

## **Automated Planning**

- **Introduction and Overview**

## Literature

- Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning—Theory and Practice*, chapter 1. Elsevier/Morgan Kaufmann, 2004.
- John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation*, chapter 2. Addison Wesley, 1979.
- Qiang Yang. *Intelligent Planning—A Decomposition and Abstraction Based Approach*. Springer, 1997.
- James Allen, James Hendler, Austin Tate (eds). *Readings in Planning*. Morgan Kaufmann, 1990.

Automated Planning: Introduction and Overview 2

## Literature

•main course book:

•**Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning—Theory and Practice*, chapter 1. Elsevier/Morgan Kaufmann, 2004.**

•for this lecture (finite state systems):

•**John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation*, chapter 2. Addison Wesley, 1979.**

•additional books on AI planning:

•**Qiang Yang. *Intelligent Planning—A Decomposition and Abstraction Based Approach*. Springer, 1997.**

•**James Allen, James Hendler, Austin Tate (eds). *Readings in Planning*. Morgan Kaufmann, 1990.**

Overview

- ▶ **What is AI Planning?**
  - A Conceptual Model for Planning
  - Restricting Assumptions
  - A Running Example: Dock-Worker Robots

Automated Planning: Introduction and Overview 3

## Overview

### •What is AI Planning?

•now: what do we mean by (AI) planning?

### •A Conceptual Model for Planning

### •Restricting Assumptions

### •A Running Example: Dock-Worker Robots

**Human Planning and Acting**

- acting without (explicit) planning:
  - when purpose is immediate
  - when performing well-trained behaviours
  - when course of action can be freely adapted
- acting after planning:
  - when addressing a new situation
  - when tasks are complex
  - when the environment imposes high risk/cost
  - when collaborating with others

• people plan only when strictly necessary

Automated Planning: Introduction and Overview 4

## Human Planning and Acting

- humans rarely plan before acting in everyday situations
- **acting without (explicit) planning:** (may be subconscious)
  - **when purpose is immediate** (e.g. switch on computer)
  - **when performing well-trained behaviours** (e.g. drive car)
  - **when course of action can be freely adapted** (e.g. shopping)
- **acting after planning:**
  - **when addressing a new situation** (e.g. move house)
  - **when tasks are complex** (e.g. plan this course)
  - **when the environment imposes high risk/cost** (e.g. manage nuclear power station)
  - **when collaborating with others** (e.g. build house)
- **people plan only when strictly necessary**
  - because planning is complicated and time-consuming (trade-off: cost vs. benefit)
  - often we seek only good rather than optimal plans

Defining AI Planning

- planning:
  - explicit deliberation process that chooses and organizes actions by anticipating their outcomes
  - aims at achieving some pre-stated objectives
- AI planning:
  - computational study of this deliberation process

Automated Planning: Introduction and Overview 5

## Defining AI Planning

### •planning:

•**explicit deliberation process that chooses and organizes actions by anticipating their outcomes**

•in short: planning is reasoning about actions

•**aims at achieving some pre-stated objectives**

•or: achieving them as best as possible (planning as optimization problem)

### •AI planning:

•**computational study of this deliberation process**

### Why Study Planning in AI?

- scientific goal of AI:  
understand intelligence
  - planning is an important component of rational (intelligent) behaviour
- engineering goal of AI:  
build intelligent entities
  - build planning software for choosing and organizing actions for autonomous intelligent machines



Automated Planning: Introduction and Overview 6

## Why Study Planning in AI?

- **scientific goal of AI: understand intelligence**
  - **planning is an important component of rational (intelligent) behaviour**
  - planning is part of intelligent behaviour
- **engineering goal of AI: build intelligent entities**
  - **build planning software for choosing and organizing actions for autonomous intelligent machines**
  - example: Mars explorer (cannot be remotely operated)
  - robot: Shakey, SRI 1968

**Domain-Specific vs. Domain-Independent Planning**

- domain-specific planning: use specific representations and techniques adapted to each problem
  - important domains: path and motion planning, perception planning, manipulation planning, communication planning
- domain-independent planning: use generic representations and techniques
  - exploit commonalities to all forms of planning
  - leads to general understanding of planning

• domain-independent planning complements domain-specific planning

Automated Planning: Introduction and Overview 7

## **Domain-Specific vs. Domain-Independent Planning**

**•domain-specific planning: use specific representations and techniques adapted to each problem**

**•important domains: path and motion planning, perception planning, manipulation planning, communication planning**

**•domain-independent planning: use generic representations and techniques**

**•exploit commonalities to all forms of planning**

•saves effort; no need to reinvent same techniques for different problems

**•leads to general understanding of planning**

•contributes to scientific goal of AI

**•domain-independent planning complements domain-specific planning**

•use domain-independent planning where highly efficient solution is required

Overview

- What is AI Planning?
- **A Conceptual Model for Planning**
- Restricting Assumptions
- A Running Example: Dock-Worker Robots

Automated Planning: Introduction and Overview 8

## Overview

### •What is AI Planning?

- just done: what do we mean by (AI) planning?

### •A Conceptual Model for Planning

- now: state-transition systems – formalizing the problem

### •Restricting Assumptions

### •A Running Example: Dock-Worker Robots

## Why a Conceptual Model?

- **conceptual model**: theoretical device for describing the elements of a problem
- good for:
  - explaining basic concepts
  - clarifying assumptions
  - analyzing requirements
  - proving semantic properties
- not good for:
  - efficient algorithms and computational concerns



Automated Planning: Introduction and Overview 9

## Why a Conceptual Model?

• **conceptual model**: theoretical device for describing the elements of a problem

• good for:

• **explaining basic concepts**: what are the objects to be manipulated during problem-solving?

• **clarifying assumptions**: what are the constraints imposed by this model?

• **analyzing requirements**: what representations do we need for the objects?

• **proving semantic properties**: when is an algorithm sound or complete?

• not good for:

• **efficient algorithms and computational concerns**

• graph: Cyc upper ontology

## Conceptual Model for Planning: State-Transition Systems

- A state-transition system is a 4-tuple  $\Sigma = (S, A, E, \gamma)$ , where:
  - $S = \{s_1, s_2, \dots\}$  is a finite or recursively enumerable set of states;
  - $A = \{a_1, a_2, \dots\}$  is a finite or recursively enumerable set of actions;
  - $E = \{e_1, e_2, \dots\}$  is a finite or recursively enumerable set of events; and
  - $\gamma: S \times (A \cup E) \rightarrow 2^S$  is a state transition function.
- if  $a \in A$  and  $\gamma(s, a) \neq \emptyset$  then  $a$  is applicable in  $s$
- applying  $a$  in  $s$  will take the system to  $s' \in \gamma(s, a)$

Automated Planning: Introduction and Overview

10

## Conceptual Model for Planning: State-Transition Systems

### • A state-transition system is a 4-tuple $\Sigma = (S, A, E, \gamma)$ , where:

- a general model for a dynamic system, common to other areas of computer science; aka. dynamic-event system

- **$S = \{s_1, s_2, \dots\}$  is a finite or recursively enumerable set of states;**

  - the possible states the world can be in

- **$A = \{a_1, a_2, \dots\}$  is a finite or recursively enumerable set of actions;**

  - the actions that can be performed by some agent in the world, transitions are controlled by the plan executor

- **$E = \{e_1, e_2, \dots\}$  is a finite or recursively enumerable set of events; and**

  - the events that can occur in the world, transitions that are contingent (correspond to the internal dynamics of the system)

- **$\gamma: S \times (A \cup E) \rightarrow 2^S$  is a state transition function.**

  - notation:  $2^S$  = powerset of  $S$ ; maps to a set of states

  - the function describing how the world evolves when actions or events occur

  - note: model does not allow for parallelism between actions and/or events

- if  $a \in A$  and  $\gamma(s, a) \neq \emptyset$  then  $a$  is applicable in  $s$

- applying  $a$  in  $s$  will take the system to  $s' \in \gamma(s, a)$

## State-Transition Systems as Graphs

- A state-transition system  $\Sigma = (S, A, E, \gamma)$  can be represented by a directed labelled graph  $G = (N_G, E_G)$  where:
  - the nodes correspond to the states in  $S$ , i.e.  $N_G = S$ ; and
  - there is an arc from  $s \in N_G$  to  $s' \in N_G$ , i.e.  $s \rightarrow s' \in E_G$ , with label  $u \in (A \cup E)$  if and only if  $s' \in \gamma(s, u)$ .

Automated Planning: Introduction and Overview

11

## State-Transition Systems as Graphs

• A state-transition system  $\Sigma = (S, A, E, \gamma)$  can be represented by a directed labelled graph  $G = (N_G, E_G)$  where:

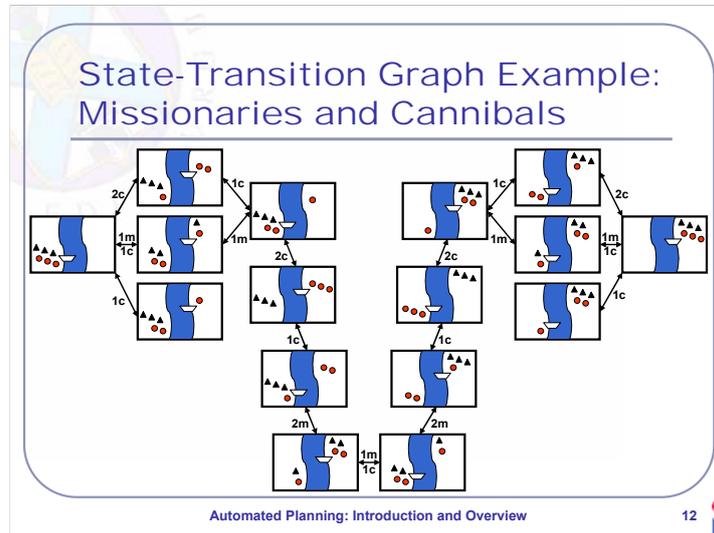
• the nodes correspond to the states in  $S$ , i.e.  $N_G = S$ ;  
and

• nodes correspond to world states

• there is an arc from  $s \in N_G$  to  $s' \in N_G$ , i.e.  $s \rightarrow s' \in E_G$ , with label  $a \in (A \cup E)$  if and only if  $s' \in \gamma(s, a)$ .

• there is an arc if there is an action or event that transforms one state into the other (called a state transition)

• the label of that arc is that action or event



## State-Transition Graph Example: Missionaries and Cannibals

- On one bank of a river are three missionaries (black triangles) and three cannibals (red circles). There is one boat available that can hold up to two people and that they would like to use to cross the river. If the cannibals ever outnumber the missionaries on either of the river's banks, the missionaries will get eaten. How can the boat be used to safely carry all the missionaries and cannibals across the river?

- see <http://www.aiai.ed.ac.uk/~gwickler/missionaries.html>

**Objectives and Plans**

- **state-transition system:**
  - describes all ways in which a system may evolve
- **plan:**
  - a structure that gives appropriate actions to apply in order to achieve some objective when starting from a given state
- **types of objective:**
  - **goal state**  $s_g$  or set of goal states  $S_g$
  - satisfy some conditions over the sequence of states
  - optimize utility function attached to states
  - task to be performed

Automated Planning: Introduction and Overview 13

## Objectives and Plans

### •state-transition system:

- describes all ways in which a system may evolve

### •plan:

- a structure that gives appropriate actions to apply in order to achieve some objective when starting from a given state

- structure: e.g. sequential list of actions to be performed in order; function mapping states to actions

- describes a path through the state-transition graph

### •types of objective:

#### •goal state $s_g$ or set of goal states $S_g$

- simplest case; objective achieved by sequence of transitions that ends in one of the goal states

#### •satisfy some conditions over the sequence of states

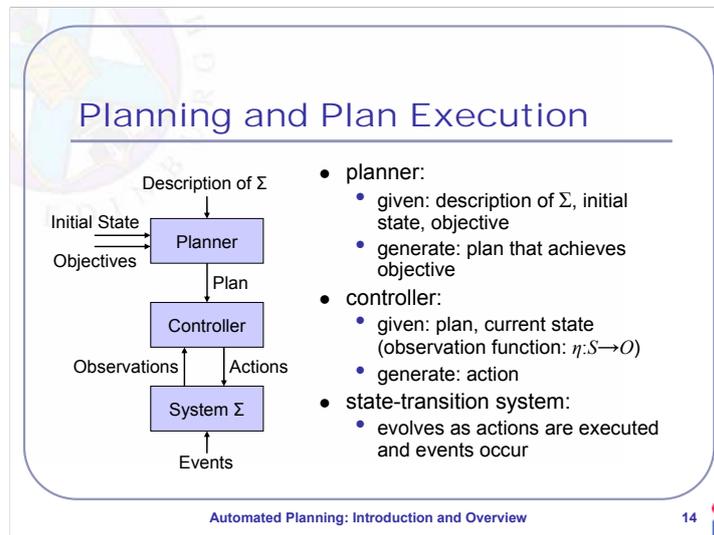
- example: states to be avoided or visited (goal state is special case)

#### •optimize utility function attached to states

- optimize compound function (sum, max) of utilities of visited states

#### •task to be performed

- alternative approach: recursively defined set of actions and (other) tasks



## Planning and Plan Execution

### •planner:

- **given: description of  $\Sigma$ , initial state, objective**

- **generate: plan that achieves objective**

- planner works offline: relies on formal description of  $\Sigma$

### •controller:

- **given: plan, current state (observation function:  $\eta: S \rightarrow O$ )**

- partial knowledge of controller about world modelled through observation function

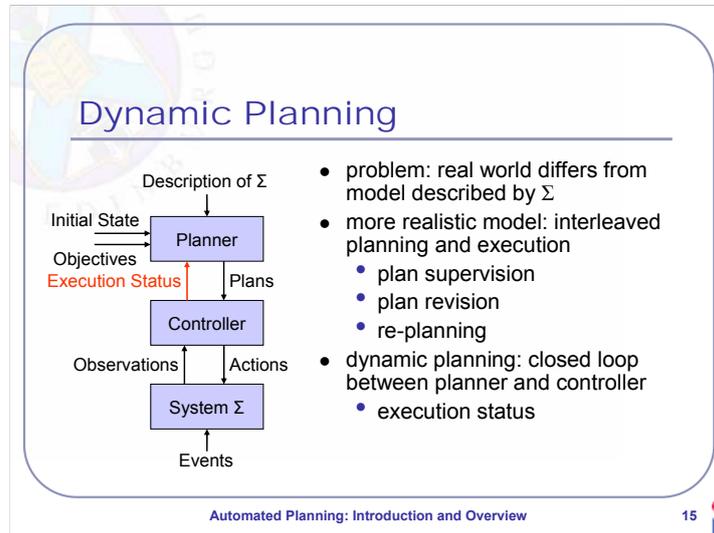
- $O$  = set of possible observations (e.g. subset); input for controller

- **generate: action**

- controller works online: along with the dynamics of  $\Sigma$

### •state-transition system:

- **evolves as actions are executed and events occur**



## Dynamic Planning

• **problem: physical system differs from model described by  $\Sigma$**

- planner only has access to model (description of  $\Sigma$ )
- controller must cope with differences between  $\Sigma$  and real world

• **more realistic model: interleaved planning and execution**

- **plan supervision:** detect when observations differ from expected results
- **plan revision:** adapt existing plan to new circumstances
- **re-planning:** generate a new plan from current (initial) state

• **dynamic planning: closed loop between planner and controller**

- **execution status**

Overview

- What is AI Planning?
- A Conceptual Model for Planning
- **Restricting Assumptions**
- A Running Example: Dock-Worker Robots

Automated Planning: Introduction and Overview 16

## Overview

### •What is AI Planning?

### •A Conceptual Model for Planning

- just done: state-transition systems – formalizing the problem

### •Restricting Assumptions

- now: assumptions behind the model and what if we relax them

### •A Running Example: Dock-Worker Robots

A0: Finite  $\Sigma$

- Assumption A0
  - system  $\Sigma$  has a finite set of states
- Relaxing A0
  - why?
    - to describe actions that construct or bring new objects into the world
    - to handle numerical state variables
  - issues:
    - decidability and termination of planners

Automated Planning: Introduction and Overview 17

## A0: Finite $\Sigma$

### •Assumption A0

#### •system $\Sigma$ has a finite set of states

- definition of STS requires  $S$  to be finite or recursively enumerable
- graph will be finite

### •Relaxing A0

#### •why?

#### •to describe actions that construct or bring new objects into the world

- example: building a car

#### •to handle numerical state variables

- example: height, weight, etc. of objects

#### •issues:

#### •decidability and termination of planners

- states in FOPL: reasoning within one state is undecidable

A1: Fully Observable  $\Sigma$

- Assumption A1
  - system  $\Sigma$  is fully observable, i.e.  $\eta$  is the identity function
- Relaxing A1
  - why?
    - to handle states in which not every aspect is or can be known
  - issues:
    - if  $\eta(s)=o$ ,  $\eta^{-1}(o)$  usually more than one state (ambiguity)
    - determining the successor state

Automated Planning: Introduction and Overview 18

## A1: Fully Observable $\Sigma$

### •Assumption A1

•**system  $\Sigma$  is fully observable, i.e.  $\eta$  is the identity function**

•planner and controller have complete knowledge of the state of the world

### •Relaxing A1

•**why?**

•**to handle states in which not every aspect is or can be known**

•example: route planning with traffic jams

•**issues:**

•**if  $\eta(s)=o$ ,  $\eta^{-1}(o)$  usually more than one state (ambiguity)**

•observations do not fully disambiguate current state

•**determining the successor state**

•state transition function only defined for individual states

A2: Deterministic  $\Sigma$

- Assumption A2
  - system  $\Sigma$  is deterministic, i.e. for all  $s \in S, u \in A \cup E$ :  $|\gamma(s,u)| \leq 1$
  - short form:  $\gamma(s,u)=s'$  for  $\gamma(s,u)=\{s'\}$
- Relaxing A2
  - why?
    - to plan with actions that may have multiple alternative outcomes
  - issues:
    - controller has to observe actual outcomes of actions
    - solution plan may include conditional and iterative constructs

Automated Planning: Introduction and Overview 19

## A2: Deterministic $\Sigma$

### • Assumption A2

• **system  $\Sigma$  is deterministic, i.e. for all  $s \in S, u \in A \cup E$ :  $|\gamma(s,u)| \leq 1$**

• if there is an applicable action it changes the deterministic STS to a single state; similarly for events

• **short form:  $\gamma(s,u)=s'$  for  $\gamma(s,u)=\{s'\}$**

### • Relaxing A2

#### • why?

• **to plan with actions that may have multiple alternative outcomes**

• example: tossing a coin

#### • issues:

• **controller has to observe actual outcomes of actions**

• **solution plan may include conditional and iterative constructs**

• to deal with alternative outcomes

• note: plan consists not only of actions but also control constructs

A3: Static  $\Sigma$

- Assumption A3
  - system  $\Sigma$  is static, i.e.  $E=\emptyset$
  - short form:  $\Sigma = (S,A,\gamma)$  for  $\Sigma = (S,A,\emptyset,\gamma)$
- Relaxing A3
  - why?
    - to model a world in which events can occur
  - issues:
    - world becomes nondeterministic from the point of view of the planner (same issues)

Automated Planning: Introduction and Overview 20

## A3: Static $\Sigma$

### •Assumption A3

#### •system $\Sigma$ is static, i.e. $E=\{\}$

- no events occur; STS remains in state until the controller applies an action

#### •short form: $\Sigma=(S,A,\gamma)$ for $\Sigma=(S,A,\{\},\gamma)$

### •Relaxing A3

#### •why?

- to model a world in which events can occur

- example: planning with weather

#### •issues:

- world becomes nondeterministic from the point of view of the planner (same issues)

- note: some simple cases can be mapped to a deterministic planning problem

**A4: Restricted Goals**

- Assumption A4
  - the planner handles only restricted goals that are given as an explicit goal state  $s_g$  or set of goal states  $S_g$
- Relaxing A4
  - why?
    - to handle constraints on states and plans, utility functions, or tasks
  - issues:
    - representation and reasoning over constraints, utility, and tasks

Automated Planning: Introduction and Overview 21

## A4: Restricted Goals

### •Assumption A4

•the planner handles only restricted goals that are given as an explicit goal state  $s_g$  or set of goal states  $S_g$

•objective: find any sequence of state transactions that ends in a goal state

### •Relaxing A4

#### •why?

•to handle constraints on states and plans, utility functions, or tasks

•examples: drive to work via a newsagent using as little fuel as possible

#### •issues:

•representation and reasoning over constraints, utility, and tasks

•STS does not have the means to express any of these

**A5: Sequential Plans**

- Assumption A5
  - a solution plan is a linearly ordered finite sequence of actions
- Relaxing A5
  - why?
    - to handle dynamic systems (see A3: static  $\Sigma$ )
    - to create different types of plans
  - issues:
    - must not shift problem to the controller
    - reasoning about (more complex) data structures

Automated Planning: Introduction and Overview 22

## **A5: Sequential Plans**

### **•Assumption A5**

**•a solution plan is a linearly ordered finite sequence of actions**

### **•Relaxing A5**

#### **•why?**

**•to handle dynamic systems (see A3: static  $\Sigma$ )**

**•to create different types of plans**

**•examples: conditional plans, partially ordered plans, universal plans (maps states to actions)**

#### **•issues:**

**•must not shift problem to the controller**

**•reasoning about (more complex) data structures**

**•example: determine what is true at a given point in the plan**

**A6: Implicit Time**

- Assumption A6
  - actions and events have no duration in state transition systems
- Relaxing A6
  - why?
    - to handle action duration, concurrency, and deadlines
  - issues:
    - representation of and reasoning about time
    - controller must wait for effects of actions to occur

Automated Planning: Introduction and Overview 23

## **A6: Implicit Time**

### **•Assumption A6**

**•actions and events have no duration in state transition systems**

•state transitions are instantaneous, no explicit representation of time

### **•Relaxing A6**

**•why?**

**•to handle action duration, concurrency, and deadlines**

•example: time-tabling problems (airport)

**•issues:**

**•representation of and reasoning about time**

•plan must include temporal information (see A5: Sequential Plans)

**•controller must wait for effects of actions to occur**

•increased complexity

A7: Offline Planning

- Assumption A7
  - planner is not concerned with changes of  $\Sigma$  while it is planning
- Relaxing A7
  - why?
    - to drive a system towards some objectives
  - issues:
    - check whether the current plan remains valid
    - if needed, revise current plan or re-plan

Automated Planning: Introduction and Overview 24

## A7: Offline Planning

### •Assumption A7

- planner is not concerned with changes of  $\Sigma$  while it is planning**

- plan for given initial state

### •Relaxing A7

- why?

- to drive a system towards some objectives**

- example: plan an investment strategy

- issues:

- check whether the current plan remains valid**

- if needed, revise current plan or re-plan**

- note: need to plan at the right level of abstraction

### The Restricted Model

- restricted model: make assumptions A0-A7
- Given a planning problem  $\mathcal{P}=(\Sigma, s_i, S_g)$  where
  - $\Sigma = (S, A, \gamma)$  is a state transition system,
  - $s_i \in S$  is the initial state, and
  - $S_g \subset S$  is a set of goal states,
- find a sequence of actions  $\langle a_1, a_2, \dots, a_k \rangle$ 
  - corresponding to a sequence of state transitions  $\langle s_i, s_1, \dots, s_k \rangle$  such that
  - $s_1 = \gamma(s_i, a_1), s_2 = \gamma(s_1, a_2), \dots, s_k = \gamma(s_{k-1}, a_k)$ , and  $s_k \in S_g$ .

Automated Planning: Introduction and Overview 25

## The Restricted Model

- restricted model: make assumptions A0-A7
- Given a planning problem  $\mathcal{P}=(\Sigma, s_i, S_g)$  where
  - $\Sigma=(S, A, \gamma)$  is a state transition system,
  - $s_i \in S$  is the initial state, and
  - $S_g \subset S$  is a set of goal states,
- find a sequence of actions  $\langle a_1, a_2, \dots, a_k \rangle$ 
  - corresponding to a sequence of state transitions  $\langle s_i, s_1, \dots, s_k \rangle$  such that
  - $s_1 = \gamma(s_i, a_1), s_2 = \gamma(s_1, a_2), \dots, s_k = \gamma(s_{k-1}, a_k)$ , and  $s_k \in S_g$ .
- full observability (A0) only required for initial state; deterministic model allows for all other states to be predicted with certainty
- plan is unconditional: no branching (if ... then ... else ...)

### Restrictedness?

- non-deterministic state-transition system:

- equivalent deterministic state-transition system:

Automated Planning: Introduction and Overview 26

## Restrictedness?

- **non-deterministic state-transition system: graph**

- non-deterministic, e.g. applying  $a_1$  in  $s_i$  results in  $s_1$  or  $s_2$

- **equivalent deterministic state-transition system: graph**

- each state may contain a set of states from the non-deterministic state-transition system

- it can be shown: for every non-deterministic STS there is an equivalent deterministic STS (possibly exponentially larger)

Overview

- What is AI Planning?
- A Conceptual Model for Planning
- Restricting Assumptions
- **A Running Example: Dock-Worker Robots**

Automated Planning: Introduction and Overview 27

## Overview

### •What is AI Planning?

### •A Conceptual Model for Planning

### •Restricting Assumptions

- just done: assumptions behind the model and what if we relax them

### •A Running Example: Dock-Worker Robots

- now: nontrivial running example used to illustrate ideas

## The Dock-Worker Robots (DWR) Domain

- aim: have one example to illustrate planning procedures and techniques
- informal description:
  - harbour with several locations (docks), docked ships, storage areas for containers, and parking areas for trucks and trains
  - cranes to load and unload ships etc., and robot carts to move containers around



Automated Planning: Introduction and Overview 28

## The Dock-Worker Robots (DWR) Domain

•aim: have one example to illustrate planning procedures and techniques

•problem must be nontrivial to be interesting, but not too much overhead to introduce problem

•informal description:

•generalization of earlier example (state transition graph)

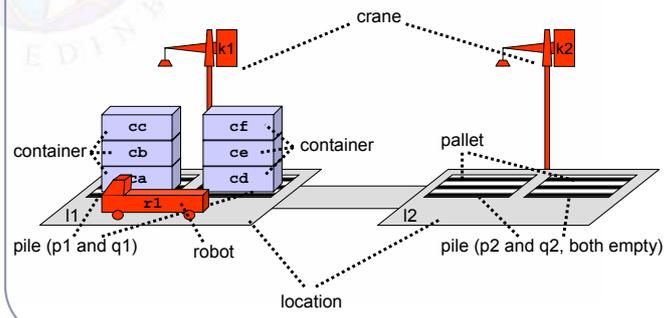
•harbour with several locations (docks), docked ships, storage areas for containers, and parking areas for trucks and trains

•cranes to load and unload ships etc., and robot carts to move containers around

•simplified and enriched version of this domain will be introduced later

•approach: use first-order predicate logic as representation

## DWR Example State



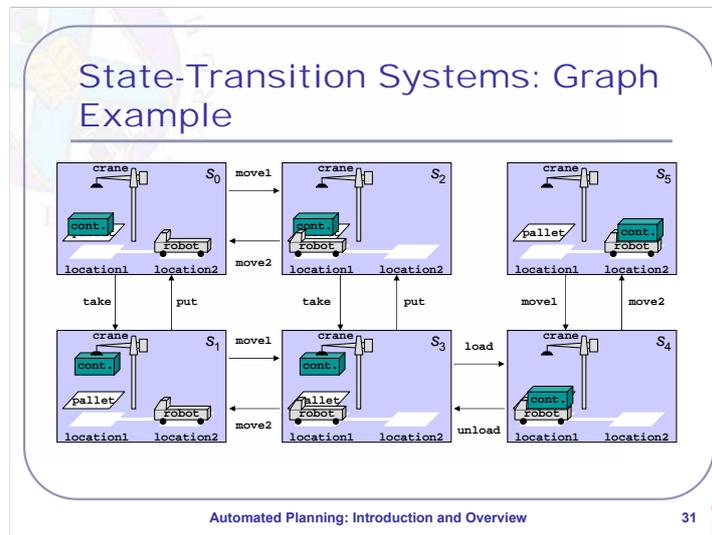
### Actions in the DWR Domain

- **move** robot  $r$  from location  $l$  to some adjacent and unoccupied location  $l'$
- **take** container  $c$  with empty crane  $k$  from the top of pile  $p$ , all located at the same location  $l$
- **put** down container  $c$  held by crane  $k$  on top of pile  $p$ , all located at location  $l$
- **load** container  $c$  held by crane  $k$  onto unloaded robot  $r$ , all located at location  $l$
- **unload** container  $c$  with empty crane  $k$  from loaded robot  $r$ , all located at location  $l$

Automated Planning: Introduction and Overview 30

## Actions in the DWR Domain

- **move robot  $r$  from location  $l$  to some adjacent and unoccupied location  $l'$**
- **take container  $c$  with empty crane  $k$  from the top of pile  $p$ , all located at the same location  $l$**
- **put down container  $c$  held by crane  $k$  on top of pile  $p$ , all located at location  $l$**
- **load container  $c$  held by crane  $k$  onto unloaded robot  $r$ , all located at location  $l$**
- **unload container  $c$  with empty crane  $k$  from loaded robot  $r$ , all located at location  $l$**
- formal specifications will follow when we have introduced a formal action description language
- problem: how to represent actions formally? first-order logic?



## State-Transition Systems: Graph Example

- states:  $s_0$  to  $s_5$ 
  - objects: robot, crane, container, pallet, two locations
- actions:
  - crane can take/put the container from the pallet/onto the pallet
  - crane can load/unload the container from the robot
  - robot can drive to either location (with or without the container loaded)
- no events
- state transition function: arcs shown in graph
- note: state transitions are deterministic: each action leads to at most one other state

Overview

- What is AI Planning?
- A Conceptual Model for Planning
- Restricting Assumptions
- A Running Example: Dock-Worker Robots

Automated Planning: Introduction and Overview 32

## Overview

### •What is AI Planning?

### •A Conceptual Model for Planning

### •Restricting Assumptions

### •A Running Example: Dock-Worker Robots

- just done: nontrivial running example used to illustrate ideas