



**OPPA European Social Fund
Prague & EU: We invest in your future.**

PLÁNOVÁNÍ A HRY - CV 1

kopriva@agents.felk.cvut.cz

Course Info

- **Gerhard Wickler lectures on planning**
 - First week of term
 - Mon 16:15 – 19:30, T2:C3-54 (Dejvice)
 - Tue 16:15 – 19:30, K-112 (Vyčichlova)
 - Wed 16:15 – 19:30, K-112 (Vyčichlova)
 - Thu 16:15 – 19:30, K-112 (Vyčichlova)
 - Fri 12:45 – 16:00, K-112 (Vyčichlova)
- **Game Theory lectures**
 - In the second half of the term
 - Date to be specified

Grading

- To get **Zápočet**
 - Each tutorial short assessment (10 together)
 - Assessments graded 1/0
 - Need to get 70% to get Zápočet
- **Exam**
 - 20% - Assignment in the 6th week of term
 - 80% - Exam

Course Preparation / Recap

- Algorithm Properties
- Searches
- Logics
- Satisfiability Problem

A decorative horizontal bar at the top of the slide, consisting of an orange rectangular segment on the left and a blue rectangular segment on the right.

ALGORITHM PROPERTIES

Algorithm Properties

□ **Soundness**

- The result returned by the algorithm is a solution to the problem

□ **Completeness**

- If a solution exists, the algorithm finds it

□ **Admissibility**

- It is guaranteed that the algorithm finds the optimal solution
- Optimality has to be defined



SEARCHING

Search Space

- **Search Space** S is a set of states, where the goal is to find the states that satisfy the condition g .
- Formally the **problem** is defined as a tuple (s_0, g, O) , where:
 - s_0 is the initial state
 - g is the goal condition
 - O is a set of state – transition operators

Breadth – First Search

□ Is **complete**

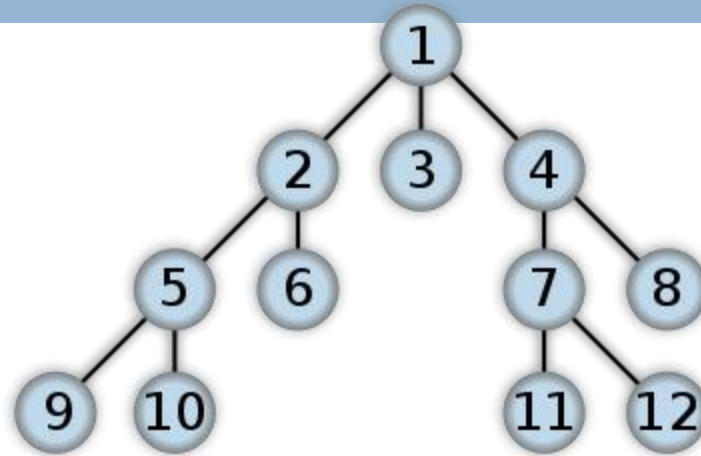
□ **Complexity**

▣ **Time** $O(b^d)$

▣ **Space** $O(b^d)$

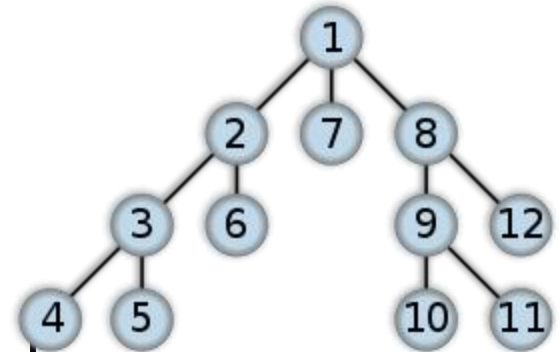
▣ **b** is the number of siblings of each node

▣ **d** is the depth of the search space



Depth-First Search

- Is complete
 - if no endless paths are present
- **Complexity**
 - **Time** depends on the way of the search
 - **Space** $O(d)$
 - **d** is the depth of the search space



A*

- $f'(n) = g(n) + h'(n)$
- $g(n)$ - total distance it has taken to get from the starting position to the current location
- $h'(n)$ - the estimated distance from the current position to the goal destination/state. A heuristic function is used to create this estimate on how far away it will take to reach the goal state.

First-order logic

- Whereas propositional logic assumes the world contains **facts**,
- first-order logic (like natural language) assumes the world contains
 - **Objects:** people, houses, numbers, colors, baseball games, wars, ...
 - **Relations:** red, round, prime, brother of, bigger than, part of, comes between, ...
 - **Functions:** father of, best friend, one more than, plus, ...

Syntax of FOL: Basic elements

- Constants KingJohn, 2, NUS,...
- Predicates Brother, >,...
- Functions Sqrt, LeftLegOf,...
- Variables x, y, a, b, \dots
- Connectives $\neg, \Rightarrow, \wedge, \vee, \Leftrightarrow$
- Equality $=$
- Quantifiers \forall, \exists

Atomic sentences

Atomic sentence = *predicate* ($term_1, \dots, term_n$)
or $term_1 = term_2$

Term = *function* ($term_1, \dots, term_n$)
or *constant* or *variable*

- E.g., $Brother(KingJohn, RichardTheLionheart) >$
 $(Length(LeftLegOf(Richard)), Length(LeftLegOf(KingJohn)))$

Complex sentences

- Complex sentences are made from atomic sentences using connectives

□

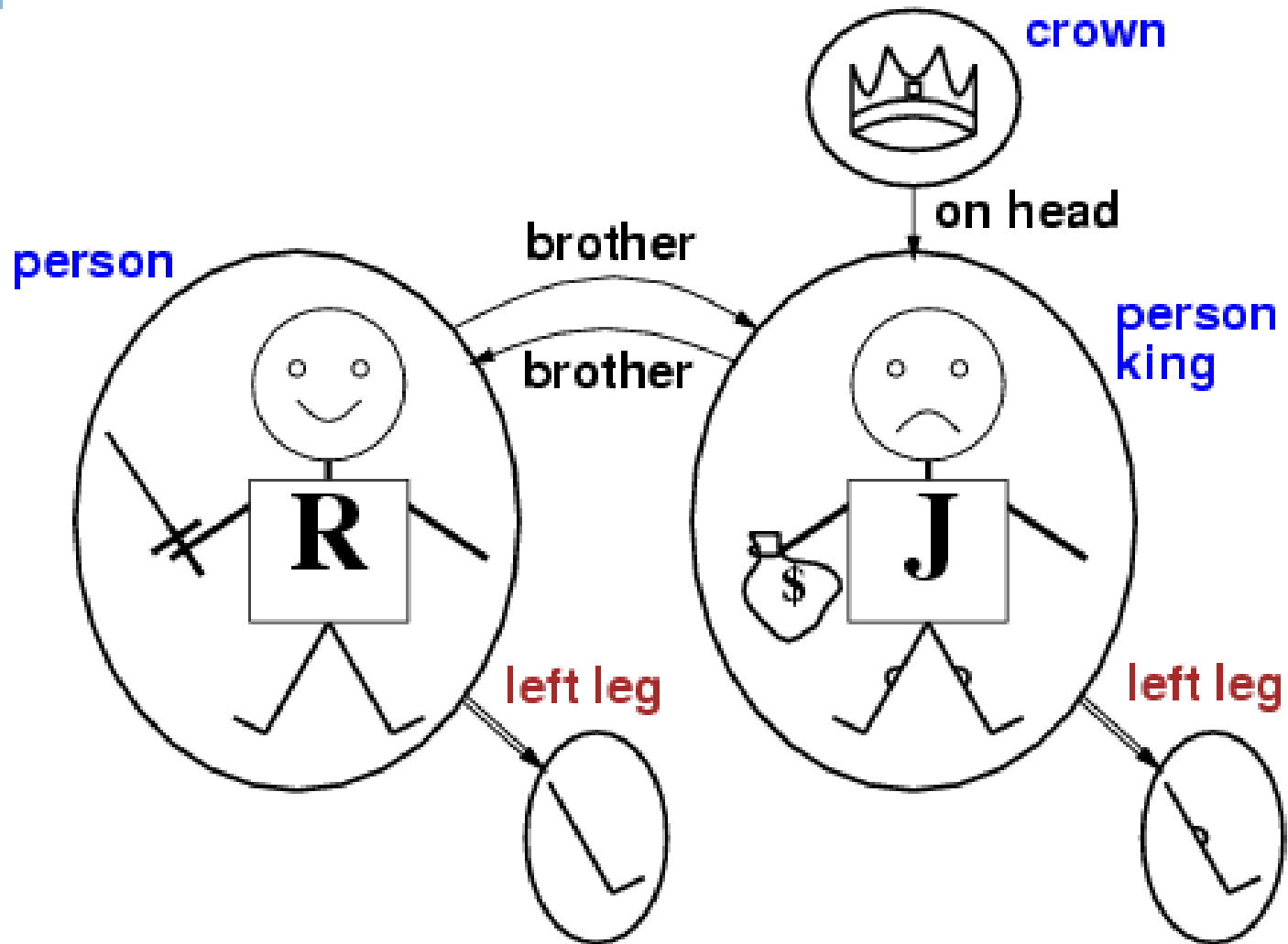
$$\neg S, S_1 \wedge S_2, S_1 \vee S_2, S_1 \Rightarrow S_2, S_1 \Leftrightarrow S_2,$$

E.g. $Sibling(KingJohn, Richard) \Rightarrow$
 $Sibling(Richard, KingJohn)$

$$>(1,2) \vee \leq (1,2)$$

$$>(1,2) \wedge \neg >(1,2)$$

Models for FOL: Example



Universal quantification

- $\forall \langle \text{variables} \rangle \langle \text{sentence} \rangle$
- Everyone at NUS is smart: $\forall x \text{ At}(x, \text{CVUT}) \Rightarrow \text{Smart}(x)$
- $\forall x P$ is true in a model m iff P is true with x being each possible object in the model
- Roughly speaking, equivalent to the **conjunction** of **instantiations** of P

A common mistake to avoid

- Typically, \Rightarrow is the main connective with \forall
-
- Common mistake: using \wedge as the main connective with \forall :
 $\forall x \text{ At}(x, \text{CVUT}) \wedge \text{Smart}(x)$
means “Everyone is at CVUT and everyone is smart”

Existential quantification

- $\exists \langle \text{variables} \rangle \langle \text{sentence} \rangle$
- Someone at CVUT is smart:
- $\exists x \text{ At}(x, \text{CVUT}) \wedge \text{Smart}(x)$
-
- $\exists x P$ is true in a model m iff P is true with x being some possible object in the model
-
- Roughly speaking, equivalent to the **disjunction** of **instantiations** of P

Another common mistake to avoid

- Typically, \wedge is the main connective with \exists
- Common mistake: using \Rightarrow as the main connective with \exists :
- $\exists x \text{ At}(x, \text{CVUT}) \Rightarrow \text{Smart}(x)$
 - is true if there is anyone who is not at CVUT!

Equality

- $term_1 = term_2$ is true under a given interpretation if and only if $term_1$ and $term_2$ refer to the same object
- E.g., definition of *Sibling* in terms of *Parent*:
-
- $\forall x,y \text{ Sibling}(x,y) \Leftrightarrow [\neg(x = y) \wedge \exists m,f \neg (m = f) \wedge \text{Parent}(m,x) \wedge \text{Parent}(f,x) \wedge \text{Parent}(m,y) \wedge \text{Parent}(f,y)]$

Satisfiability

- **Model** of the formula is a set of assignments of the true/false values to the variables in a way that the formula is evaluated to be **true**.
 - $\neg p$ is true iff p is false
 - $p \wedge q$ is true iff p is true and q is true
- Satisfiability problem (SAT) is a problem of evaluating, whether a model for the given formula exists.

3-SAT problem

- Conjunctive normal form
 - 3-CNF
- First known NP-complete problem
- $(x_{11} \text{ OR } x_{12} \text{ OR } x_{13}) \text{ AND}$
 $(x_{21} \text{ OR } x_{22} \text{ OR } x_{23}) \text{ AND}$
 $(x_{31} \text{ OR } x_{32} \text{ OR } x_{33}) \text{ AND}$
.....



**OPPA European Social Fund
Prague & EU: We invest in your future.**
