

## Expected knowledge from course NAIL069 Artificial Intelligence I.

Version 30.12.2024

Intelligent agents, environment, and structure of agents:

- Define an agent and a rational agent.
- Explain properties of environment (partial/full observability, deterministic/stochastic, episodic/sequential, static/dynamic, discrete/continuous, single/multi-agent).
- Explain typical agent structures and their differences (single reflex, model-based reflex, goal-based, utility-based).
- Explain possible representations of states (atomic, factored, structured) and give examples of their usage.

Problem solving and uninformed search:

- Formulate a well-defined problem and its (optimal) solution, give some examples.
- Explain and compare tree search and graph search, discuss memory vs time.
- Explain and compare node selection strategies (depth-first, breadth-first, uniform-cost).
- Explain depth-limited search and iterative deepening and why they are used.
- Define backward search, bidirectional search, and discuss their difficulties.
- Describe how to handle partial observability (belief states) and non-determinism (AND-OR search).

Informed (heuristic) search:

- Explain evaluation function  $f$  and heuristic function  $h$ , give some examples.
- Describe greedy best-first search and its properties.
- Describe algorithm  $A^*$ , define and compare admissible and consistent (monotonous) heuristics, prove completeness and optimality of  $A^*$  implemented using tree-search and graph-search.
- Describe approaches to overcome memory issues of  $A^*$  (IDA\*, RBFS, SMBA\*)
- Describe suboptimal version of  $A^*$  (Weighted  $A^*$ ).
- Discuss how to obtain heuristics for  $A^*$  (relaxation, pattern databases), give some examples.

Local search:

- Explain the concept of local search.
- Describe hill-climbing (and its versions) and simulated annealing.
- Describe and compare population-based techniques (beam search, genetic algorithms).

On-line search:

- Formulate on-line search and compare it with off-line search.
- Explain competitive ratio and safely explorable environments.
- Describe algorithms on-line DFS and LRTA\*.

#### Constraint satisfaction:

- Define a constraint satisfaction problem (including the notion of constraint) and give some examples of CSPs.
- Apply problem solving techniques to CSPs, explain why a given technique is appropriate for CSPs.
- Explain principles of variable and value ordering and give examples of heuristics.
- Define arc consistency and show an algorithm to achieve it (AC-3).
- Define k-consistency and explain its relation to backtrack-free search; give an example of a global constraint.
- Explain forward checking and look ahead techniques.

#### Adversarial search and games:

- Explain core properties of environment and information needed to apply adversarial search.
- Explain and compare mini-max and alpha-beta search.
- Define an evaluation function and give some examples.
- Explain quiescent and horizon effect.
- Describe how stochastic games are handled (expected mini-max).
- Describe Monte Carlo tree search and explain exploration vs exploitation.

#### Knowledge representation and propositional logic:

- Define a knowledge-based agent.
- Define a formula in propositional logic, describe conjunctive and disjunctive normal forms (and how to obtain them), define Horn clauses.
- Explain the notions of model, entailment, inference, satisfiability, and their relations.
- Explain DPLL and WalkSAT algorithms (including the notions of pure symbol and unit clause).
- Describe resolution algorithm (and how it proves entailment) and explain forward and backward chaining as its special cases.

#### Knowledge representation and first-order logic:

- Define a formula in first-order logic, explain the role of constants, function and predicate symbols, universal and existential quantifiers and their relation.
- Describe a conjunctive normal form and show how to obtain it, explain standardization and Skolemization.
- Define the notion of substitution, explain unification and describe the unification algorithm.
- Describe reduction of first-order logic to propositional logic (grounding), discuss the role functions there.
- Describe resolution algorithm (including resolution strategies) and explain the role of answer literal.
- Describe forward and backward chaining (including pattern matching, Rete technique, and a magic set).

#### Knowledge engineering:

- Describe the knowledge engineering process.
- Describe how to efficiently represent objects (taxonomy).
- Describe situation calculus to model actions and time (representing situations, fluent and rigid predicates; possibility, effect, frame, and successor-state axioms).
- Describe frame and ramification problems and how solve them.

#### Automated planning:

- Define planning and projection tasks.
- Describe a classical representation of a planning problem (representation of states, planning operator vs action, transition function, goal function, planning domain, planning problem).
- Explain state-space planning (progression/forward planning, regression/backward planning).
- Explain plan-space planning (partial plan, solution plan, open goal, threat).