AIF recap

1 Agents

Agents interact with environments through **actuators** and **sensors**. The agent function describes what the agent does in all circumstances. The **performance** measure evaluates the environment sequence, a perfectly rational agent maximizes expected performance. Agent programs implement (some) agent functions.
PEAS (Performance, Environment, Actuators, Sensors) descriptions define task environment. **Environments** are categorized along several dimensions: observable, deterministic, episodic, static, discrete, single-agent.
Several basic agent **architectures** exist: reflex, reflex with state, goal-based, utility-based, etc.

2 Solving by Search

A problem consists of five parts: the **initial state**, a set of **actions**, a **transition model** describing the results of those actions, a **set of goal states**, and an action **cost function**.
Uninformed search methods have access only to the problem definition. Algorithms build a search tree in an attempt to find a solution.
Informed search methods have access to a heuristic function h(n) that estimates the cost of a solution from n.

3 Search in Complex Environment

Local search methods keep only a small number of states in memory that are useful for optimization.
In **nondeterministic environments**, agents can apply AND–OR search to generate contingency plans that reach the goal regardless of which outcomes occur during execution.
Belief-state is the set of possible states that the agent is in for partially observable environments.
Standard search algorithms can be applied directly to belief-state space to solve sensorless problems.

4 Constraint Satisfaction Problem

CSPs are a special kind of problem: states are defined by values of a fixed set of variables, goal test are defined by **constraints** on variable values.
Backtracking is a depth-first search with one variable assigned per node. Variable

ordering and value selection heuristics help significantly.
Forward checking keeps track of remaining legal values for unassigned variables. Terminate search when any variable has no legal values. **Constraint propagation** (e.g., arc consistency) repeatedly enforces constraints locally and early detects inconsistencies.
Local search using plateau search, min-conflicts, constraint weighting **heuristics** has also been applied to constraint satisfaction problems with great success.
The CSP representation allows analysis of problem structure. **Tree-structured** CSPs can be solved in linear time.

5 Games

Minimax algorithm: selects optimal moves by a depth-first enumeration of the game tree. We don't need to expand every path: **alpha–beta pruning** allows greater efficiency by eliminating subtrees that have worse minimax value.
Evaluation function is a heuristic that estimates utility of state.
Monte Carlo tree search (MCTS): no heuristic, play game to the end with rules and repeated multiple times to determine optimal moves during playout.

6 Logical Agents

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 wrt models;
- entailment: necessary truth of one sentence given another;
- inference: deriving sentences from other sentences;
- soundess: derivations produce only entailed sentences
- completeness: derivations can produce all entailed sentences

A **Horn clause** is a proposition symbol or a conjunction of symbols implying a symbol.
Forward, backward chaining are linear-time, complete for Horn clauses.
Resolution rule produces a new clause implied by two clauses containing complementary literals. It is complete for propositional logic but propositional logic lacks expressive power.
Local search methods (WALKSAT) find solutions (sound but not complete).

7 First Order Logic

First-order logic assumes the world contains:

- objects and relations are semantic primitives
- syntax: constants, functions, predicates, equality, quantifiers

Sentences are true with respect to a model and an interpretation.
Increased expressive power: sufficient to define wumpus world.
Situation calculus: conventions for describing actions and change in FOL, can formulate planning as inference on a situation calculus KB.
Developing a KB in FOL requires a careful process of analyzing the domain, choosing a vocabulary, and encoding the axioms required to support the desired inferences.

8 Inference in First Order Logic

Universal and **existential** instantiation produces a lot of irrelevant propositions.
Unification identify appropriate substitutions for variables, eliminates the instantiation step in first-order proofs, making the process more efficient in many cases.
Forward chaining is used in deductive databases, where it can be combined with relational database operations. It is also used in production systems.
Backward chaining is used in logic programming systems, which employ sophisticated compiler technology to provide very fast inference.
Prolog, unlike first-order logic, uses a closed world with the unique names assumption and negation as failure.
The **generalized resolution** inference rule provides a complete proof system for first order logic, using knowledge bases in conjunctive normal form.

9 Knowledge

Upper ontology is a general, flexible and hierarchical representation for complex domain based on categories and the event calculus.
Mental objects are knowledge in someone's head. **Modal logic** uses modal operators and needs models that are collections of the all possible believed worlds.
Special-purpose representation systems, such as **semantic networks** and **description logics** (CLASSIC language), have been devised to help in organizing a hierarchy of categories.
Nonmonotonic logics, such as circumscription and default logic, are intended to capture default reasoning in general.
Truth maintenance systems handle knowledge updates and revisions efficiently.
Cyc is a titanic effort trying to model common sense knowledge overcoming the limitations of knowledge engineering in Logic systems.

10 Prolog

Supports **symbolic**, **non-numerical** computation.
Facts and rules constitute the KB. Tries to satify the **queries** (goals) via **backtracking**. **Knowledge representation** is via states as a list of stack.

11 Planning

PDDL, the Planning Domain Definition Language, describes the initial and goal states as conjunctions of literals, and actions (schemas) in terms of their precondi-

tions and effects.
Some **heuristics** for planning are: ignore preconditions, remove negative literals from effects, pruning, relaxation and state abstraction.
Hierarchical task network (HTN) allows higher levels of abstraction with hierarchical decomposition to manage complexity.
Contingent plans allow the agent to sense the world during execution to decide what branch of the plan to follow.
An online planning agent uses execution **monitoring** and splices in repairs as needed to recover from unexpected situations, which can be due to nondeterministic actions, exogenous events, or incorrect models of the environment.
Many actions consume **resources** so planning plus scheduling is needed.

12 Uncertainty

Probabilities express the agent's inability to reach a definite decision regarding the truth of a sentence. **Decision theory** combines the agent's beliefs and desires, defining the best action as the one that maximizes **expected utility**.
Basic probability statements include **prior or unconditional** probabilities and **posterior or conditional** - $P(a|b) = \frac{P(a \wedge b)}{P(b)}$ - probabilities over simple and complex propositions. The axioms of probability constrain the probabilities of logically related propositions. The **full joint probability** distribution specifies the probability of each complete assignment of values to random variables. Absolute independence between subsets of random variables allows the full joint distribution to be factored into smaller joint distributions, greatly reducing its complexity.
Bayes' rule - $P(b|a) = \frac{P(a|b)P(b)}{P(b)}$ - allows unknown probabilities to be computed from known conditional probabilities, usually in the causal direction.
Conditional independence is when an observation is irrelevant or redundant and allows the full joint distribution to be factored into smaller, conditional distributions.

13 Probabilistic Reasoning

Bayes networks provide a natural representation for (causally induced) conditional independence. Topology and Conditional Probability Table (CPT) give compact representation of joint distribution. Generally easy for (non)experts to construct.
Canonical distributions (e.g., noisy-OR) are compact representation of CPTs.
Continuous variables need a parameterized distributions (e.g., linear Gaussian).
Exact inference by variable elimination: carry out summations right-to-left, storing intermediate results (factors) to avoid recomputations. Polytime on polytrees and it's NP-hard on general graphs. The space occupied is very sensitive to topology.
Approximated inference by random sampling techniques can give reasonable estimates of the true posterior probabilities in a network and can cope with much larger networks than can exact algorithms.

14 Probabilistic Reasoning Over Time

The changing state of the world is handled by using a set of random variables to represent the state at each point in **time**.
Representations can be designed to (roughly) satisfy the Markov property, so that

the future is independent of the past given the present. Combined with the assumption that the process is time homogeneous, this greatly simplifies the representation. A **temporal probability model** can be thought of as containing a transition model describing the state evolution and a sensor model describing the observation process. The principal inference tasks in temporal models are:

- filtering: computing the state estimation and updating it;
- prediction: computing the posterior distribution over a future state, given all evidence to date (filtering without the contribution of new evidence);
- smoothing: computing the posterior distribution over a past state, given all evidence up to the present;
- most likely explanation: given a sequence of observations, we might wish to find the sequence of states that is most likely to have generated those observations.

Hidden Markov Models are a family of "simple" markov models with single discrete variables but extensively used in practice to solve real-world problems.

15 Multi Agent Decision Making

Multiagent planning is necessary when there are other agents in the environment with which to **cooperate or compete**.
Game theory describes rational behavior for agents in situations in which multiple agents interact.
Solution concepts in game theory are intended to characterize rational outcomes of a game.
Non-cooperative game theory assumes that agents must make their decisions independently.
Cooperative game theory considers settings in which agents can make binding agreements to form coalitions in order to cooperate.

16 Probabilistic Programming

Probability models define a set of possible worlds with a probability for each world. **Relational probability models** (RPMs) define probability models on worlds derived from the database semantics for first-order languages. RPMs provide very concise models for worlds with large numbers of objects and can handle relational uncertainty.
Open-universe probability models (OUPMs) build on the full semantics of first-order logic, allowing for new kinds of uncertainty such as identity and existence uncertainty.
Probabilistic Programming Languages typically provide universal expressive power for probability models.
Generative programs are representations of probability models, including OUPMs, as executable programs in a probabilistic programming language. A generative program represents a distribution over execution traces of the program.

17 Philosophy, Ethics, Safety and Future of AI

Philosophers use the term **weak AI** for the hypothesis that machines could possibly behave intelligently, and **strong AI** for the hypothesis that such machines would count as having actual minds (as opposed to simulated minds).
AI is a powerful technology, and as such it **poses potential dangers**, through lethal autonomous weapons, security and privacy breaches, unintended side effects, unintentional errors, and malignant misuse. Those who work with AI technology have an **ethical imperative to responsibly reduce those dangers**. AI systems must be able to demonstrate they are **fair**, **trustworthy**, and **transparent**. There are multiple aspects of fairness, and it is impossible to maximize all of them at once. So, a first step is to decide what counts as fair.
Automation is already changing the way people work. As a society, we will have to deal with these **changes**.