Chapter 1

Introduction

Planning in Artificial Intelligence is *decision making* about the *actions* to be taken.

Consider an intelligent robot. The robot is a computational mechanism that takes input through its sensors that allow the robot to *observe* its environment and to build *a representation* of its immediate surroundings and parts of the world it has observed earlier. For a robot to be useful it has to be able to *act*. A robot acts through its *effectors* which are devices that allow the robot to move itself and other objects in its immediate surroundings. A robot resembling a human being has hands and feet, or their muscles, as effectors.

At an abstract level, a robot is a mechanism that maps its observations, which are obtained through the sensors, to actions which are performed by means of the effectors. Planning is the decision making needed in producing a sequence of actions given a sequence of observations. The more complicated the environment and the tasks of the robot are, the more intelligent the robot has to be. For genuine intelligence it is important that the robot is able to plan its actions also in challenging situations.

No intelligent robots exist yet. The most intelligent existing robots carry out tasks that do not require genuine intelligence, like transporting objects from one place to another in environments that are predictable and known in advance. For more challenging tasks in which the working environment of the robot is not exactly known in advance, the biggest challenges are currently in interpreting the sensor data reliably and controlling the basic movements of the robot effectively. Before these research problems have been solved adequately, the employment of robots for more intelligent tasks is not feasible. When this stage will be reached some time in the future, powerful techniques for knowledge representation and task planning will be needed to bring the intelligence of the robots to a sufficiently high level.

Impediments for the success of AI in producing genuinely intelligent beings are related to perceiving and representing knowledge concerning the world. The real world is very complicated in all its physical and geometric as well as social aspects, and representing all the knowledge required by an intelligent being may be too inflexible and complicated by the logical and symbolical means almost exclusively used in artificial intelligence and in planning. This has been criticized by many researchers [Brooks, 1991] and it is a topic of continuing scientific debate.

AI planning – like knowledge representation and learning techniques in AI in general – are currently best applicable in restricted domains in which it is easy to identify what the atomic facts are and to exactly describe how the world behaves. These properties are best fulfilled by systems that are completely man-made, or systems in which planning can view the world at a sufficiently abstract level.

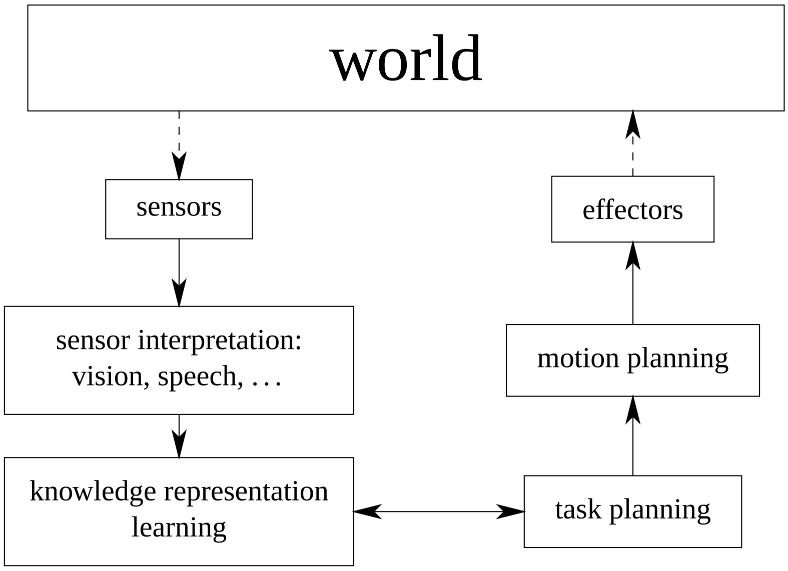

Figure 1.1: Software architecture of an intelligent robot

An example of a completely man-made system to which planning techniques have successfully been applied include the control of autonomous spacecraft [Muscettola *et al.*, 1998]. The vacuum of the outer space is a very simple environment without most of the uncertainties typically present on the surface of the earth. Other current robotic applications like delivering mail in an office or distributing medicine in a hospital, employ only very little from the potential of AI planning.

A simple real-world application in which abstracting away the details of the real world is possible is transportation planning: how to get from Freiburg to London by public transportation, trains, airplanes and buses. If a robot were capable of finding its way between the couple of hundred of meters between the various forms of transportation and recognize the trains and buses to board it could easily travel all over the world. Planning what transportation to use is an easy problem in this case.

1.1 Types of planning problems

The word *planning* is very general and denotes many different things. Even in the AI and robotics context there are many types of planning.

Simply controlling the basic movement of robots is a very challenging problem. *Path planning* is needed in finding a way for one location to another, and *motion planning* is needed in moving the hands and feet of the robot to produce meaningful behavior. and they are not discussed in this lecture as they require specialized representations of the geometric properties of the world and cannot usually be efficiently represented in the general state-based model we are interested in. There is also the well established research area of *scheduling* which is concerned with ordering and choosing a schedule for executing a number of predefined actions.

The topic of this lecture is sometimes called *task planning* in order to distinguish it from the more concrete geometric and physical forms of planning which are used in controlling the movements of robots and similar systems.

Even within task planning, there are many different types of planning problems, depending on the assumptions concerning the properties of actions and the world that are made. Some of these are the following.

1. Determinism versus nondeterminism.

   In the simplest form of planning the state of the world at any moment is unambiguously determined by the initial state of the world and the sequence of actions that have been taken. Hence the world is completely deterministic.

   The assumption of a deterministic world holds in many simple planning problems. However, when the world is modeled in more detail and more realistically, the assumption does not hold any more: the plans have to take into account events that take place independently of the actions and also the possibility that the effects of an action are not the same every time the action is taken, even when the world appears to be the same.

   Nondeterminism comes from two different sources.

   First, any feasible model of the world is very incomplete, and events that are possible as far as our beliefs are concerned can be viewed as nondeterministic: we do not know whether somebody is going to phone or visit us, and the visit or phone call can be modeled as a nondeterministic event that may or may not take place.

   Second, many actions themselves are by their nature nondeterministic, either intentionally or unintentionally. Throwing two dice and summing the result has 11 possible outcomes that cannot be predicted. Throwing an object to a garbage bin from a distance may or may not succeed.

   Notice that there is still the possibility that the physical universe is completely deterministic, but as long as we do not know the exact causes of events, we might just as well consider them nondeterministic.

2. Observability.

   For deterministic planning problems with one initial state the world is completely predictable. As the state of the world after taking certain actions can be completely predicted, there is no need to use observations. Hence a plan, if one exists, is simply a sequence of actions.

   When the actions or the environment can be nondeterministic, or when the initial state is not exactly known, it is not in general possible to reach the goals by using one fixed sequence of actions. The actions have to depend on the observations.

   There are two possibilities. First, planning could be interleaved with plan execution: only one action is chosen at a time, it is executed, and based on the observations the next action is chosen, and so on. Second, a complete plan is generated, covering all possible events that can happen, and it is executed, without further planning during execution. This kind of plans could be formalized as programs with conditionals (*if-then-else*) and loops.

   These two approaches are computationally very close, but the first approach does not require explicitly representing all the action sequences that might be needed, it only has to find a guarantee that such action sequences exist.

The possible observations have a strong impact on how exactly the actual state of the world can be determined: the more facts can be observed, the more precisely the current state of the world can be determined, and the better the most appropriate action can be chosen. If there is a lot of uncertainty concerning the current state of the world it may be impossible to choose an appropriate action.

If the current state can always be determined uniquely we have *full observability*. If the current state cannot be determined uniquely we have *partial observability*, and planning algorithms are forced to consider sets of possible current states.

3. Time.

   Most work on planning uses discrete (integer) time and actions of unit duration. This means that all changes caused by an action at time point t are visible at time point $t+1$. So changes in the world take only one unit of time, and what happens between two time points is not analyzed further.

   More complicated models of time and change are possible, but in this lecture we consider only discrete time. Most types of problems can be analyzed in terms of discrete time by making the unit duration sufficiently small. Rational and real time cause conceptual difficulties. Effects of actions that are not immediate can be reduced to the basic case by encoding the delayed effects in the state description.

4. Control information and plan structure.

   In the basic planning problem a plan is to be synthesized based on a generic description of how the actions affect the world.

   There may be, however, further control information that may affect the planning process and the plans that are produced. In hierarchical planning, for example, information on the structure of the possible plans is given in the form of a hierarchical task network, and the plans that are produced must conform to this structure. This kind of structural information may substantially improve the efficiency of planning. Another way of restricting the structure of plans, for efficiency or other reasons, is by using temporal logics [Bacchus and Kabanza, 2000].

5. Plan quality.

   The purpose of a plan is often just to reach one of the predefined goal states, and plans are judged only with respect to the satisfaction of this property. However, actions may have differing costs and durations, and plans could be assessed in terms of their time consumption or cost.

   As different executions of a plan in a nondeterministic world produce different sequences of actions, plans can be valued in terms of their expected costs, best-case costs, worst-case costs, and probability of eventually reaching the goals.

   Plans with an infinite execution length can also be considered, and then plans may be valued according to their average cost per unit time, or according to their geometrically discounted costs.

1.2 Related topics

Reasoning about action has emerged as a separate research area with the goal of making inferences about actions and their effects [Ginsberg and Smith, 1988; Shoham, 1988; Sandewall, 1994a; 1994b; Stein and Morgenstern, 1994]. Important research topics include the qualification and the ramification problems, which respectively involve deciding whether a certain action can be performed to have its anticipated effects and what are the indirect effects of an action. These problems are important because of their relation to the reasoning performed by human beings and their importance in representing the world as required by intelligent systems employing planning. In this lecture, however, we assume that a description of some actions is given, with all preconditions and direct and indirect effects fully spelled out, and concentrate on what kind of planning can be performed with these actions. The problems are also fully orthogonal, that is, the planning algorithms do not need to depend on the solution to the ramification and qualification problems that are used.

Markov decision processes [Puterman, 1994] in operations research is essentially a formalization of planning. In contrast to AI planning, work in that area has used explicit enumerative representations of transition systems, like those used in Section 2.1, and as a consequence the algorithms have a different flavor than most planning algorithms do. However, most recent work on probabilistic planning is based on Markov decision processes.

Discrete event systems (DES) in control engineering have been proposed as a model for synthesizing controllers for systems like automated factories [Ramadge and Wonham, 1987; Wonham, 1988], and this topic is closely related to planning. Again, there are differences in the problem formulation, with state spaces being represented enumeratively or more succinctly, for example as Petri nets [Ichikawa and Hiraishi, 1988] or vector additions systems [Li and Wonham, 1993].

Synthesis of programs for reactive systems that work in nondeterministic and partially observable environments is similar to planning under same conditions. Program synthesis has been considered for example from specifications of their input-output behavior in different types of temporal logics [Vardi and Stockmeyer, 1985; Kupferman and Vardi, 1999].

1.3 Early research on AI planning

Research that has lead to current AI planning started in the 1960's in the form of programs that tried to simulate problem solving abilities of human beings. One of the first programs of this kind was the General Problem Solver (GPS) by Newell and Simon [Ernst *et al.*, 1969]. GPS performed state space search guided by estimated differences between the current state and the goal states.

At the end of 1960's Green proposed the use of theorem-provers for constructing plans [Green, 1969]. However, because of the immaturity of theorem-proving techniques at that time, this approach was soon mostly abandoned in favor of specialized planning algorithms. There was theoretically oriented work on deductive planning which used different kinds of modal and dynamic logics [Rosenschein, 1981] but these works had little impact on the development of efficient planning algorithms. Deductive and logic-based approaches to planning gained popularity again only at the end of the 1990's as a consequence of the development of more sophisticated programs for the satisfiability problem of the classical propositional logic [Kautz and Selman, 1996].

One of the most well known early planning systems is the STRIPS planner from the beginning of the 1970's [Fikes and Nilsson, 1971]. The states in STRIPS are sets of formulae, and the operators change these state descriptions by adding and deleting formulae in the sets. Heuristics similar to the ones used in the GPS system were used in guiding the search. The definition of operators, with a *precondition* as well as *add* and *delete* lists, corresponding to the facts that respectively become true and false, and the associated terminology, is still in common use, although restricted to atomic facts, that is, the add list is simply the set of state variables that the action makes true, and the delete list similarly consists of the state variables that become false.

Starting in the mid 1970's the dominating approach to domain-independent planning was the so-called partial-order, or causal link, or nonlinear planning [Sacerdoti, 1975; McAllester and Rosenblitt, 1991], which remained popular until the mid-1990's and the introduction of the Graphplan planner [Blum and Furst, 1997] which started the shift away from partial-order planning to types of algorithms that had earlier been considered infeasible, even the then-notorious total-order planners. The basic idea of partial-order planning is that a plan is incrementally constructed starting from the initial state and the goals, by either adding an action to the plan so that one of the open goals or operator preconditions is fulfilled, or adding an ordering constraint on operators already in the plan in order to resolve a potential conflict between them. In contrast to the forward or backward search strategies in Chapter 3 partial-order planners tried to avoid unnecessarily imposing an ordering on operators. The main advantages of both partial-order planners and Graphplan are present in the SAT/CSP approach to planning which is discussed in Section 3.6.

In parallel to partial-order planning, the notion of hierarchical planning emerged [Sacerdoti, 1974], and it has been deployed in many real-world applications. The idea in hierarchical planning is that the problem description imposes a structure on solutions and restricts the number of choices the planning algorithm has to make. A hierarchical plan consists of a main task which is decomposed to smaller tasks which are recursively solved. For each task there is a choice between solution methods. The less choice there is, the more efficiently the problem is solved. Furthermore, many hierarchical planners allow the embedding of problem-specific heuristics and problem-solvers to further speed up planning.

A collection of articles on AI planning starting from the late 1960's has been edited by Allen et al. [1990]. Many of the papers are mainly of historical interest, and some of them outline ideas that are still in use.

1.4 This book

My intention in writing these lecture notes was to cover planning problems of different generality and some of the most important approaches to solving each type of problem. Of course, during the last several decades of planning research a lot of work has been done that are not covered in these notes.

Important differences to most textbooks and research papers on planning is that I use a unified and rather expressive syntax for representing operators, including nondeterministic and conditional effects. This has several implications on the material covered in this book. For example, it may surprising that I do not use a concept viewed very central for deterministic planning by some researchers, *the planning graphs* of Blum and Furst [1997]. This is a direct implication of the general syntax for operators I use, as discussed in more detail in Section 3.8. It seems that any useful graph-theoretic properties planning graphs have lose their meaning when a definition of operators more general than STRIPS operators is used.

One of the messages of these notes is the importance of logic (propositional logic in our case) for all forms of planning ranging from the simplest deterministic case to the most general types of planning with partial observability. As we will see, states, sets of states, belief states and transition relations associated with operators are often most naturally represented as propositional formulae. This representation shows up once and again in connection with different types of planning algorithms, including backward search in classical/deterministic planning, planning as satisfiability, and in implementations of nondeterministic planning algorithms by means of binary decision diagrams and similar data structures.

In addition to generalizing many existing techniques to the more general definition of planning problems, many of the algorithms are either new or have been developed further from earlier algorithms. I cite the original sources in the literature sections in the end of every chapter. Some of my contributions can be singled out rather precisely. They include the following.

1. The definition of regression for conditional and nondeterministic operators in Sections 3.1.2 and 4.1.1.

2. The algorithm for computing invariants in Section 3.5. The computation of mutexes in Blum and Furst's [1997] planning graphs can be viewed as a special case of my algorithm, restricted to unconditional operators only.

3. The algorithm for planning with full observability in Section 4.4.2. This algorithm is based on a similar but more complicated algorithm by Cimatti et al. [2003].

4. The representation of planning without observability as quantified Boolean formulae in Section 4.6.

5. The framework for non-probabilistic planning with partial observability in Section 4.7.

6. The complexity results in Section 4.8.3, most importantly the 2-EXP-completeness result for conditional planning with partial observability.