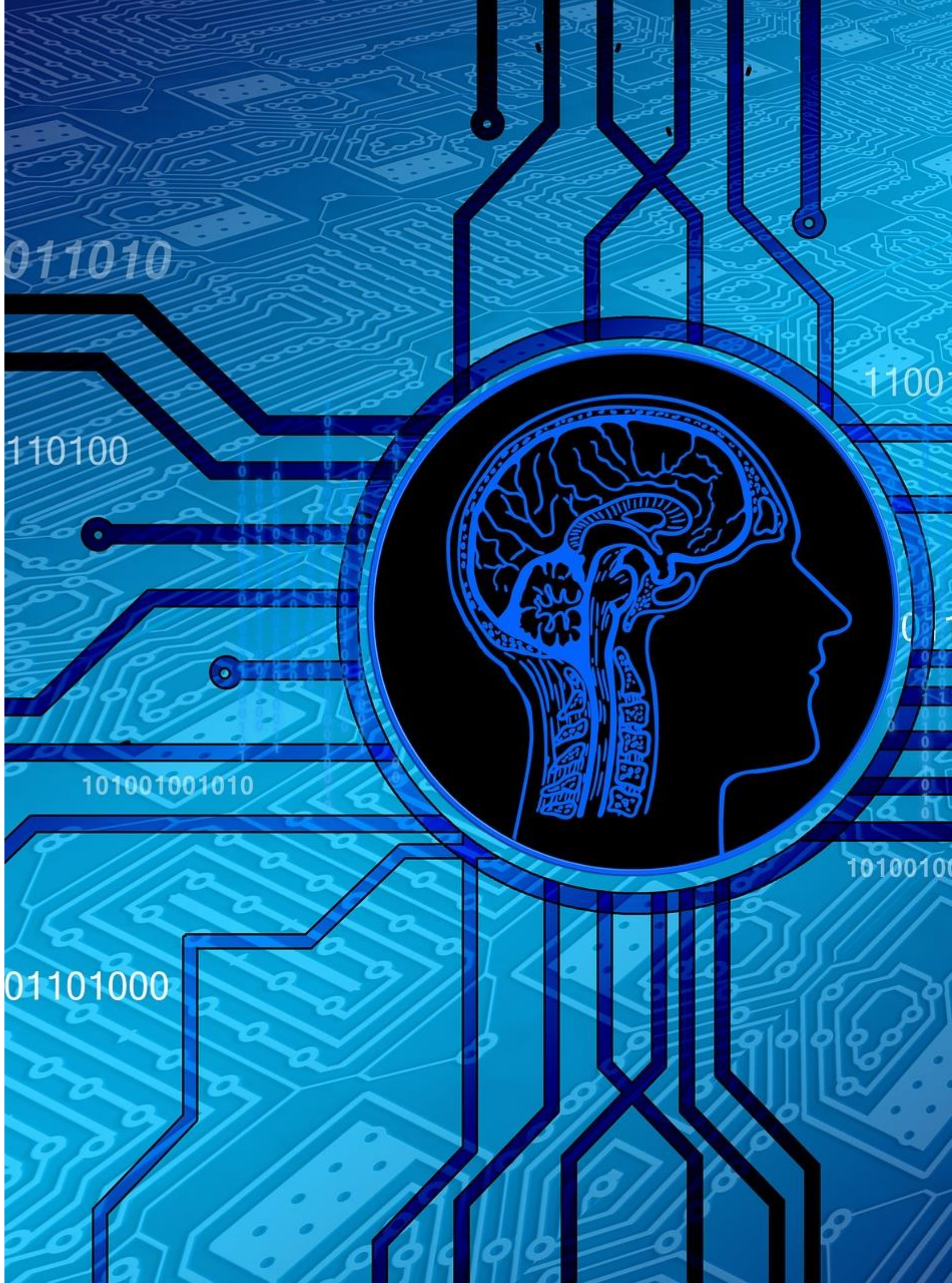
Revision

Petros Papapanagiotou

Informatics 2D: Reasoning and Agents
Lecture 15



Intelligent Agents and their Environments

Simple reflex agents

Model-based reflex agents

Goal-based agents

Utility-based agents

Learning agents

Properties of environments

- Partially vs. fully observable
- Deterministic vs. stochastic
- Episodic vs. sequential
- Static vs. dynamic
- Discrete vs. continuous
- Single vs. multi-agent

Problem Solving by Searching

Problem formulation usually requires abstracting away real-world details to define a state space that can feasibly be explored.

Variety of uninformed search strategies:

- breadth-first, depth-first, iterative deepening

Iterative deepening search uses only linear space and not much more time than other uninformed algorithms.

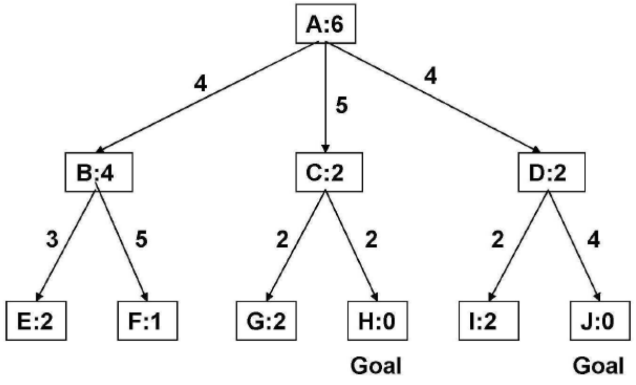
Informed Search and Exploration for Agents

Smart search based on heuristic scores

- Best-first search
- Greedy best-first search
- A* search
- Admissible heuristics and optimality.

Example

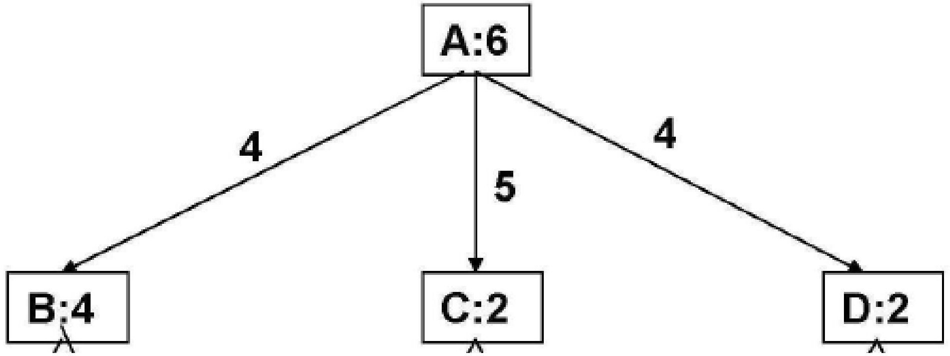
3. Consider the following search tree in which the nodes represent states and the arcs represent the moves connecting these states. Each node is labelled by a letter. The numbers on the arcs represent the *true* cost of the associated move. The numbers on the nodes represent the *estimated* cost of reaching the goal state from that node.



In which order would the A^* algorithm explore this search tree?

(a) A, B, C, D, I, J, G, H.
(b) A, B, C, D, I, J.
(c) A, C, H.
(d) A, D, I, J.
(e) A, C, G, H.

Example



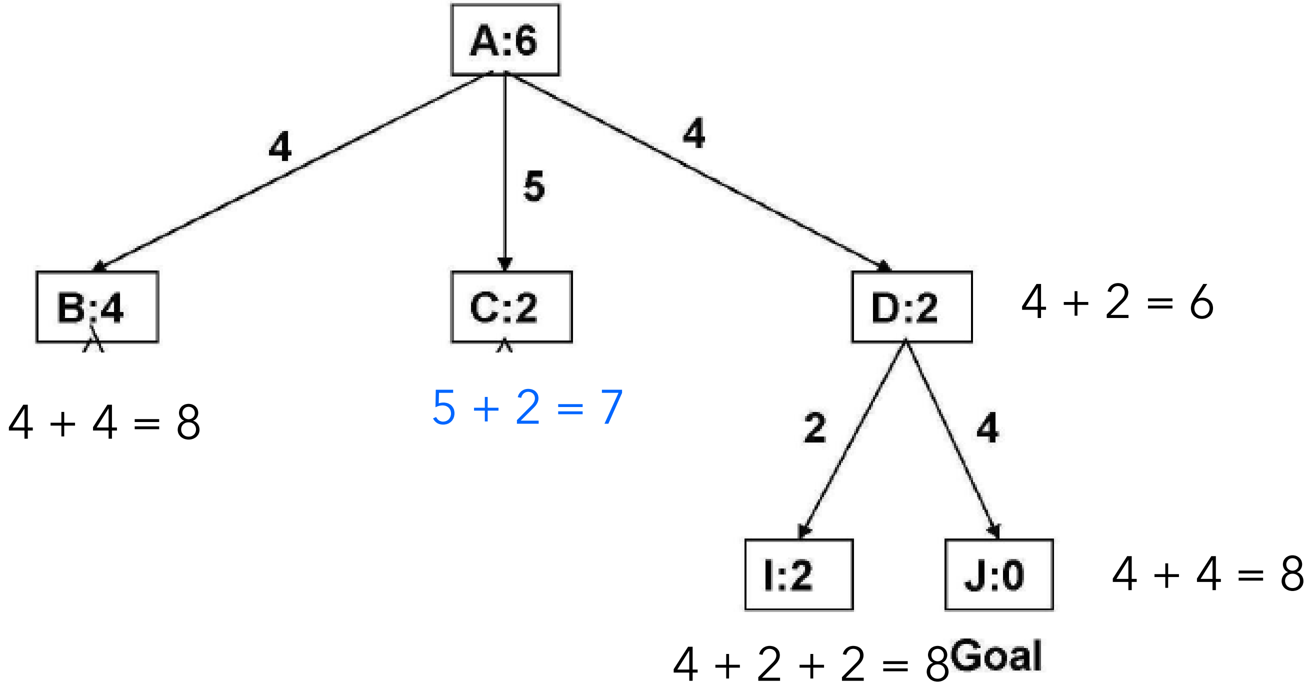
4 + 4 = 8

5 + 2 = 7

4 + 2 = 6

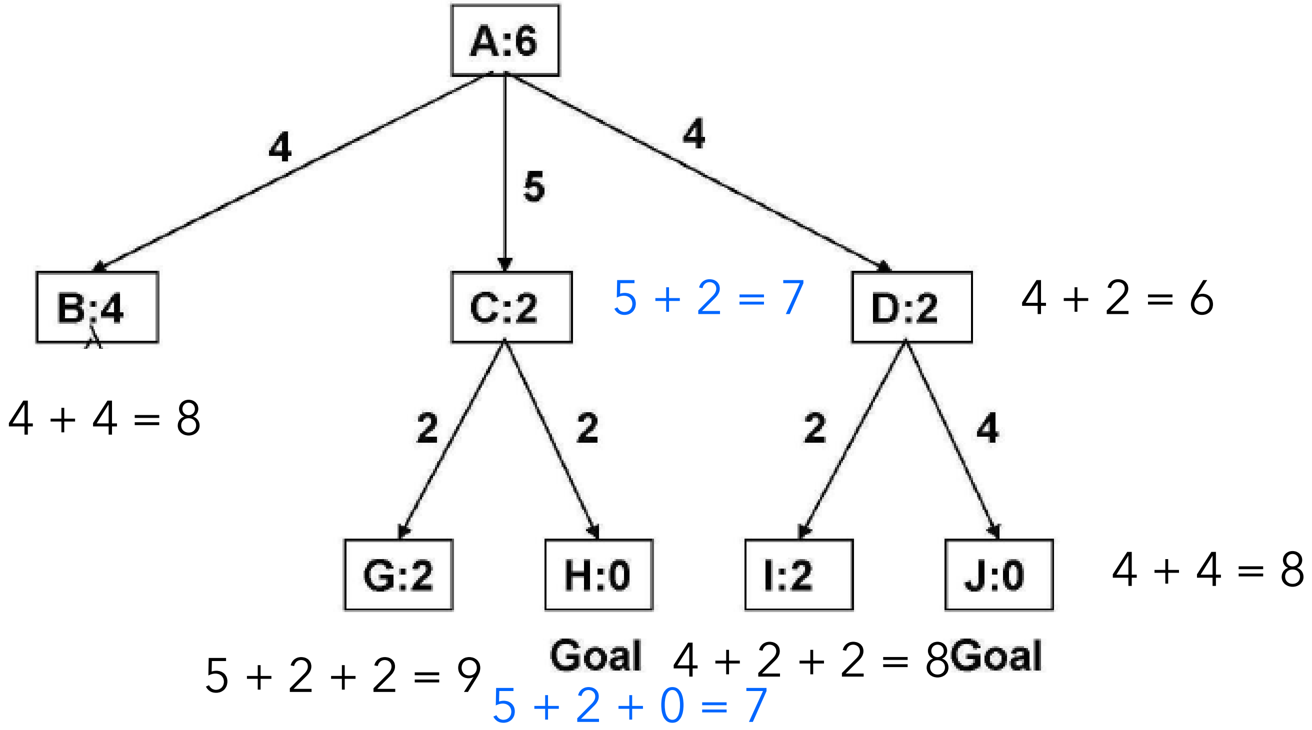
A
B C D

Example



A
B C D
I J

Example



A
B C D
I J
G H

We're done as we've expanded a node containing a goal state

Smart Searching Using Constraints

CSPs are a special kind of problem:

- states defined by values of a fixed set of variables
- goal test defined by constraints on variable values

Backtracking = depth-first search with one variable assigned per node.

Variable ordering and value selection heuristics help significantly.

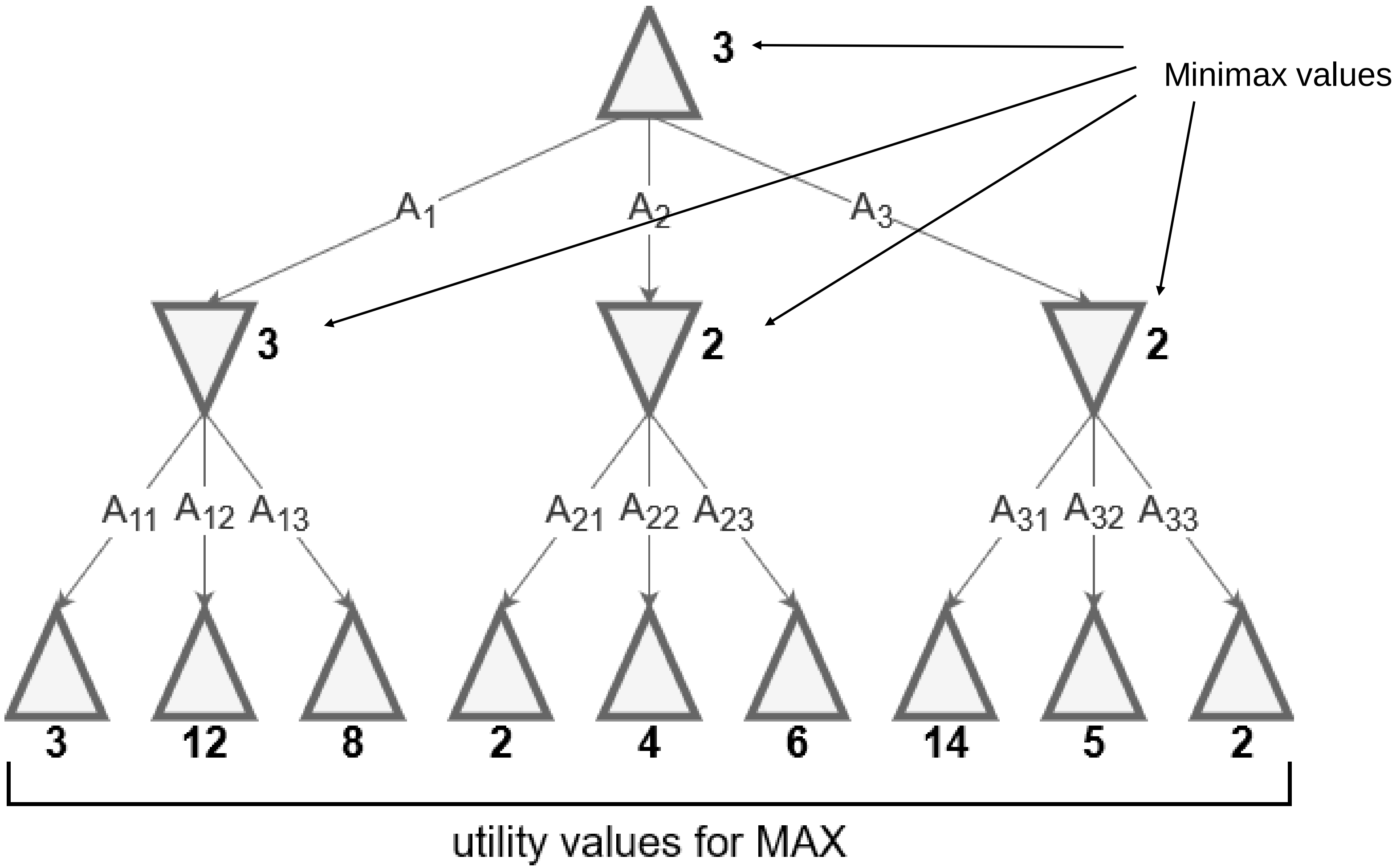
Forward checking prevents assignments that guarantee later failure.

Constraint propagation (e.g. arc consistency) does additional work to constrain values and detect inconsistencies.

Adversarial Search

Minimax assumes that both players play optimally

- Informally, each agent is making its decision for the next move based on the assumption that the other agent is playing as well as it can.
- If not things can go wrong!



Adversarial Search (Contd)

α-β Pruning and its properties

Reasoning about relevant computations only enables search space to be pruned.

How to deal with deep trees: need for evaluation functions.

Logical Agents: Knowledge Bases and the Wumpus World

Logical agents apply inference to a knowledge base to derive new information and make decisions.

Basic concepts of logic:

- syntax: formal structure of sentences
- semantics: truth of sentences with respect to models
- entailment: necessary truth of one sentence given another
- inference: deriving sentences from other sentences
- soundness: derivations produce only entailed sentences
- completeness: derivations can produce all entailed sentences

Wumpus world requires the ability to represent partial and negated information, reason by cases, etc.

Propositional logic solves many problems, but lacks expressive power.

Effective Propositional Inference

Logical agents apply inference to a knowledge base to derive new information and make decisions.

Two algorithms: DPLL & WalkSAT

- DPLL is a decision procedure, i.e. it will return true (yes) or false (no) for a set of propositional clauses (cf. complete algorithm)

Tautology refers to a valid statement i.e. one that is true under all possible interpretations

- e.g. $P \lor \neg P$ or $P \Rightarrow P$

A sentence is satisfiable if there exists an interpretation that makes it true

- e.g. $P \land Q$ is true with P true and Q true. It is false if any of P or Q is false.

Hard satisfiability problems.

First-Order Logic

First-order logic:

- objects and relations are semantic primitives
- syntax: constants, functions, predicates, equality, quantifiers.
- Informally:
  - Predicates (applied to terms) have a truth value (i.e. true or false)
    - e.g. < (less than) is a predicate so, $x < 3 + 5$ is either true or false
  - Functions just construct new terms out of other terms
    - So, a function such as + (addition) can construct a new term $3 + 5$. This does not have a truth value.

Increased expressive power: sufficient to define Wumpus world.

Unification and Generalised Modus Ponens

Rules for quantifiers

- Need to be able to state rules such as UI, EI etc.

Reducing FOL to PL

Unification as equation solving

- How to apply rules to 2 expressions produce unifier
- All steps should be clearly labelled

Generalised modus ponens (GMP)

$$\frac{p_1', p_2', \ldots, p_n', (p_1 \wedge p_2 \wedge \ldots \wedge p_n \Rightarrow q)}{q\theta} \qquad \text{where } p_i'\theta \equiv p_i\theta \text{ for all } i$$

Example:

p_1' is *King*(*John*)	p_1 is *King*(x)
p_2' is *Greedy*(y)	p_2 is *Greedy*(x)
θ is {x/John,y/John}	q is *Evil*(x)
$q\theta$ is *Evil*(*John*)	

- GMP used with KB of definite clauses (exactly one positive literal)
- All variables assumed universally quantified

Inference

Forward chaining

- Informally: Unify all the assumptions in an implication rule with facts in KB to discharge them. If successful, add instantiated conclusion to KB. This is similar to a discovery process.

Backward chaining

- Informally: Unify conclusion of an implication rule with some fact in KB. If successful, add instantiated assumptions to KB as new goals. This is similar to a decomposition into sub-problems.

Resolution

- Formal statements for various types of resolution.
- Refutation method i.e. works by looking for a contradiction i.e. tries to derive falsity (empty clause).
- Need to negate the goal before converting it to clausal form.

Example

9. Which of the following clauses is the result of resolving clause $P(G(y)) \vee \neg Q(G(y))$ with clause $\neg P(F(x)) \vee Q(x)$, assuming they can be resolved?

   (a) $P(G(y)) \vee \neg P(F(x))$.

   (b) $Q(y) \vee \neg Q(F(y))$.

   (c) Resolution fails.

   (d) $P(G(y)) \vee P(F(G(y))) \vee \neg Q(G(y))$.

   (e) $P(G(y)) \vee \neg P(F(G(y)))$.

$P(G(y)) \vee \neg Q(G(y))$ $\qquad$ $\neg P(F(x)) \vee Q(x)$

Identify unifiable, complementary literals

$\{x/G(y)\}$

Resulting unifier

$P(G(y)) \vee \neg P(F(G(y)))$

Substitute to get resolvant

Situation Calculus

Planning

Situations

- These are not states of the world, despite what their names might suggest. A situation is a (history of a) sequence of actions.
- Reiter:
  - *The central ontological ingredient of the sitcalc is the situation. Even at this late stage in AI, many people still don't understand what a situation is, so here's the secret: A situation is a finite sequence of actions. Period. It's not a state, it's not a snapshot, it's a history.*
- E.g. Result(Move(A, B), Result(Move(B,C),S0))

Frame problem

- How to formulate the various properties logically etc.