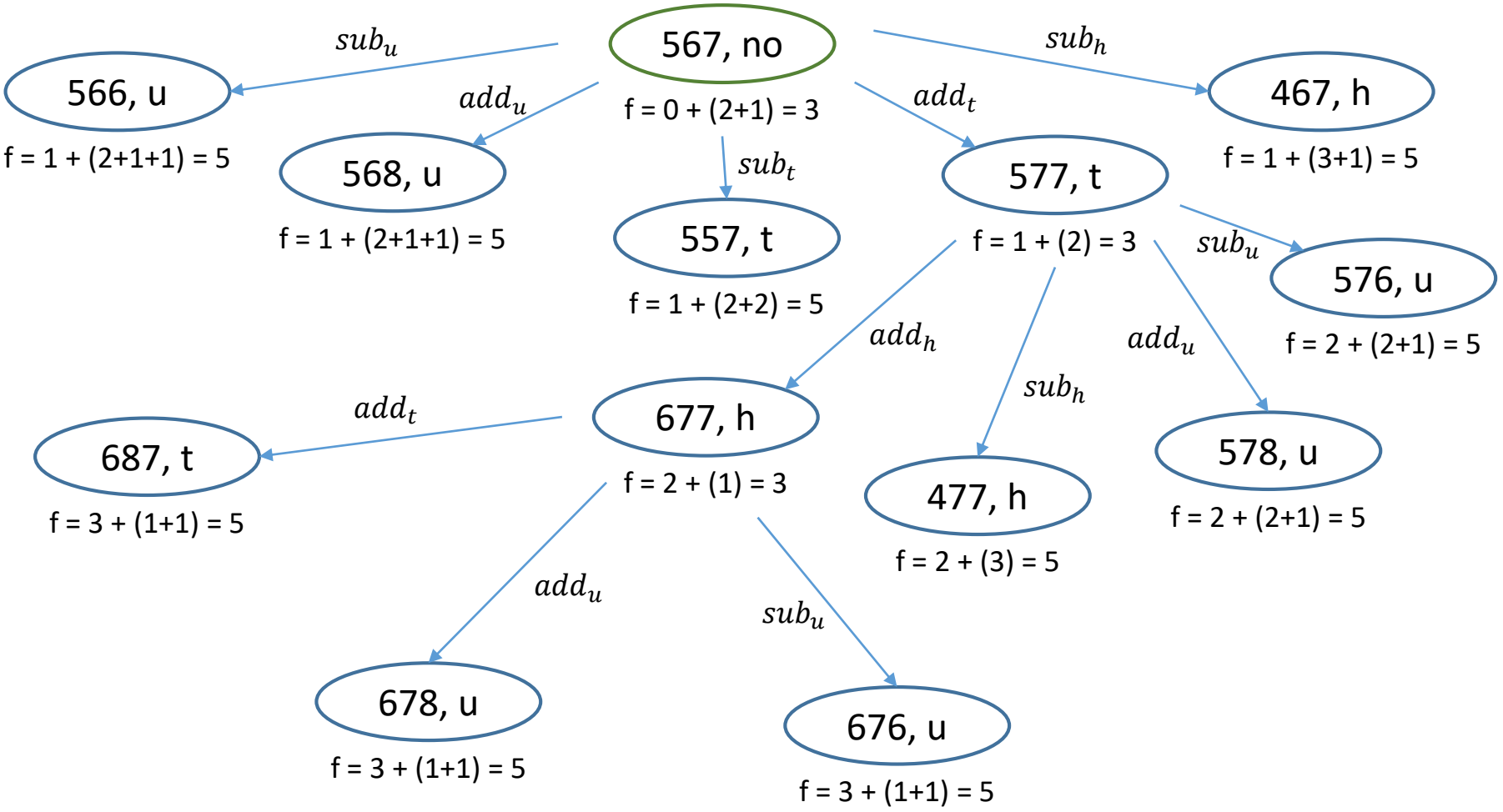
Each operator has **cost *c = 1***

Example: Forbidden numbers

A* tree

G=777, Forbidden set: {666, 667}

$$f(s) = g(s) + h(s), \qquad h(s) = |G_h - S_h| + |G_t - S_t| + |G_u - S_u|$$

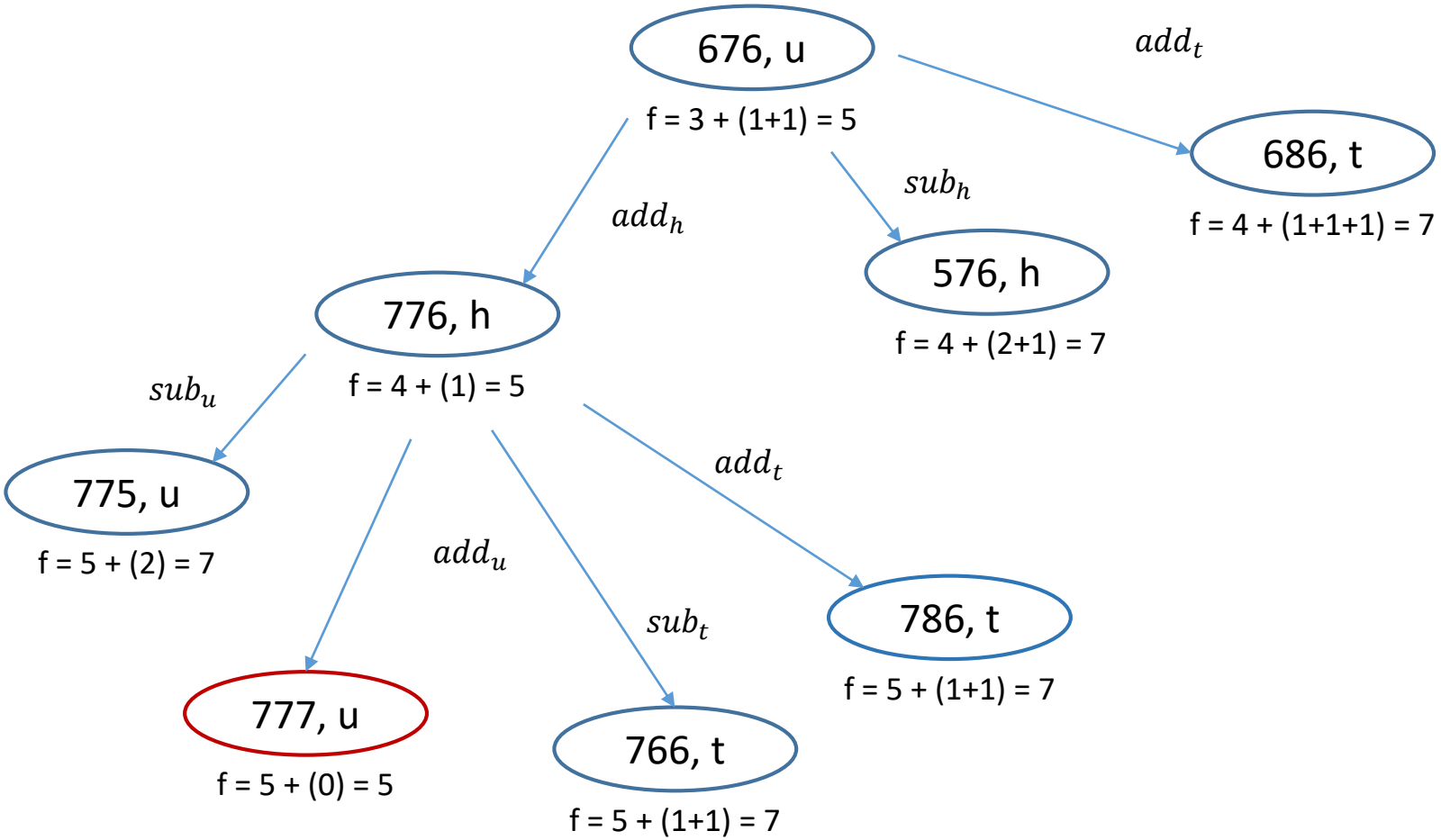


Example: Forbidden numbers

A* tree

G=777, Forbidden set: {666, 667}

$f(s) = g(s) + h(s), \qquad h(s) = |G_h - S_h| + |G_t - S_t| + |G_u - S_u|$



Example: Missionaries and Cannibals

3 missionaries and **3 cannibals** are on one of the sides of a river with a small **boat that can hold 2 passengers at most**. The **number of missionaries must always be more or equal wrt the number of cannibals**, otherwise they are eaten by the cannibals. How can the missionaries cross the river without being eaten?

1. Characterize the **state space**
2. Specify the **operators**
3. Find a minimal sequence of moves to solve the problem
4. Find a good **heuristics** to be used by A*.
5. Draw the **search tree generated by A***. For each node indicate: the number (state), the cost (f, g, and h) and an integer indicating the expansion order.

Example: Missionaries and Cannibals

3 missionaries and **3 cannibals** are on one of the sides of a river with a small **boat that can hold 2 passengers at most**. The **number of missionaries must always be more or equal wrt the number of cannibals**, otherwise they are eaten by the cannibals. How can the missionaries cross the river without being eaten?

1. Characterize the **state space**

$S = N^6$, with $\langle M_a, C_a, B_a, M_b, C_b, B_b \rangle$ denoting the number of missionaries and cannibals on the starting (a) and ending side (b)
Initial state: $\langle 3, 3, 1, 0, 0, 0 \rangle$, Goal state: $\langle 0, 0, 0, 3, 3, 1 \rangle$

Example: Missionaries and Cannibals

Operators:

carry(*m*, *c*):

meaning: carry *m* missionaries and *c* cannibals from a to b

Preconditions*:

- Boat side: $B_a = 1$
- Boat capacity: $1 \leq m + c \leq 2$
- Negative quantities:
  $$m \leq M_a \quad and \quad c \leq C_a$$
- Avoid eatings:
  $$M_a - m = 0 \;\; or \;\; C_a - c \;\; \leq M_a - m$$
  $$M_b + m = 0 \;\; or \;\; C_a + c \;\; \leq M_a + m$$

new state:

$$\langle M_a - m, C_a - c, 0, M_b + m, C_b + c, 1 \rangle$$

carryback(*m*, *c*):

meaning: carryback *m* missionaries and *c* cannibals from b to a

preconditions:

- Boat side: $B_b = 1$
- Boat capacity: $1 \leq m + c \leq 2$
- Negative quantities:
  $$m \leq M_b \quad and \quad c \leq C_b$$
- Avoid eatings:
  $$M_a + m = 0 \;\; or \;\; C_a + c \;\; \leq M_a + m$$
  $$M_b - m = 0 \;\; or \;\; C_a - c \;\; \leq M_a - m$$

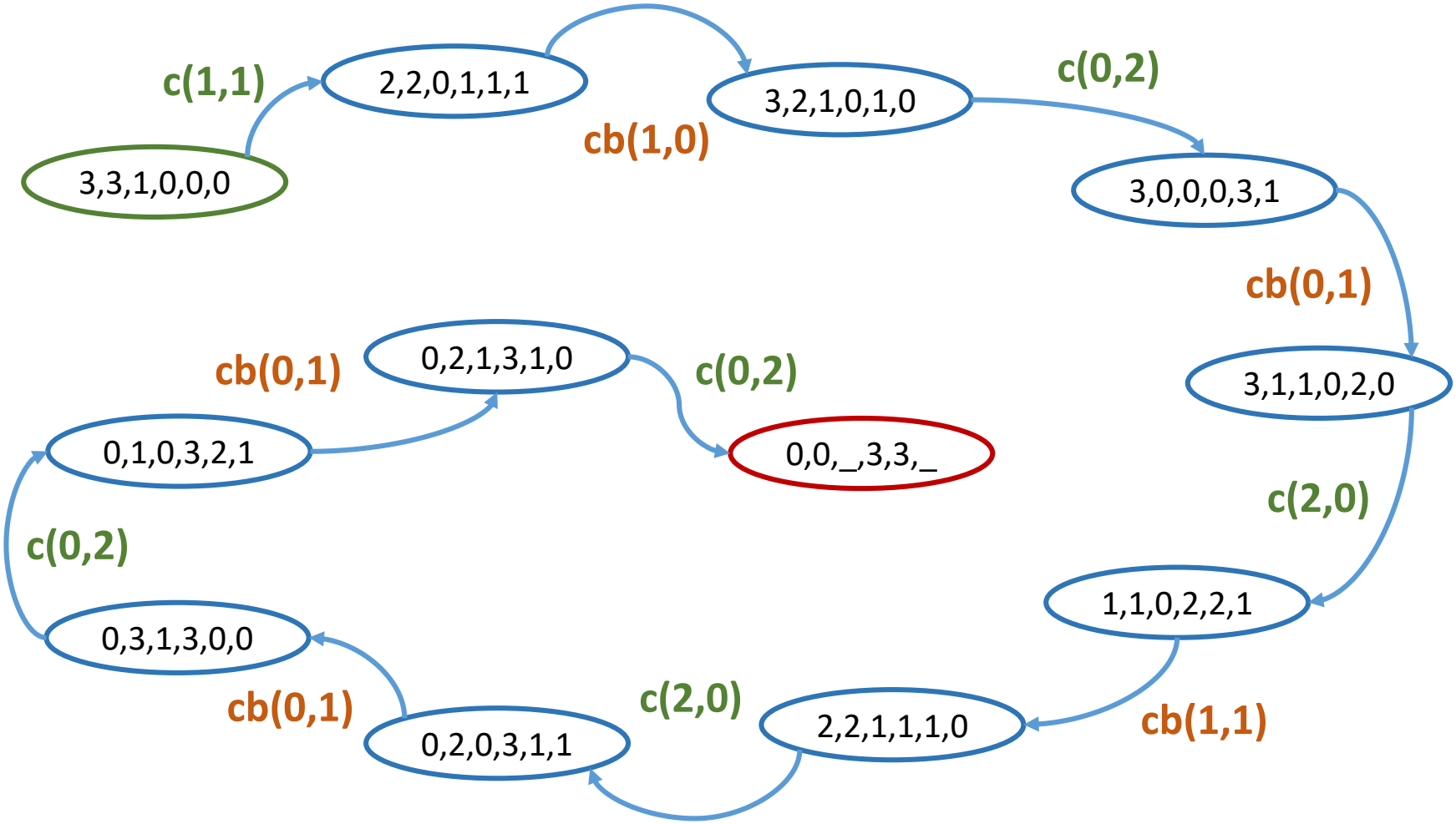
new state:

$$\langle M_a + m, C_a + c, 1, M_b - m, C_b - c, 0 \rangle$$

**preconditions* represent the possible configurations of the state-space that allow to execute an action

Example: Missionaries and Cannibals

Possible solution



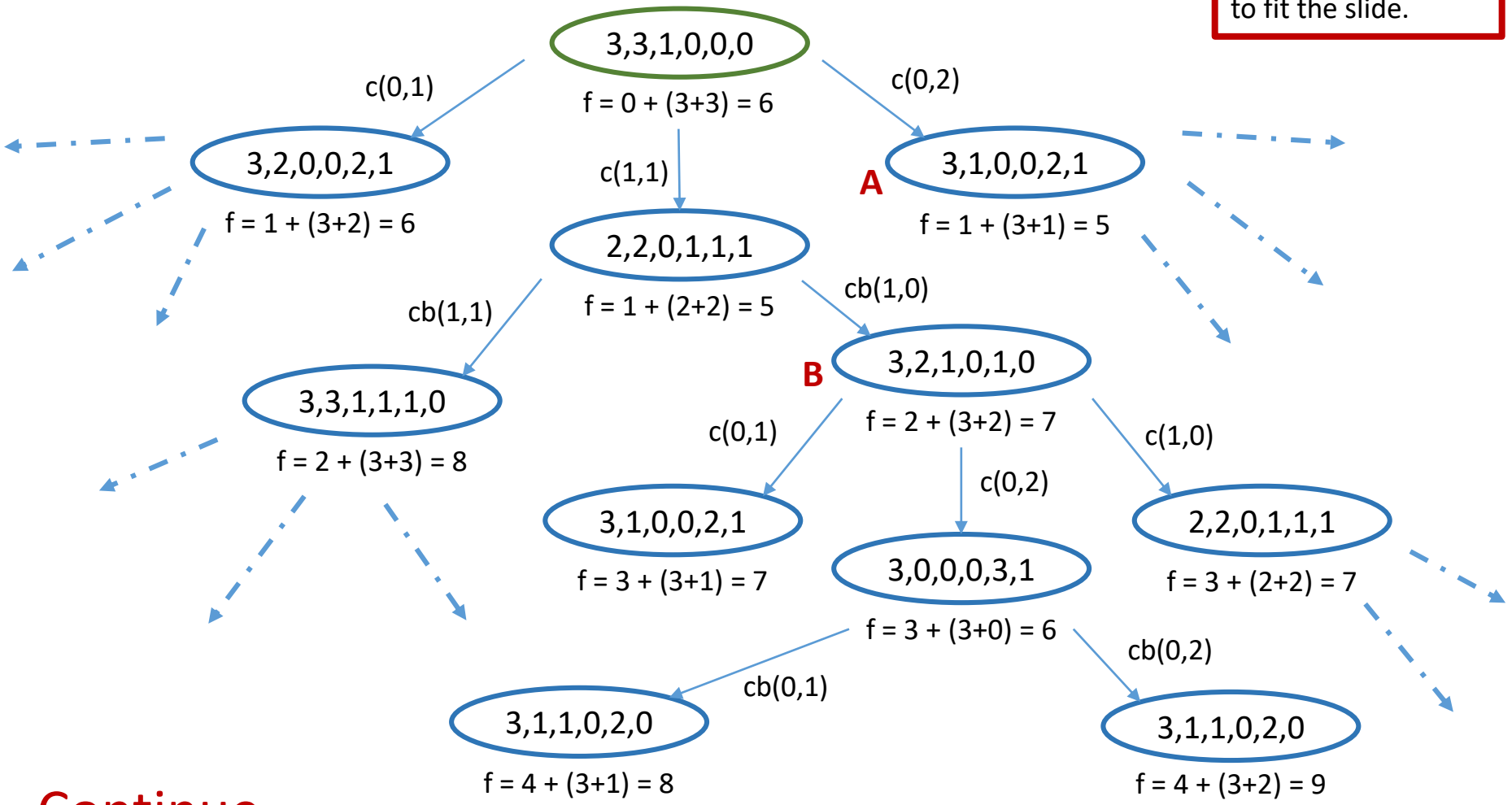
Example: Missionaries and Cannibals

A* solution

$$f(s) = g(s) + h(s),$$
$$h(s) = |3 - M_b| + |3 - C_b|$$

0,0,_,3,3,_

Note. A* would expand first node **A**, and then continue on node **B**. Here, that part of the tree has been removed to fit the slide.



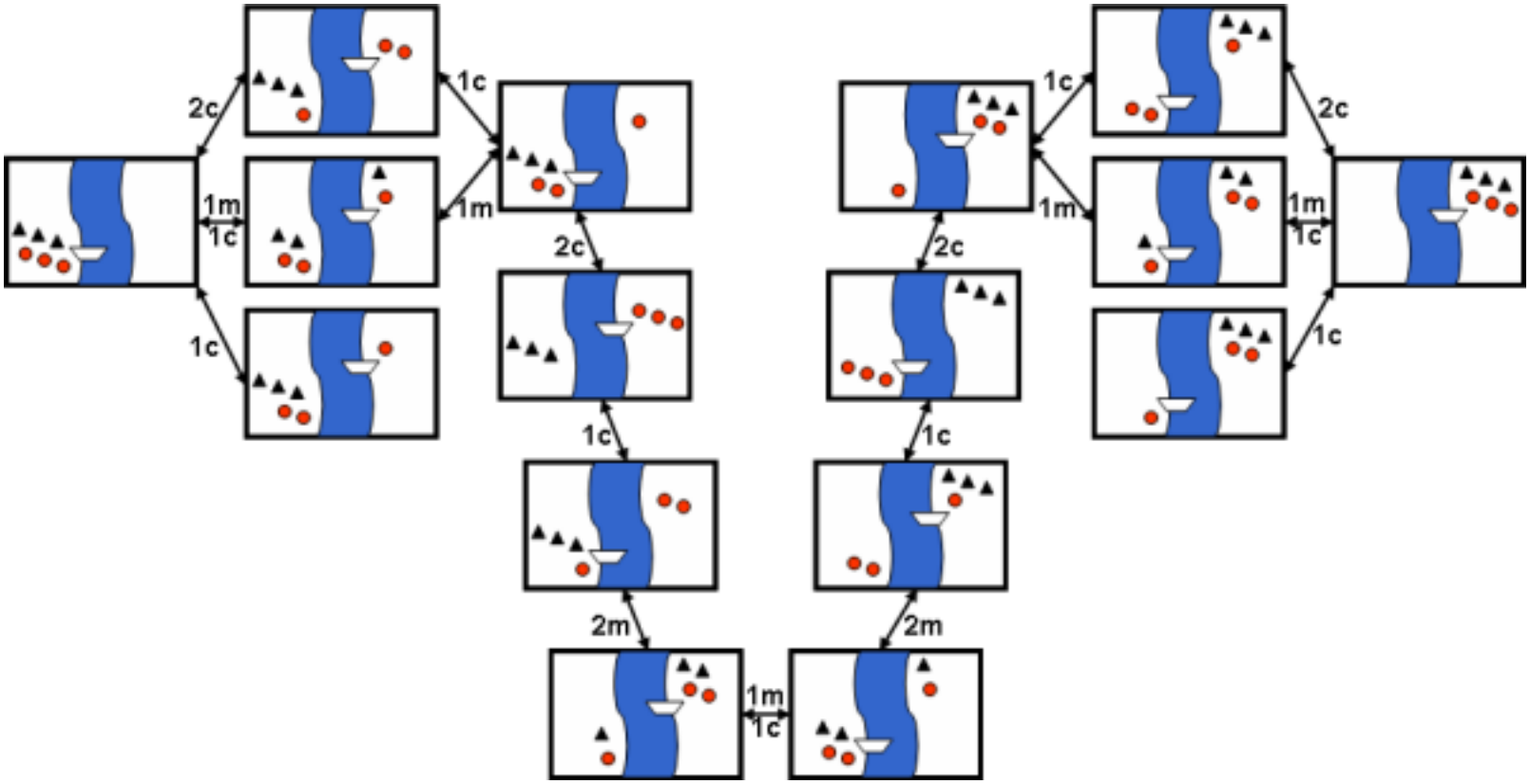
Continue…

Example: Missionaries and Cannibals

A* solution

$$f(s) = g(s) + h(s),$$
$$h(s) = |3 - M_b| + |3 - C_b|$$

0,0,_,3,3,_

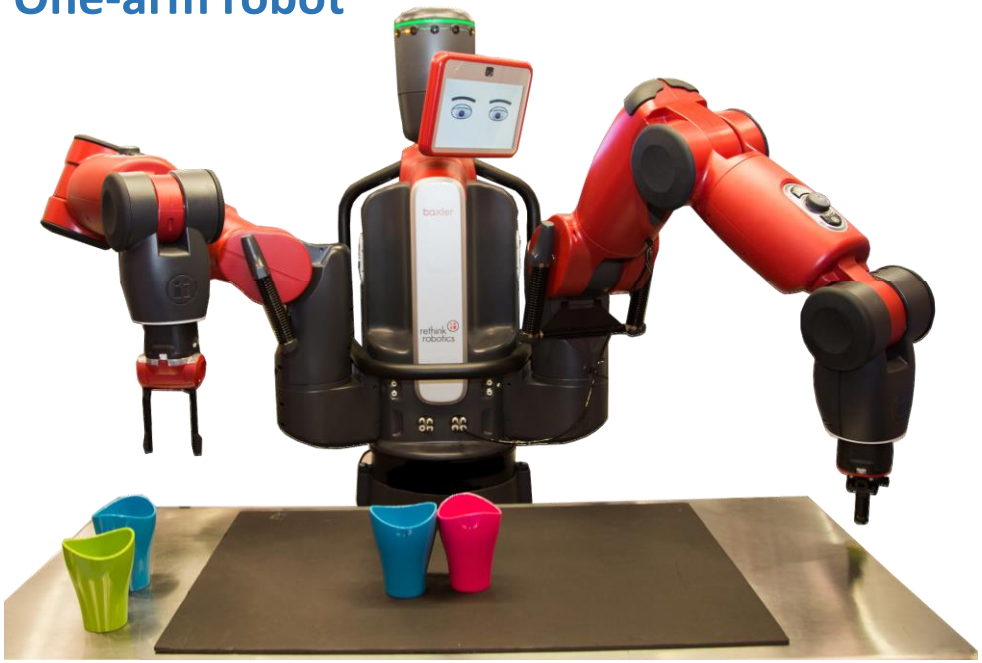


Exercises

- **State** representation
- **Operator** specification
- **Search** of solution:

  *breadth-first, depth-first, iterative deepening, A**

1. One-arm robot



2. Tower of Hanoi

3. Implement A* with your favorite programming language