OPPA European Social Fund
Prague & EU: We invest in your future.

Plánování a hry - Automated planning and game playing

Michal Pěchouček

Katedra kybernetiky,
České vysoké učení technické v Praze

February 14, 2010

Intro & admin

- **Instructor:**
  - Michal Pechoucek, `pechoucek@fel.cvut.cz`, 7355, K120
  - Carmel Domshlak , `http://iew3.technion.ac.il/ dcarmel/`
- **Teaching assistants:**
  - Stepan Kopriva, Jiri Vokrinek, Lukas Chrpa and Martin Grill (all ATG)

Intro & admin

- **Instructor:**
  - Michal Pechoucek, `pechoucek@fel.cvut.cz`, 7355, K120
  - Carmel Domshlak , `http://iew3.technion.ac.il/ dcarmel/`
- **Teaching assistants:**
  - Stepan Kopriva, Jiri Vokrinek, Lukas Chrpa and Martin Grill (all ATG)
- **Web support** is on OI courseware:
  `http://cw.felk.cvut.cz/doku.php/courses/a4m33pah/start`

Intro & admin

- **Instructor:**
  - Michal Pechoucek, `pechoucek@fel.cvut.cz`, 7355, K120
  - Carmel Domshlak , `http://iew3.technion.ac.il/ dcarmel/`
- **Teaching assistants:**
  - Stepan Kopriva, Jiri Vokrinek, Lukas Chrpa and Martin Grill (all ATG)
- **Web support** is on OI courseware:
  `http://cw.felk.cvut.cz/doku.php/courses/a4m33pah/start`
- **Requirements:**
  1. project - explain the rules of the game 30%
  2. test - 70%

Content of the lecture ONE

1. Components
2. Mode of the lecture
3. Motivation
4. Preliminaries

Content of the lecture ONE

1. **Components**
   - Foundation of automated planning
   - Game playing (adversarial planning) .. to be continued in MAS
2. Mode of the lecture
3. Motivation
4. Preliminaries

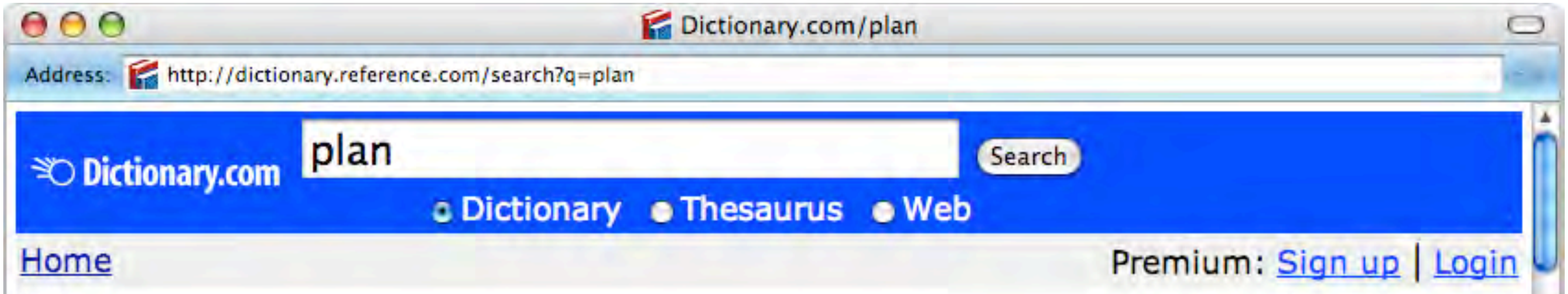
Content of the lecture ONE

1. Components
2. **Mode of the lecture**
   - 1st part will be lectured by Carmel Domshlak in the 2nd week of the term:

     Po: 16:15 - 17:45 (T2:C3-54)
     Út: 16:15 - 17:45 a 18:00 - 19:30 (KN:E112)
     St: 16:15 - 17:45 a 18:00 - 19:30 (KN:E112)
     Ct: 16:15 - 17:45 a 18:00 - 19:30 (KN:E112)
     Pá: 11:00 - 12:30 a 12:45 - 14:15 (KN:G205)

   - March 1 - March 20: Consultation on planning provided by the TAs at tutorials and upon request by the instructor.
   - 3 lectures on adversarial planning will be provided by Michal Pechoucek on 29 March, 5 April, 12 April on Adversarial planning and game playing
3. Motivation
4. Preliminaries

Content of the lecture ONE



To start writing this lecture, I went to the web to get the definition of *plan* as a noun …

Dictionary.com/plan

Address: http://dictionary.reference.com/search?q=plan

Dictionary.com plan Search

Dictionary Thesaurus Web

Home Premium: Sign up | Login

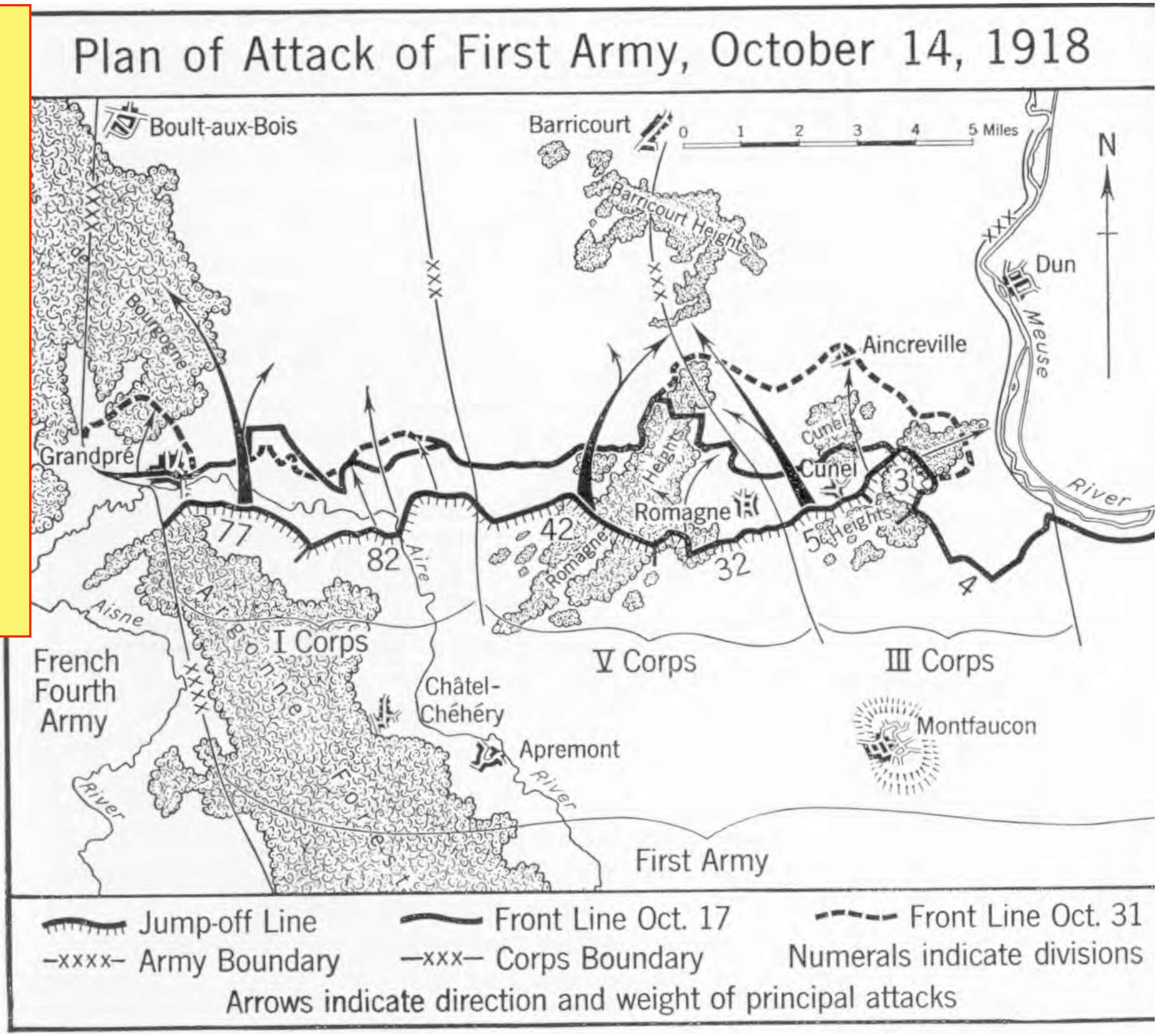
plan *n.*

1. A scheme, program, or method worked out beforehand for the accomplishment of an objective: *a plan of attack.*
2. A proposed or tentative project or course of action: *had no plans for the evening.*
3. A systematic arrangement of elements or important parts; a configuration or outline: *a seating plan; the plan of a story.*
4. A drawing or diagram made to scale showing the structure or arrangement of something.
5. In perspective rendering, one of several imaginary planes perpendicular to the line of vision between the viewer and the object being depicted.
6. A program or policy stipulating a service or benefit: *a pension plan.*

Synonyms: *blueprint, design, project, scheme, strategy*

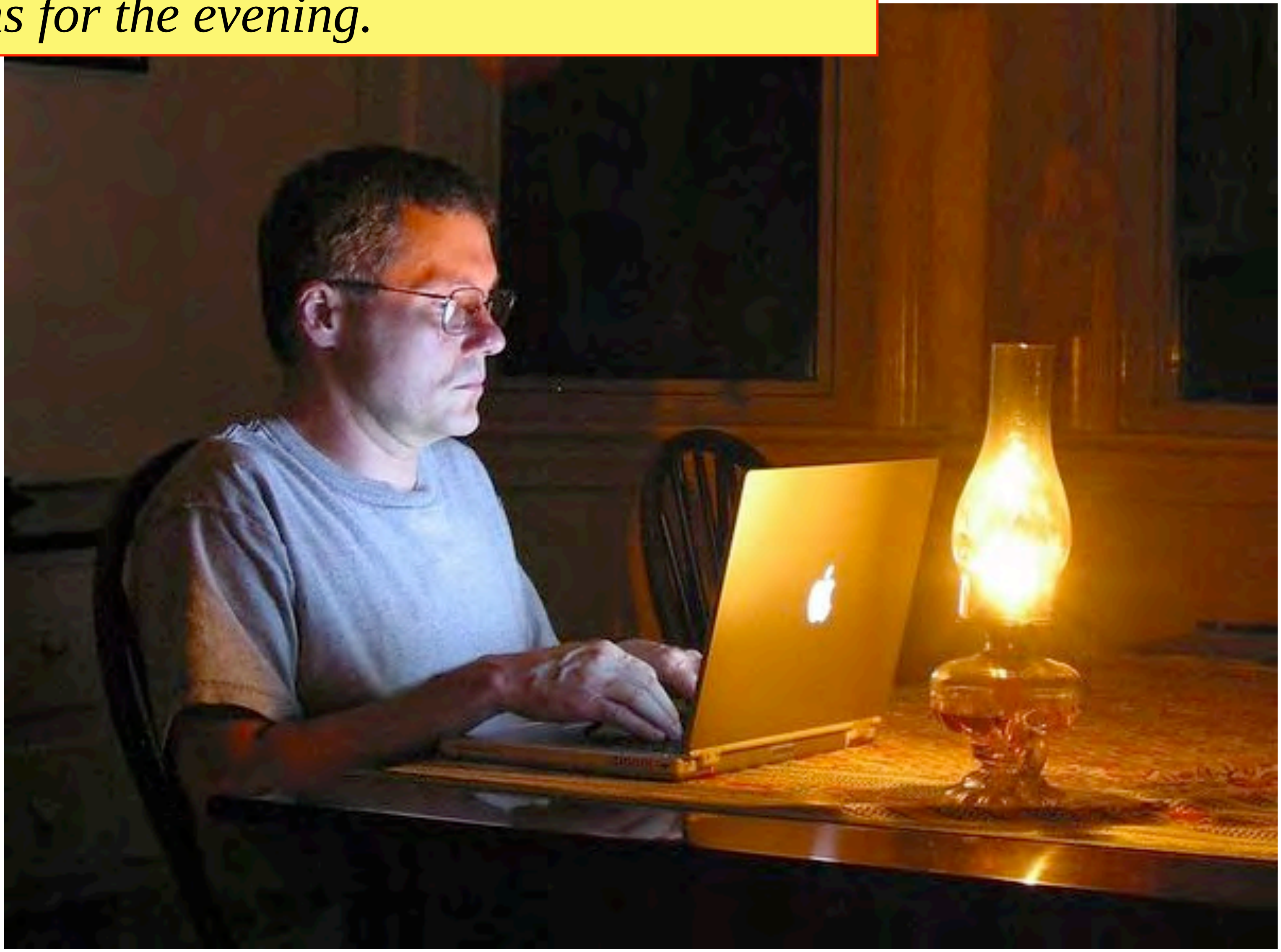
plan *n.*

1. A scheme, program, or method worked out beforehand for the accomplishment of an objective: *a plan of attack.*



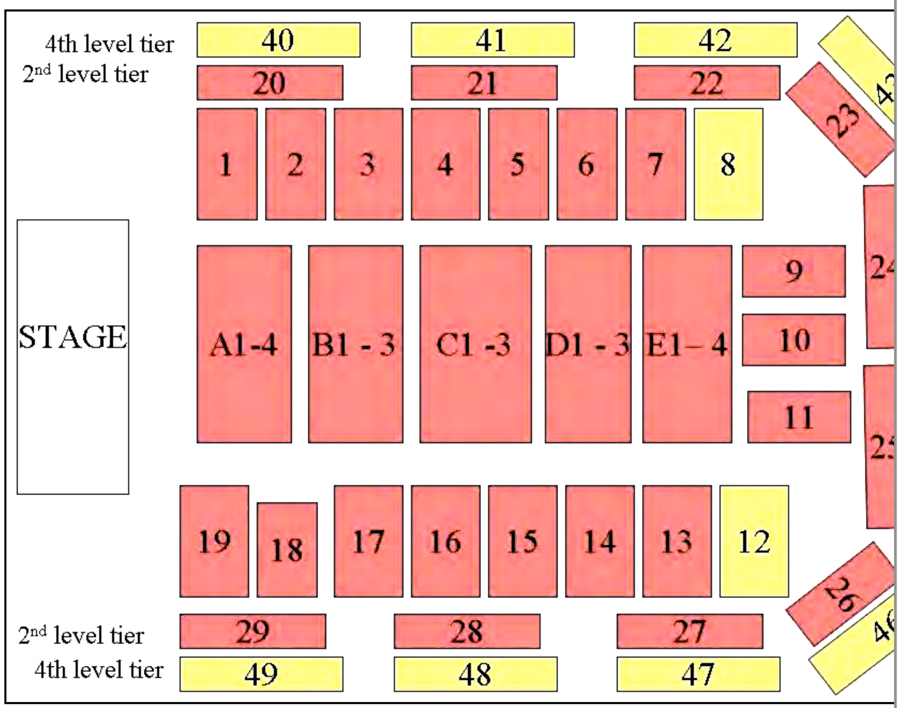
plan *n.*

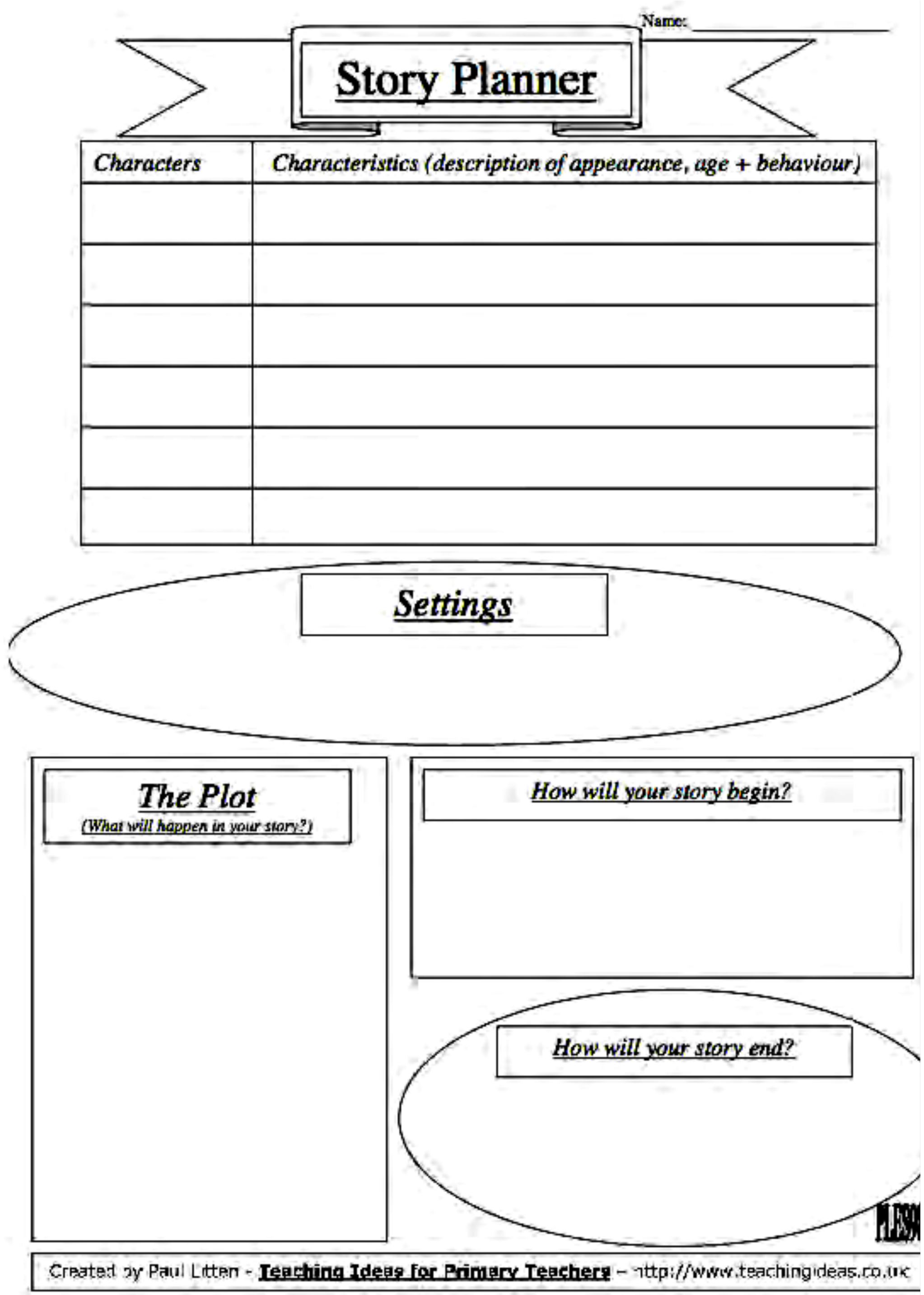
2. A proposed or tentative project or course of action: *had no plans for the evening.*



plan *n.*

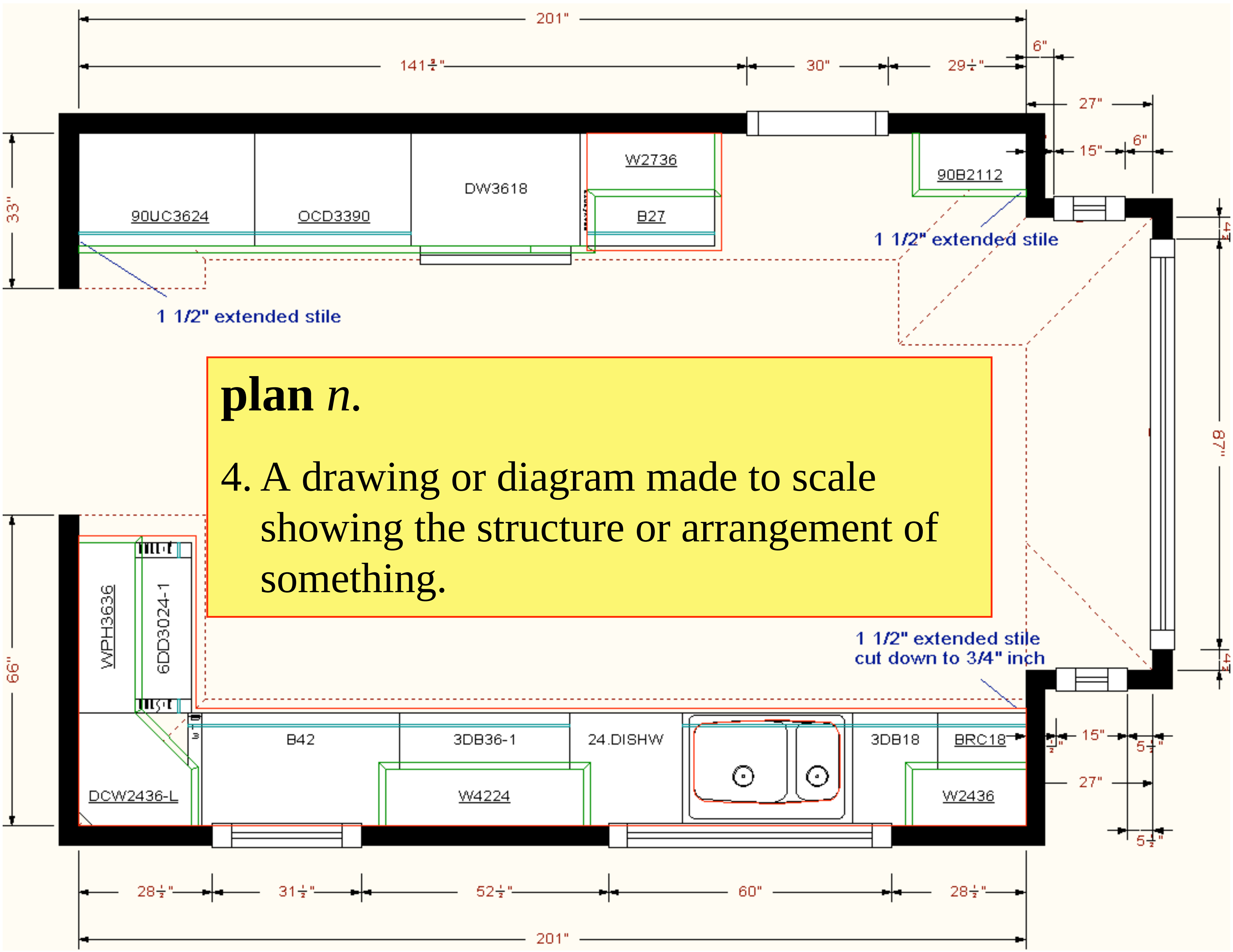
3. A systematic arrangement of elements or important parts; a configuration or outline: *a seating plan; the plan of a story*.





plan *n.*

4. A drawing or diagram made to scale showing the structure or arrangement of something.

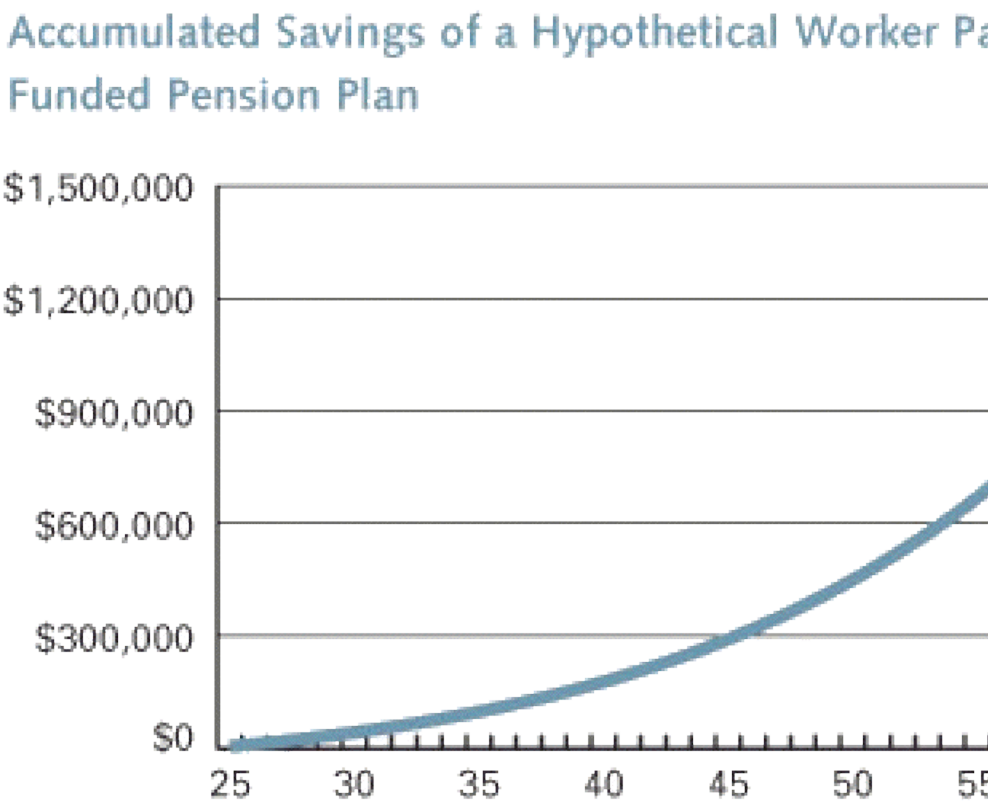


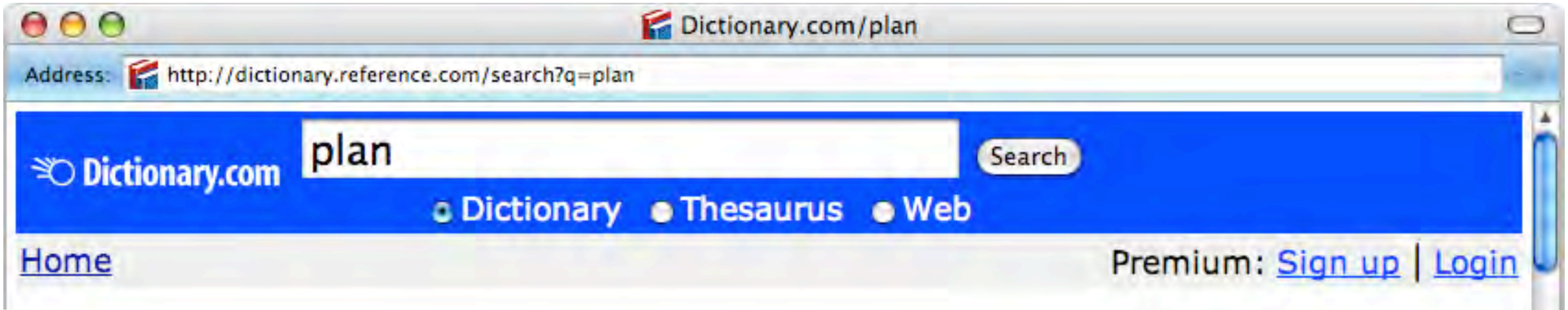
plan *n.*

5. In perspective rendering, one of several imaginary planes perpendicular to the line of vision between the viewer and the object being depicted.

plan *n.*

6. A program or policy stipulating a service or benefit: *a pension plan.*





plan *n.*

1. A scheme, program, or method worked out beforehand for the accomplishment of an objective: *a plan of attack.*
2. A proposed or tentative project or course of action: *had no plans for the evening.*
3. A systematic arrangement of elements or important parts; a configuration or outline: *a seating plan; the plan of a story.*
4. A drawing or diagram made to scale showing the structure or arrangement of something.
5. In perspective rendering, one of several imaginary planes perpendicular to the line of vision between the viewer and the object being depicted.
6. A program or policy stipulating a service or benefit: *a pension plan.*

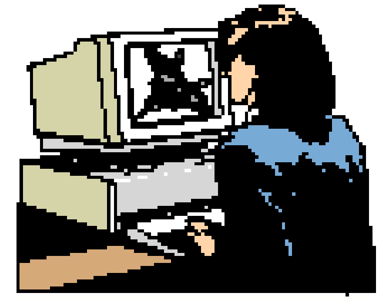
Synonyms: *blueprint, design, project, scheme, strategy* [a representation] of future behavior … usually a set of actions, with temporal and other constraints on them, for execution by some agent or agents. - Austin Tate [*MIT Encyclopedia of the Cognitive Sciences*, 1999]

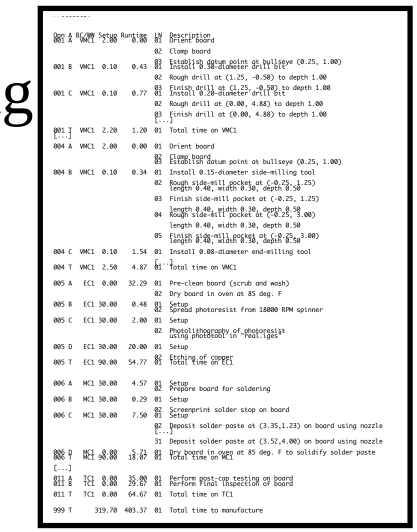
```
                                  03  Establish datum point at bullseye (0.25, 1.00)
                                  01  Install 0.15-diameter side-milling tool
                                  02  Rough side-mill pocket at (-0.25, 1.25)
                                      length 0.40, width 0.30, depth 0.50
                                  03  Finish side-mill pocket at (-0.25, 1.25)
                                      length 0.40, width 0.30, depth 0.50
                                  04  Rough side-mill pocket at (-0.25, 3.00)
                                      length 0.40, width 0.30, depth 0.50
                                  05  Finish side-mill pocket at (-0.25, 3.00)
                                      length 0.40, width 0.30, depth 0.50
                                  01  Install 0.08-diameter end-milling tool
                                  [...]
                                  01  Total time on VMC1
                                  01  Pre-clean board (scrub and wash)
                                  02  Dry board in oven at 85 deg. F
005 B   EC1 30.00    0.48         01  Setup
                                  02  Spread photoresist from 18000 RPM spinner
005 C   EC1 30.00    2.00         01  Setup
                                  02  Photolithography of photoresist
                                      using phototool in "real.iges"
005 D   EC1 30.00   20.00         01  Setup
                                  02  Etching of copper
005 T   EC1 90.00   54.77         01  Total time on EC1
006 A   MC1 30.00    4.57         01  Setup
                                  02  Prepare board for soldering
006 B
                                  02  Screenprint solder stop on board
006 C   MC1 30.00    7.50         01  Setup
```

A portion of a manufacturing process plan

Generating Plans of Action

- Computer programs to aid human planners
  - ◆ Project management (consumer software)
  - ◆ Plan storage and retrieval
    - » e.g., ***variant process planning*** in manufacturing
  - ◆ Automatic schedule generation
    - » various OR and AI techniques
- For some problems, we would like generate plans (or pieces of plans) automatically
  - ◆ Much more difficult
  - ◆ Automated-planning research is starting to pay off
- Here are some examples …

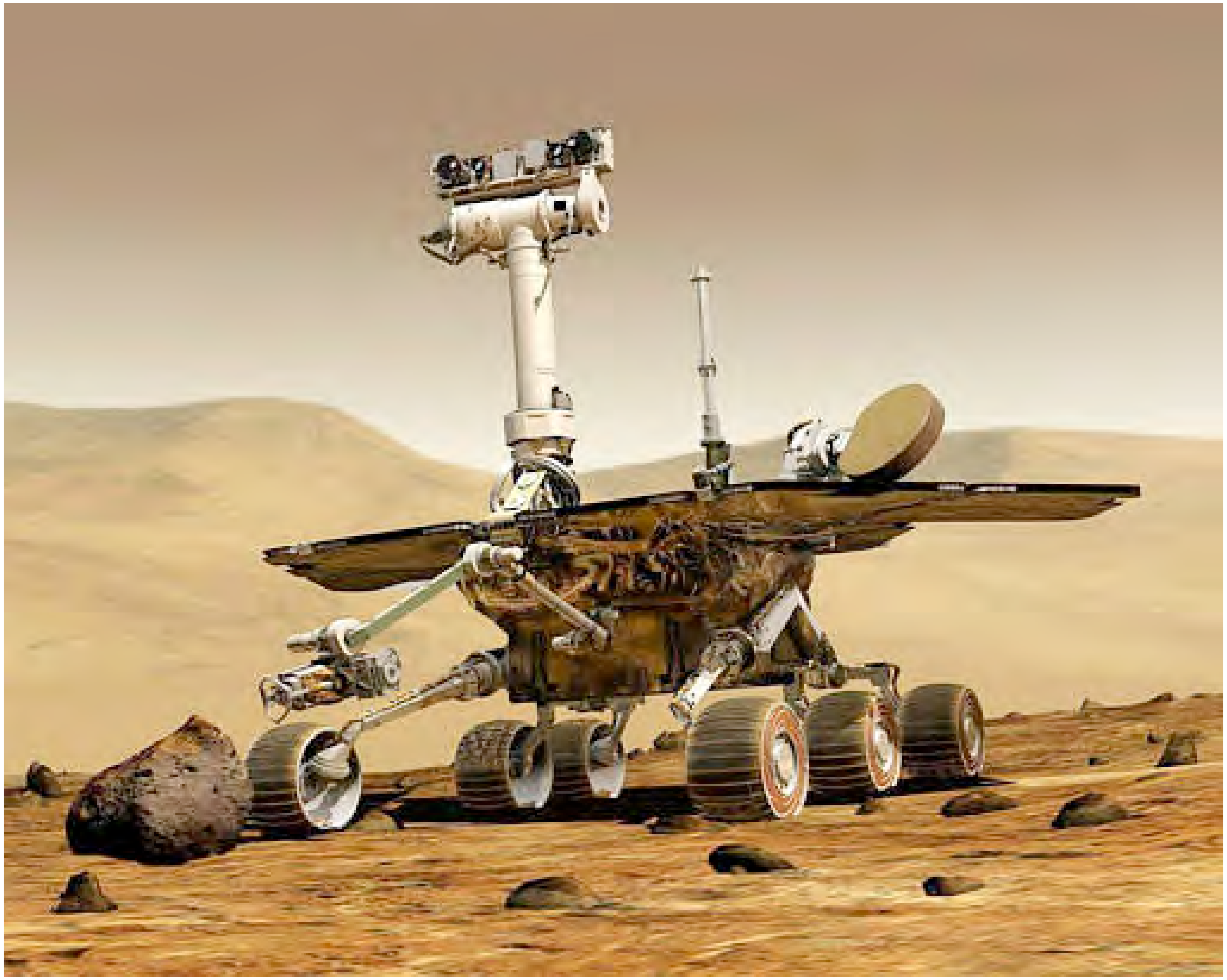




Opn	A	BC/WW	Setup	Runtime	LN	Description
001	A	VMC1	2.00	0.00	01	Orient board
					02	Clamp board
					03	Establish datum point at bullseye (0.25, 1.00)
001	B	VMC1	0.10	0.43	01	Install 0.30-diameter drill bit
					02	Rough drill at (1.25, -0.50) to depth 1.00
					03	Finish drill at (1.25, -0.50) to depth 1.00
001	C	VMC1	0.10	0.77	01	Install 0.20-diameter drill bit
					02	Rough drill at (0.00, 4.88) to depth 1.00
					03	Finish drill at (0.00, 4.88) to depth 1.00
					[...]	
001	T	VMC1	2.20	1.20	01	Total time on VMC1
[...]						
004	A	VMC1	2.00	0.00	01	Orient board
					02	Clamp board
					03	Establish datum point at bullseye (0.25, 1.00)
004	B	VMC1	0.10	0.34	01	Install 0.15-diameter side-milling tool
					02	Rough side-mill pocket at (-0.25, 1.25) length 0.40, width 0.30, depth 0.50
					03	Finish side-mill pocket at (-0.25, 1.25) length 0.40, width 0.30, depth 0.50
					04	Rough side-mill pocket at (-0.25, 3.00) length 0.40, width 0.30, depth 0.50
					05	Finish side-mill pocket at (-0.25, 3.00) length 0.40, width 0.30, depth 0.50
004	C	VMC1	0.10	1.54	01	Install 0.08-diameter end-milling tool
					[...]	
004	T	VMC1	2.50	4.87	01	Total time on VMC1
005	A	EC1	0.00	32.29	01	Pre-clean board (scrub and wash)
					02	Dry board in oven at 85 deg. F
005	B	EC1	30.00	0.48	01	Setup
					02	Spread photoresist from 18000 RPM spinner
005	C	EC1	30.00	2.00	01	Setup
					02	Photolithography of photoresist using phototool in "real.iges"
005	D	EC1	30.00	20.00	01	Setup
					02	Etching of copper
005	T	EC1	90.00	54.77	01	Total time on EC1
006	A	MC1	30.00	4.57	01	Setup
					02	Prepare board for soldering
006	B	MC1	30.00	0.29	01	Setup
					02	Screenprint solder stop on board
006	C	MC1	30.00	7.50	01	Setup
					02	Deposit solder paste at (3.35,1.23) on board using nozzle
					[...]	
					31	Deposit solder paste at (3.52,4.00) on board using nozzle
006	D	MC1	0.00	5.71	01	Dry board in oven at 85 deg. F to solidify solder paste
006	T	MC1	90.00	18.07	01	Total time on MC1
[...]						
011	A	TC1	0.00	35.00	01	Perform post-cap testing on board
011	B	TC1	0.00	29.67	01	Perform final inspection of board
011	T	TC1	0.00	64.67	01	Total time on TC1
999	T		319.70	403.37	01	Total time to manufacture

Space Exploration

- Autonomous planning, scheduling, control
  - NASA: JPL and Ames
- Remote Agent Experiment (RAX)
  - Deep Space 1
- Mars Exploration Rover (MER)

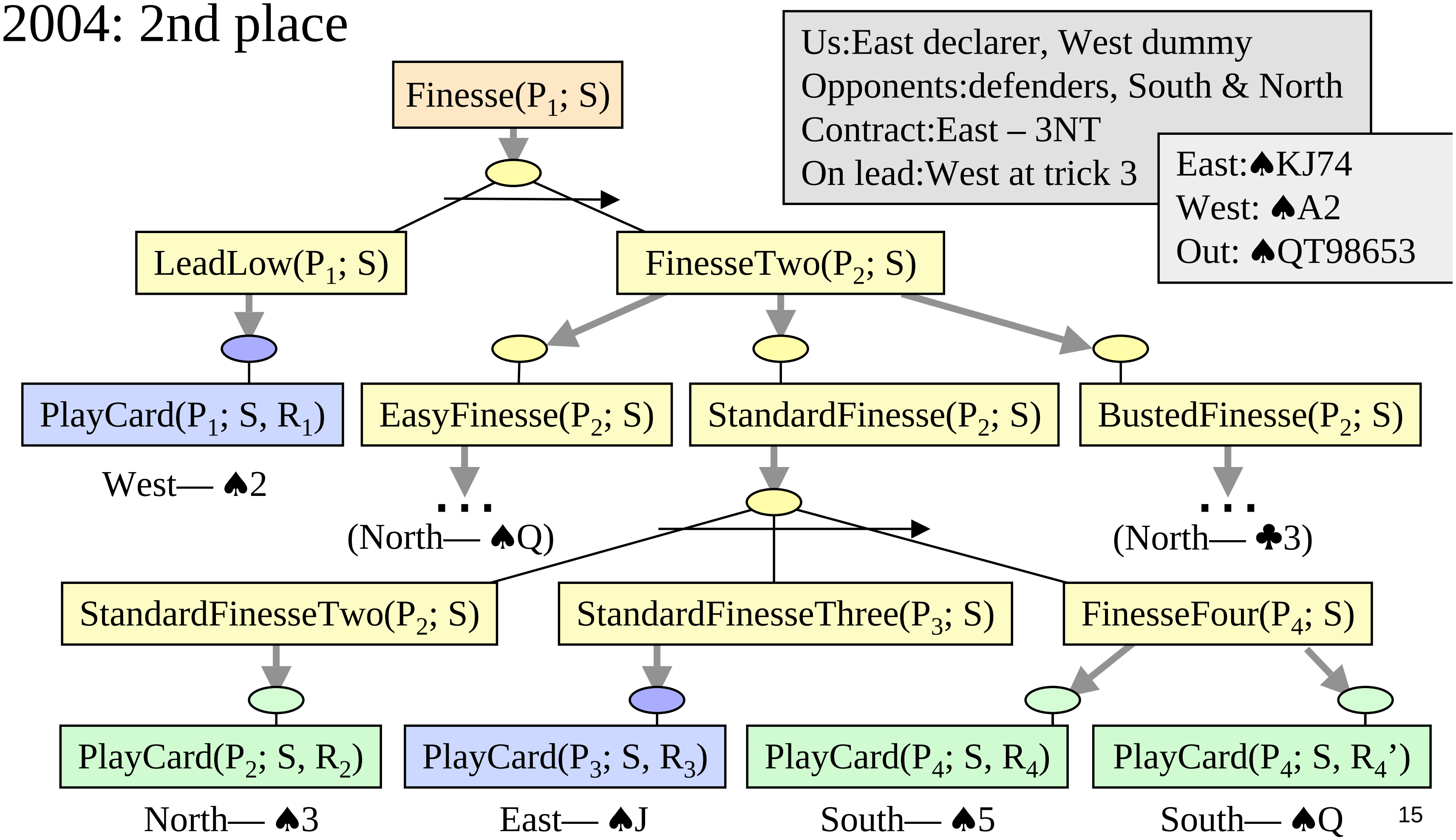


Manufacturing

- Sheet-metal bending machines - Amada Corporation
  - Software to plan the sequence of bends
    [Gupta and Bourne, *J. Manufacturing Sci. and Engr.*, 1999]

Games

- *Bridge Baron* - Great Game Products
  - 1997 world champion of computer bridge
    [Smith, Nau, and Throop, *AI Magazine*, 1998]
  - 2004: 2nd place



Planning the *free-flight* UAV

Planning the *free-flight* UAV

Planning for the information collection

Planning in urban areas

Adversarial planning

Maritime domain planning

Definition of planning

Planning

Reasoning about about hypothetical interaction among the agent and the environment with respect to a given task. motivation of the planning process is to reason about possible course of actions that will change the environment in order to reach the goal (task).

Planning $\times$ Scheduling

while scheduling assigns in time resources to separate processes planning considers possible interaction among components of plan

- **planning**: we have the initial state, goal state, operators and want to find a sequence of operators that will reach the goal state from the initial state (by selecting appropriate actions, arranging the action and considering the causalities)
- **scheduling**: we have set of resources, actions and constraints and we want to form an appropriate schedule that meets the constraints (by arranging the actions, assigning resources and satisfying the constrains)

Content of the lecture ONE

Preliminaries

- Propositional logic

Preliminaries

- Propositional logic
- Hill-climbing

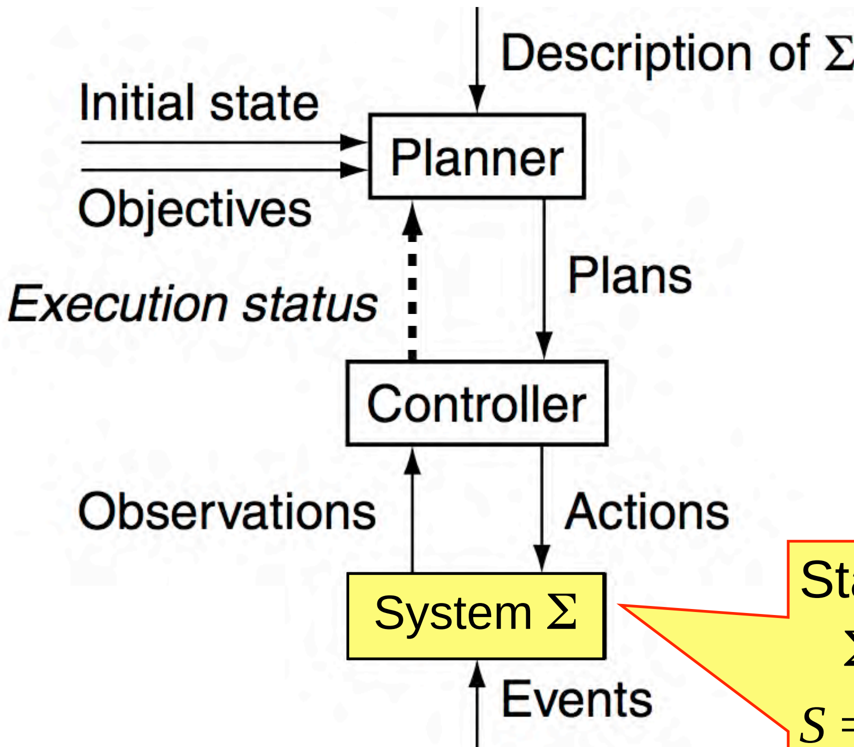
Preliminaries

- Propositional logic
- Hill-climbing
- A*

Outline

- Conceptual model for planning
- Example planning algorithms
- What's bad
- What's good

Conceptual Model
1. Environment



State transition system
$\Sigma = (S,A,E,\gamma)$

S = {states}

A = {actions}

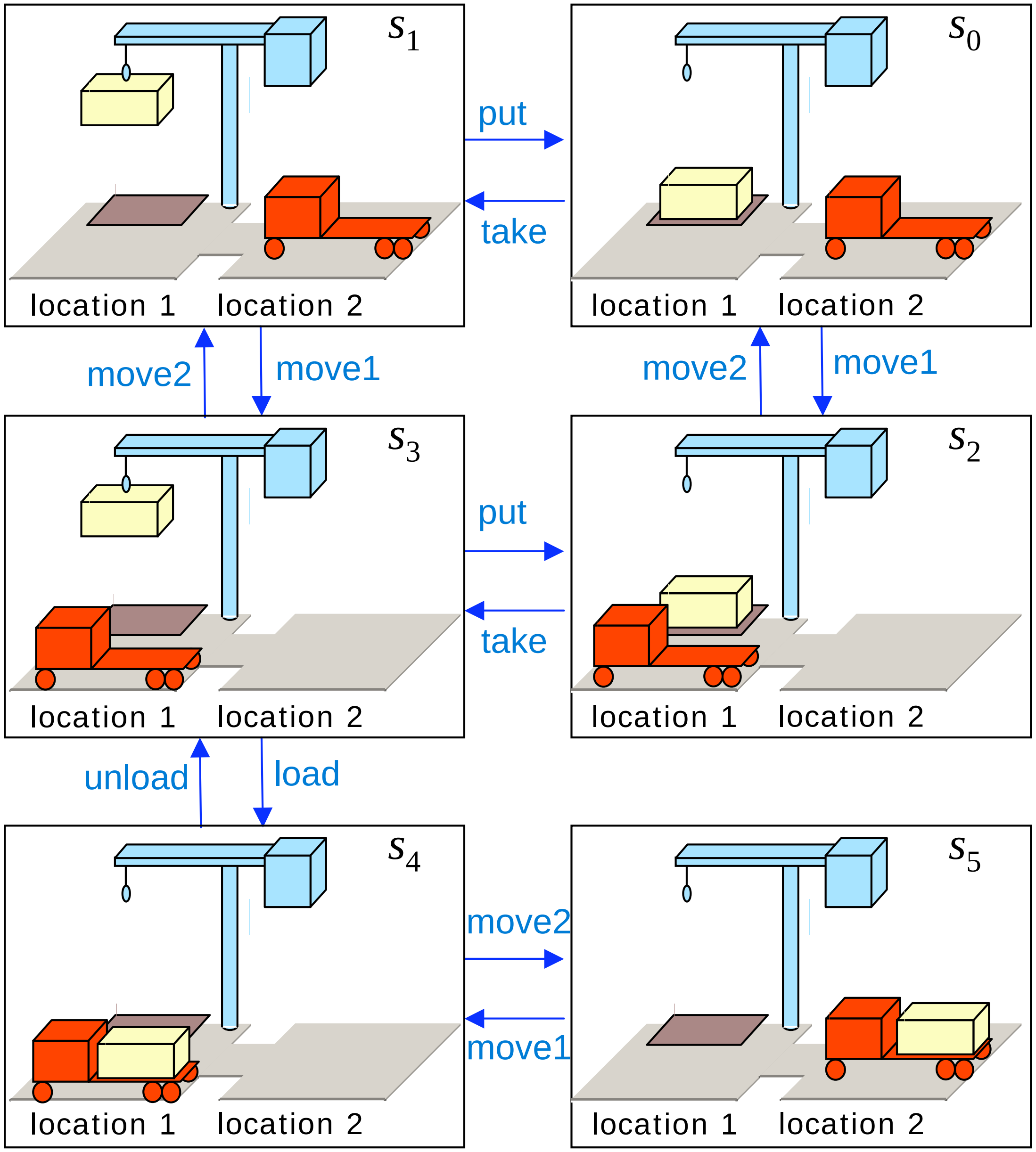
E = {exogenous events}

γ = state-transition function

State Transition System

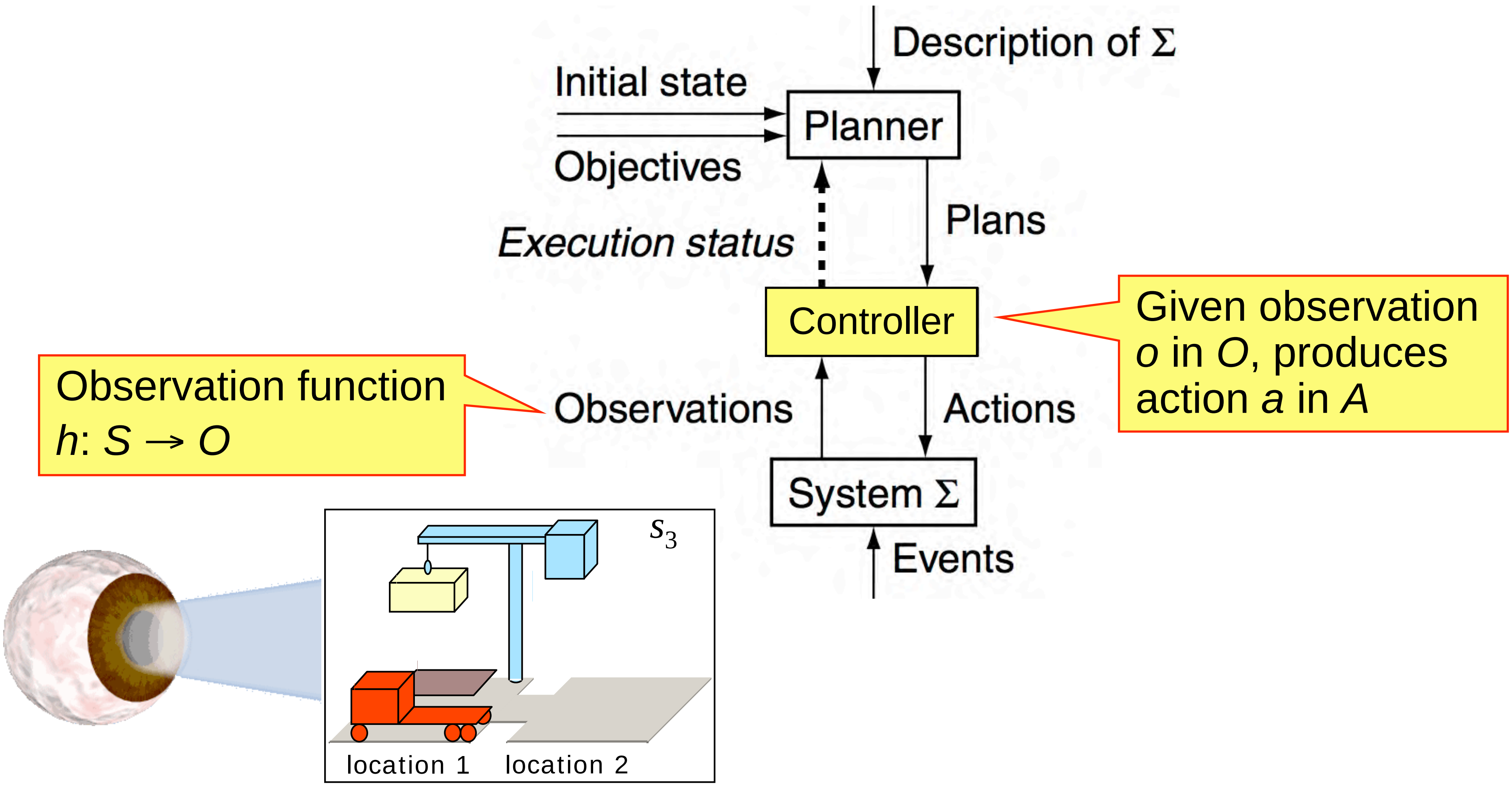
$\Sigma = (S,A,E,\gamma)$

- S = {states}
- A = {actions}
- E = {exogenous events}
- State-transition function $\gamma: S \times (A \cup E) \rightarrow 2^S$
  - $S = \{s_0, \ldots, s_5\}$
  - A = {move1, move2, put, take, load, unload}
  - $E = \{\}$
  - γ: see the arrows

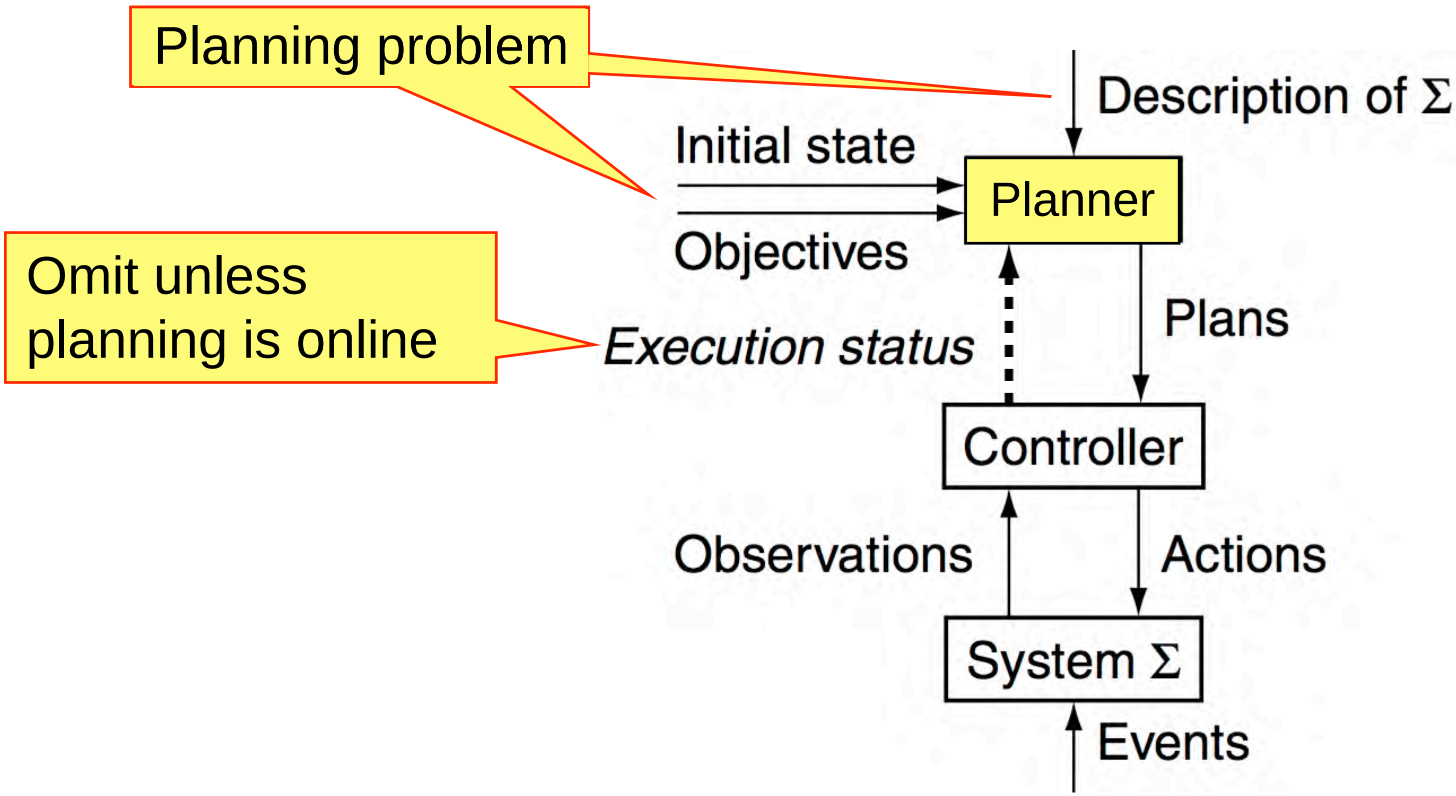


The Dock Worker Robots (DWR) domain

Conceptual Model
2. Controller



Conceptual Model
3. Planner's Input



Planning Problem

Description of Σ

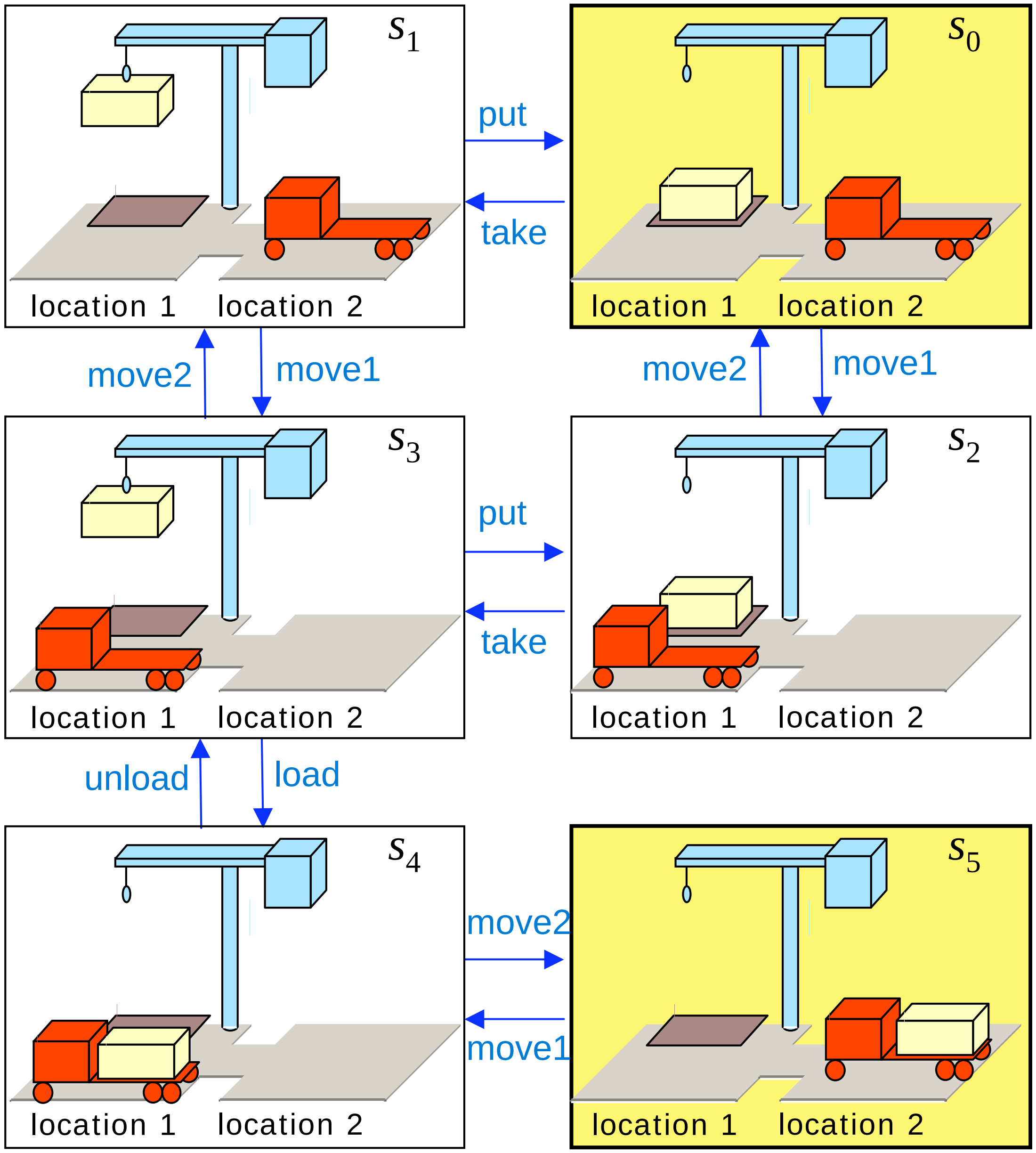
Initial state or set of states

Initial state = s_0

Objective

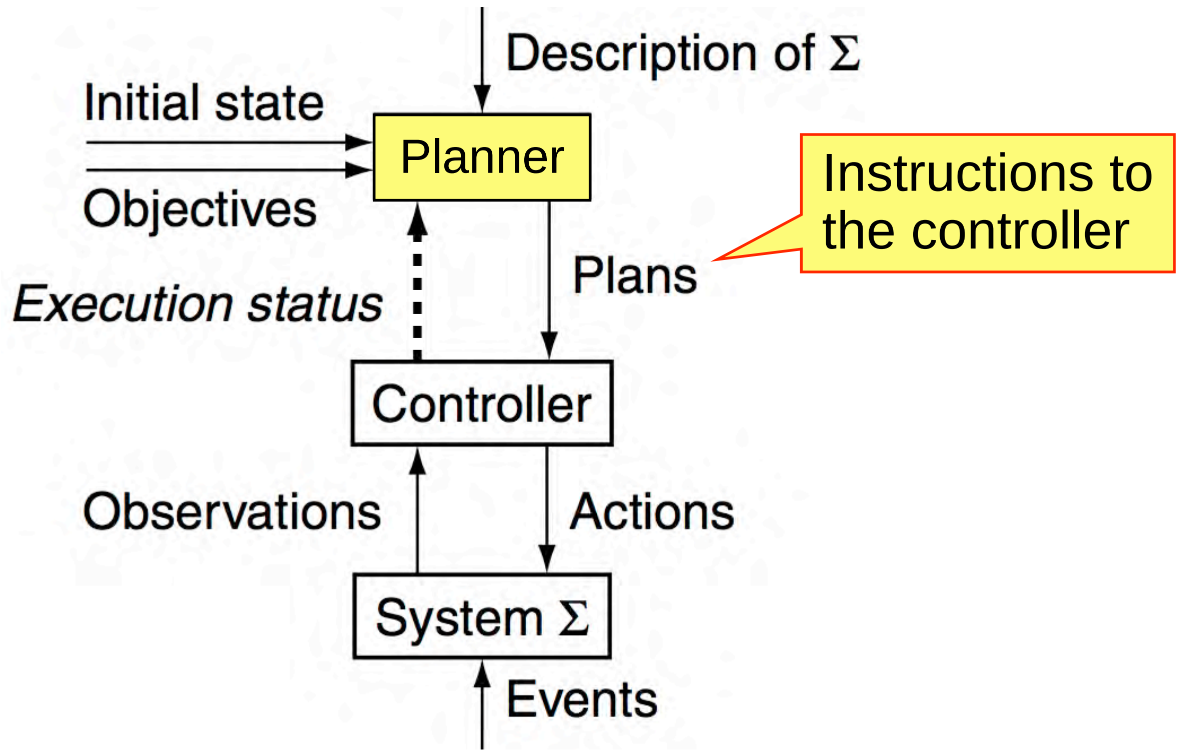
Goal state, set of goal states, set of tasks, “trajectory” of states, objective function, …

Goal state = s_5



The Dock Worker Robots (DWR) domain

Conceptual Model
4. Planner’s Output



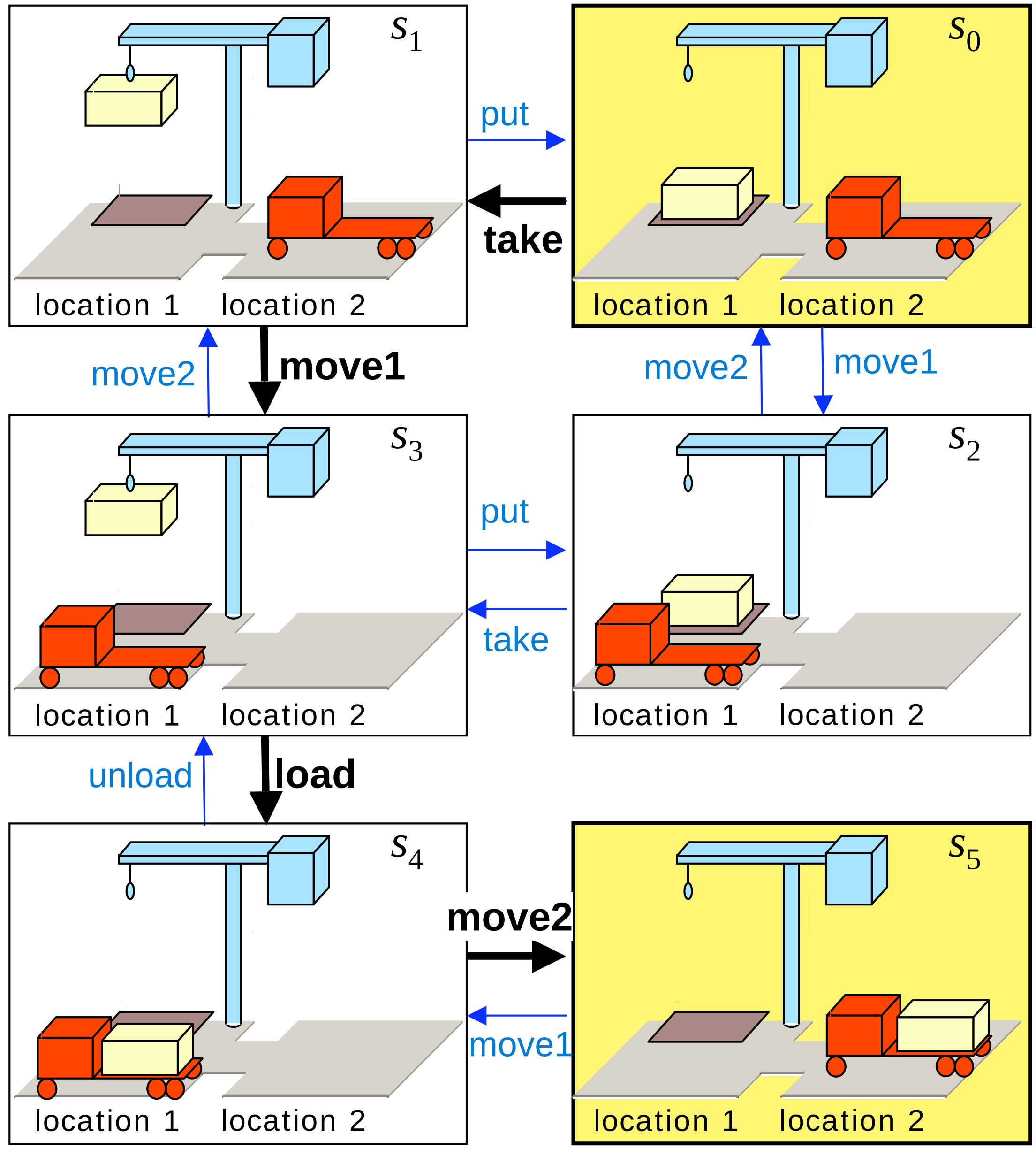
Plans

Classical plan: a sequence of actions

⟨take, move1, load, move2⟩

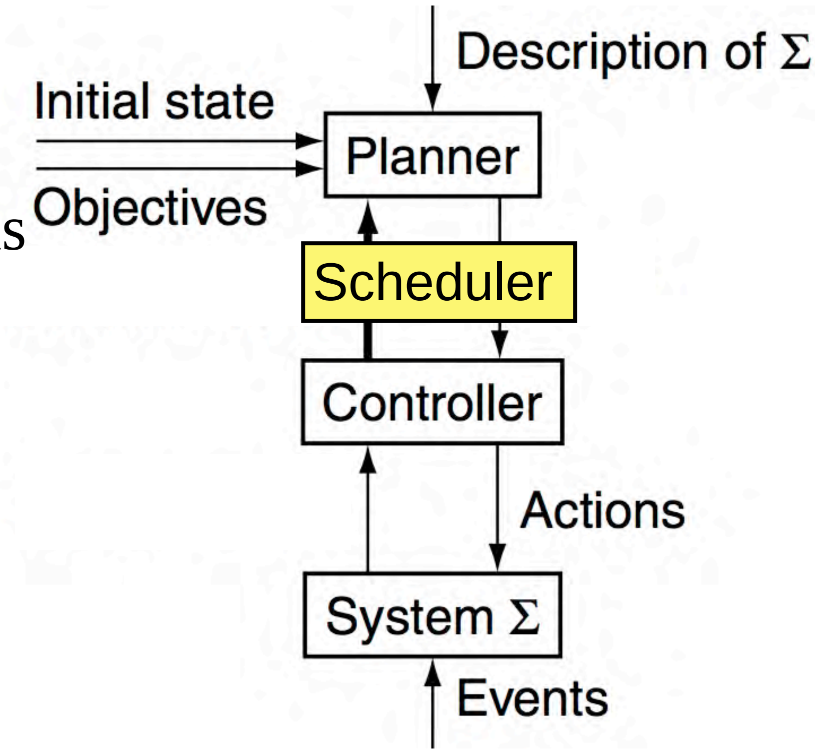
Policy: partial function from *S* into *A*

{(s_0, take),
(s_1, move1),
(s_3, load),
(s_4, move2)}

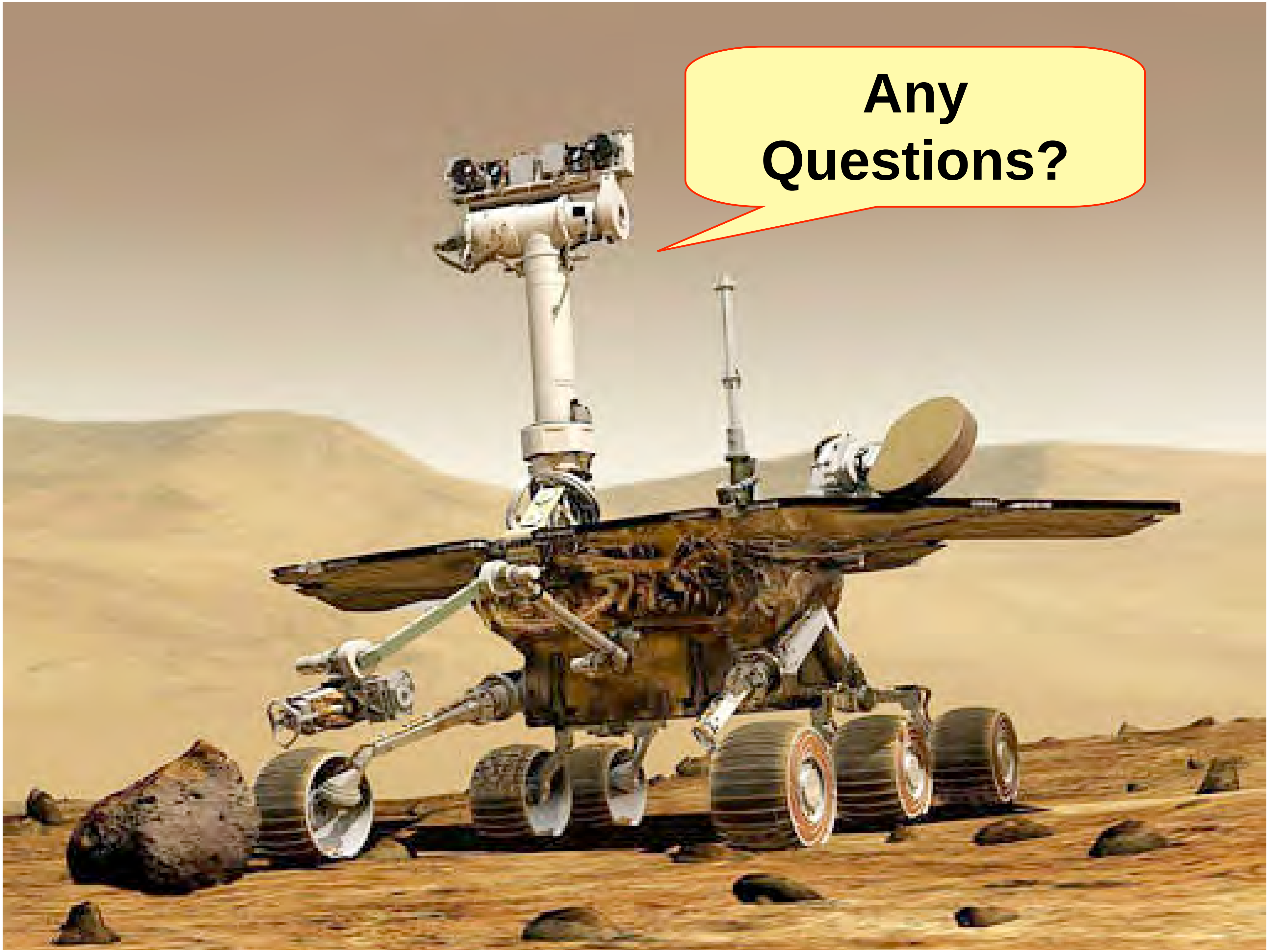


The Dock Worker Robots (DWR) domain

Planning Versus Scheduling



- Scheduling
  - Decide when and how to perform a given set of actions
    - » Time constraints
    - » Resource constraints
    - » Objective functions
  - Typically NP-complete
- Planning
  - Decide what actions to use to achieve some set of objectives
  - Can be much worse than NP-complete; worst case is undecidable


Any
Questions?

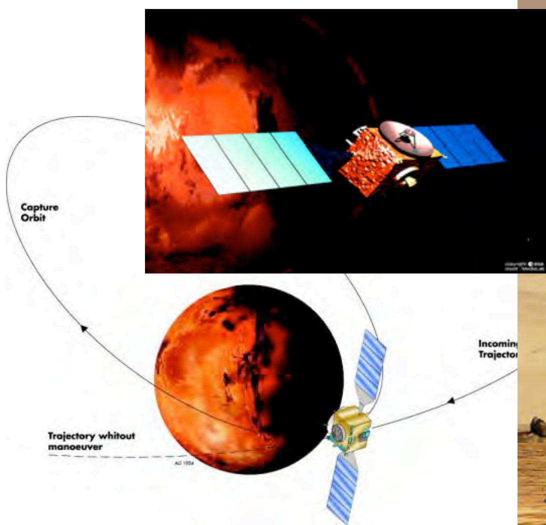
Three Main Types of Planners

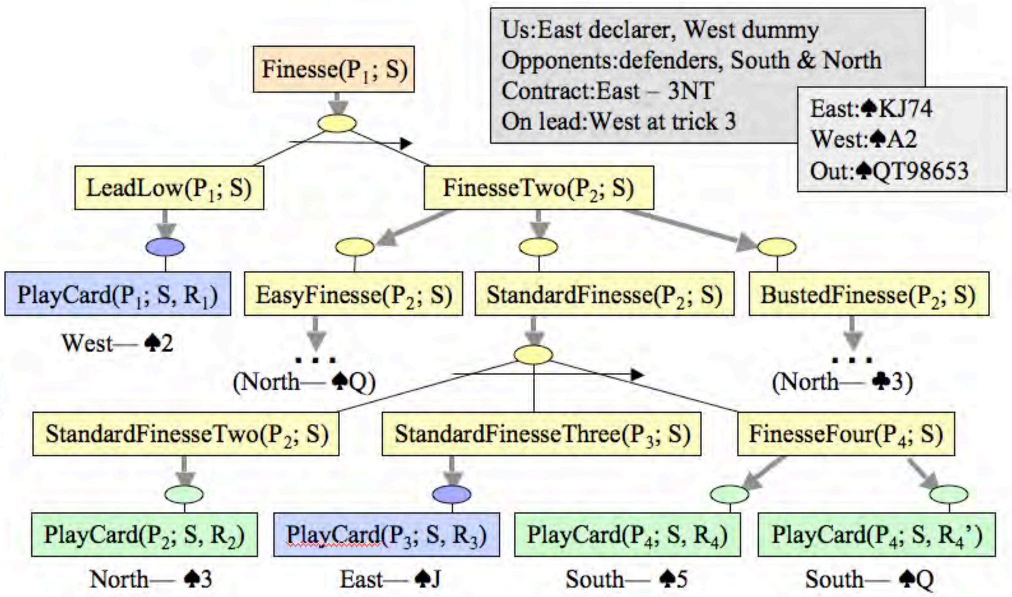
1. Domain-specific
2. Domain-independent
3. Configurable

- I'll talk briefly about each

Types of Planners: 1. Domain-Specific (Chapters 19-23)

- Made or tuned for a specific domain
- Won't work well (if at all) in any other domain
- Most successful real-world planning systems work this way



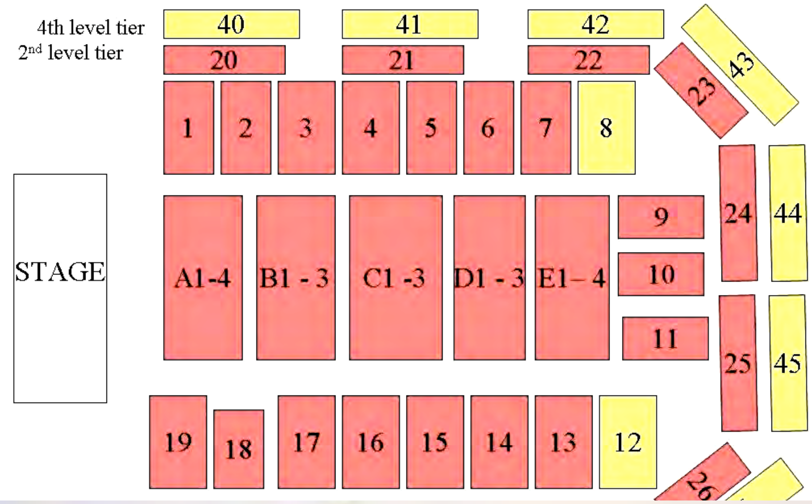


Types of Planners
2. Domain-Independent

- In principle, a domain-independent planner works in any planning domain
- Uses no domain-specific knowledge except the definitions of the basic actions

Types of Planners
2. Domain-Independent

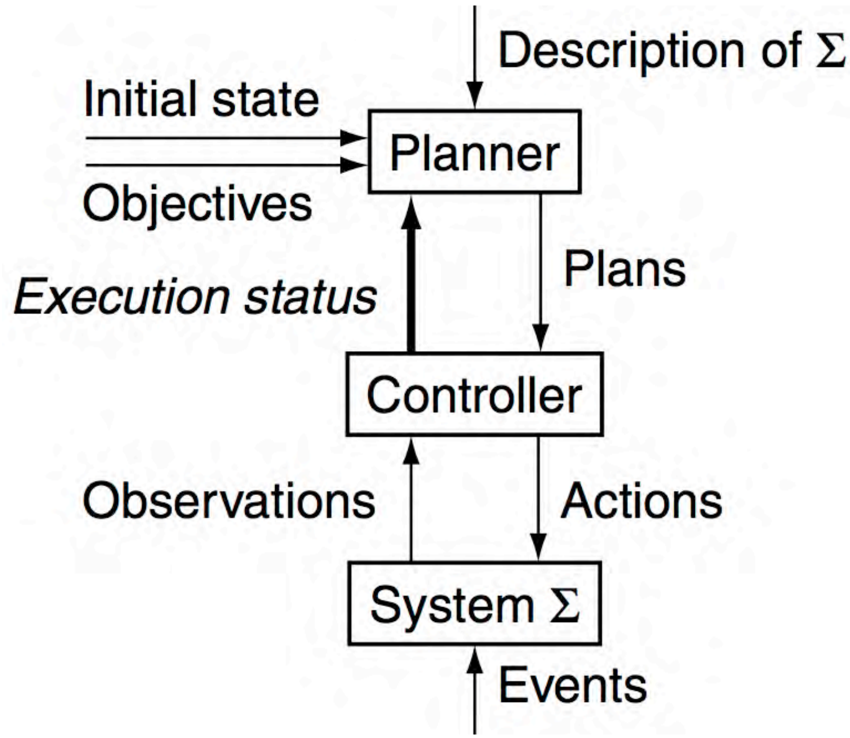
- In practice,
  - Not feasible to develop domain-independent planners that work in *every* possible domain



- Make simplifying assumptions to restrict the set of domains
  - *Classical planning*
  - Historical focus of most automated-planning research

Restrictive Assumptions

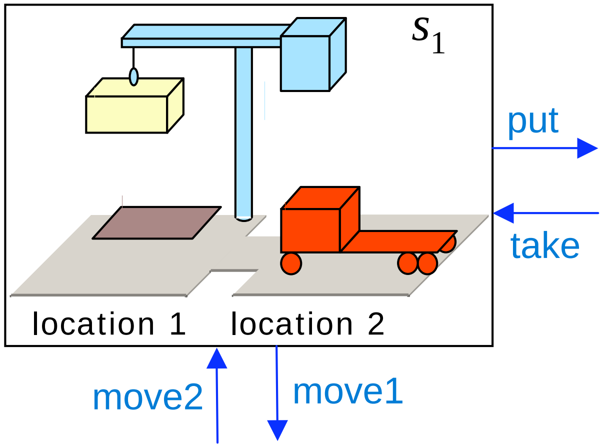
- **A0: Finite system:**
  - finitely many states, actions, events
- **A1: Fully observable:**
  - the controller always Σ's current state
- **A2: Deterministic:**
  - each action has only one outcome
- **A3: Static** (no exogenous events):
  - no changes but the controller's actions
- **A4: Attainment goals:**
  - a set of goal states S_g
- **A5: Sequential plans:**
  - a plan is a linearly ordered sequence of actions $(a_1, a_2, \ldots a_n)$
- **A6: Implicit time:**
  - no time durations; linear sequence of instantaneous states
- **A7: Off-line planning:**
  - planner doesn't know the execution status



Classical Planning (Chapters 2-9)

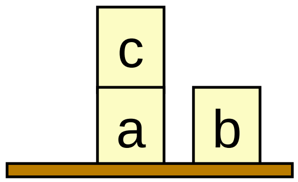
- Classical planning requires all eight restrictive assumptions
  - Offline generation of action sequences for a deterministic, static, finite system, with complete knowledge, attainment goals, and implicit time
- Reduces to the following problem:
  - Given (Σ, s_0, S_g)
  - Find a sequence of actions $(a_1, a_2, \ldots a_n)$ that produces a sequence of state transitions $(s_1, s_2, \ldots, s_n)$ such that s_n is in S_g.
- This is just path-searching in a graph
  - Nodes = states
  - Edges = actions
- ***Is this trivial?***

Classical Planning (Chapters 2-9)



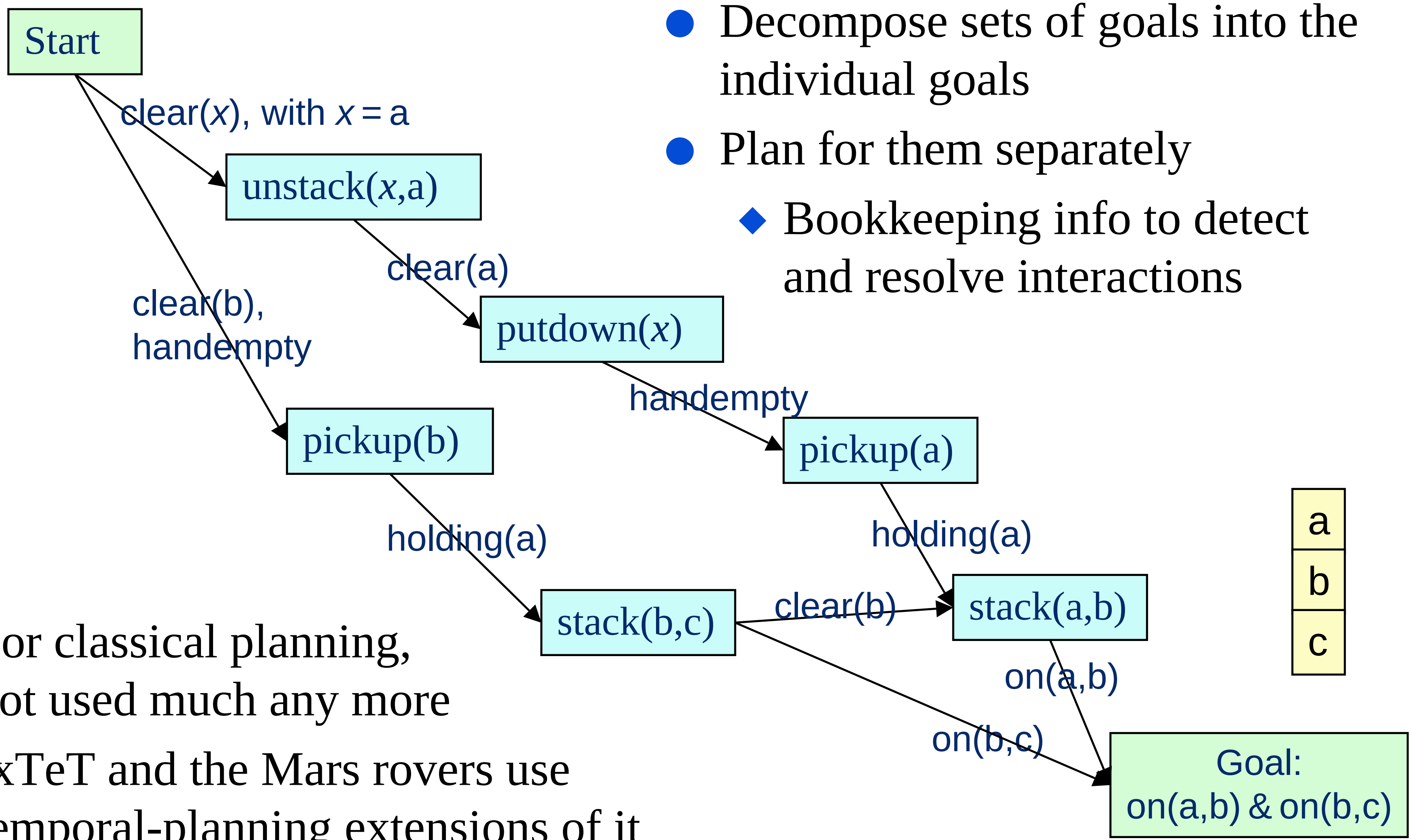
- Generalize the earlier example:
  - Five locations, three robot carts, 100 containers, three piles
    - » Then there are 10^{277} states
- Number of particles in the universe is only about 10^{87}
  - The example is more than 10^{190} times as large!

- Automated-planning research has been heavily dominated by classical planning
  - Dozens (hundreds?) of different algorithms
  - I'll briefly describe a few of the best-known ones



Plan-Space Planning (Chapter 5)

- Decompose sets of goals into the individual goals
- Plan for them separately
  - Bookkeeping info to detect and resolve interactions



For classical planning, not used much any more

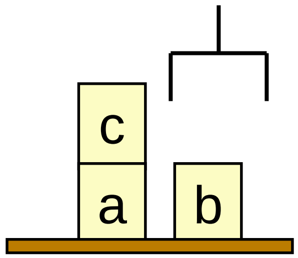
IxTeT and the Mars rovers use temporal-planning extensions of it

Planning Graphs (Chapter 6)

- Relaxed problem [Blum & Furst, 1995]
- Apply all applicable actions at once
- Next "level" contains all the effects of all of those actions

Level 0

Literals in s_0

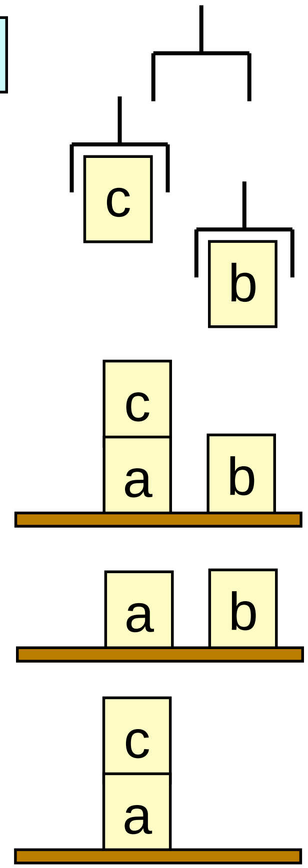


Level 1

All actions applicable to s_0

unstack(c,a)
pickup(b)
no-op

All effects of those actions



Level 2

All actions applicable to subsets of Level 1

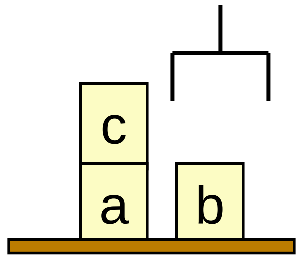
unstack(c,a)
pickup(b)
pickup(a)
stack(b,c)
stack(b,a)
putdown(b)
stack(c,b)
stack(c,a)
putdown(c)
no-op

All effects of those actions

• • •

Level 0

Literals in s_0



Level 1

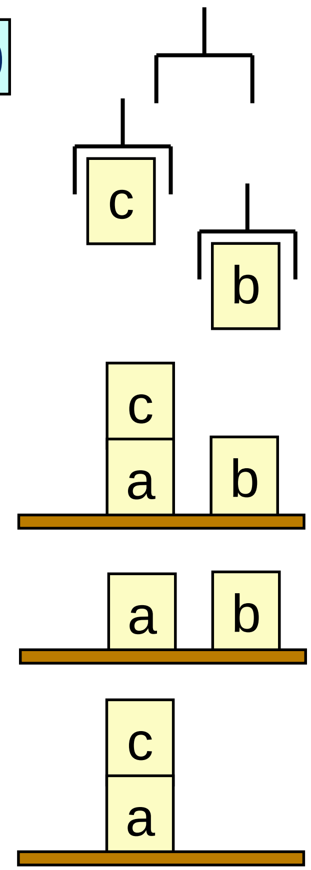
All actions applicable to s_0

unstack(c,a)

pickup(b)

no-op

All effects of those actions



Level 2

All actions applicable to subsets of Level 1

unstack(c,a)

pickup(b)

pickup(a)

stack(b,c)

stack(b,a)

putdown(b)

stack(c,b)

stack(c,a)

putdown(c)

no-op

All effects of those actions

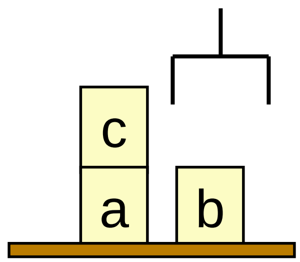
• • •

Graphplan

- For n = 1, 2, …
  - Make planning graph of n levels (*polynomial time*)
  - State-space search *within the planning graph*
- Graphplan's many children
  - IPP, CGP, DGP, LGP, PGP, SGP, TGP, …

Level 0

Literals in s_0



Level 1

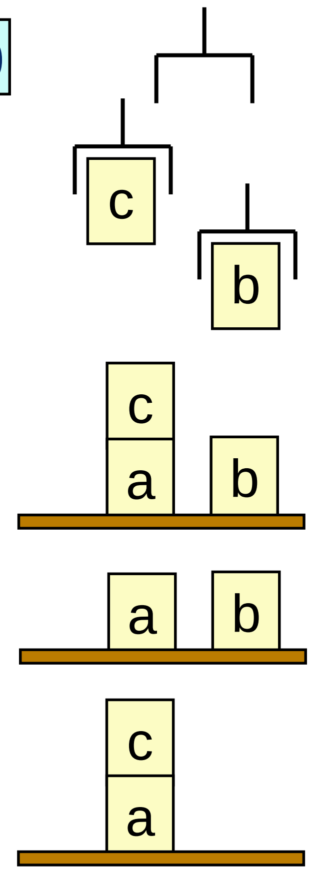
All actions applicable to s_0

unstack(c,a)

pickup(b)

no-op

All effects of those actions



Level 2

All actions applicable to subsets of Level 1

unstack(c,a)

pickup(b)

pickup(a)

stack(b,c)

stack(b,a)

putdown(b)

stack(c,b)

stack(c,a)

putdown(c)

no-op

All effects of those actions

• • •

Graphplan

- For n = 1, 2, …
  - Make planning graph of n levels (*polynomial time*)
  - State-space search *within the planning graph*
- Graphplan's many children
  - IPP, CGP, DGP, LGP, PGP, SGP, TGP, …

Running out of names

Heuristic Search (Chapter 9)

- Can we do an A*-style heuristic search?
- For many years, nobody could come up with a good *h* function
  - But planning graphs make it feasible
    - » Can extract *h* from the planning graph
- Problem: A* quickly runs out of memory
  - So do a greedy search
- Greedy search can get trapped in local minima
  - Greedy search plus local search at local minima
- HSP [Bonet & Geffner]
- FastForward [Hoffmann]

Translation to Other Domains (Chapters 7, 8)

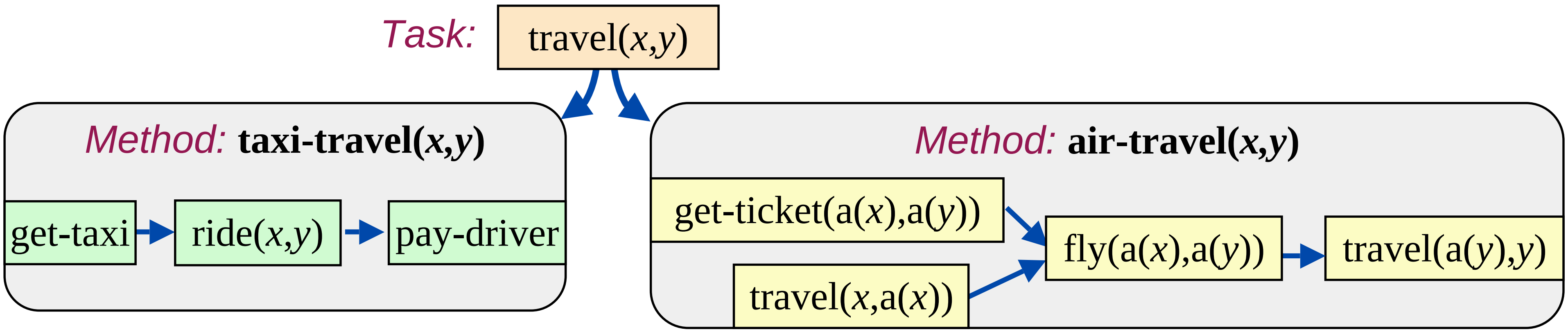
- Translate the planning problem or the planning graph into another kind of problem for which there are efficient solvers
  - Find a solution to that problem
  - Translate the solution back into a plan

- Satisfiability solvers, especially those that use local search
  - Satplan and Blackbox [Kautz & Selman]

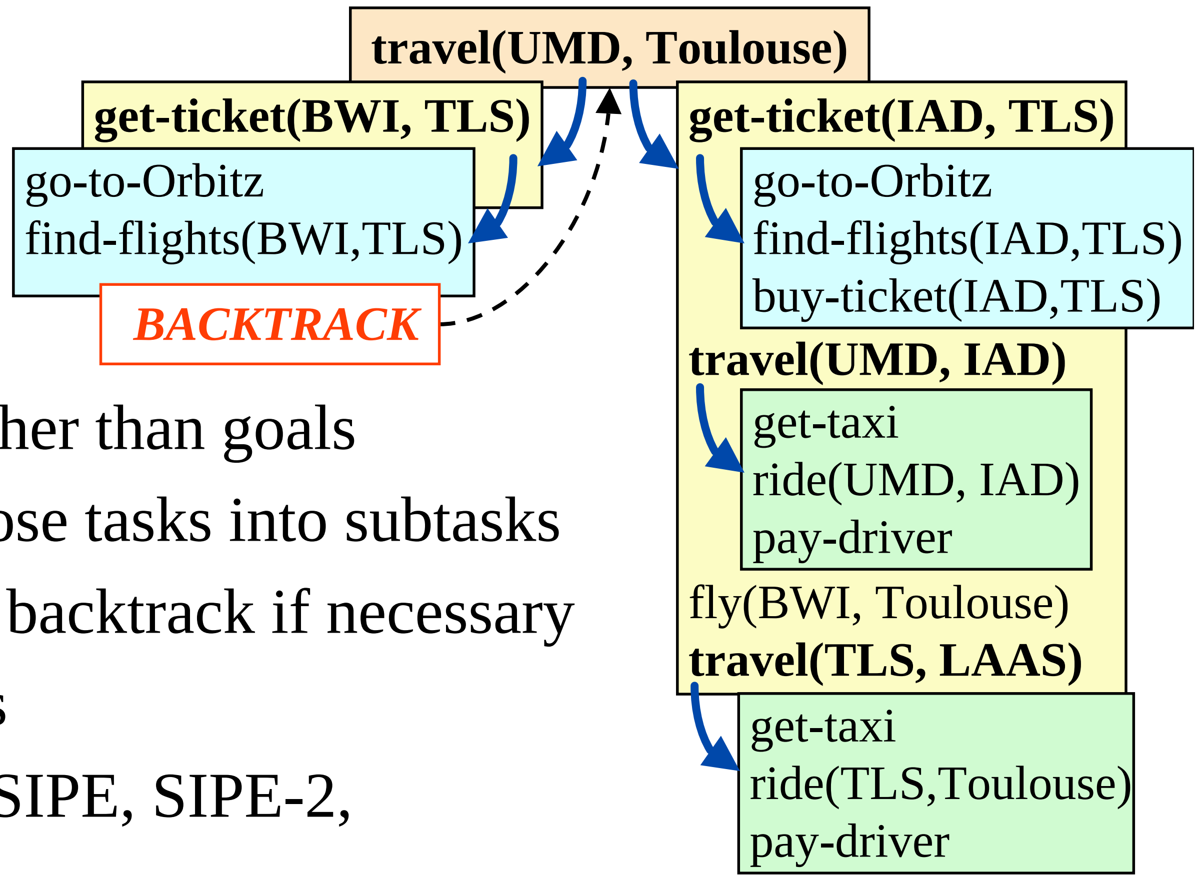
- Integer programming solvers such as Cplex
  - [Vossen *et al.*]

Types of Planners: 3. Configurable

- Domain-independent planners are quite slow compared with domain-specific planners
  - Blocks world in linear time [Slaney and Thiébaux, *A.I.*, 2001]
  - Can get analogous results in many other domains
- But we don't want to write a whole new planner for every domain!
- **Configurable planners**
  - Domain-independent planning engine
  - Input includes info about how to solve problems in the domain
    - » Hierarchical Task Network (HTN) planning
    - » Planning with control formulas

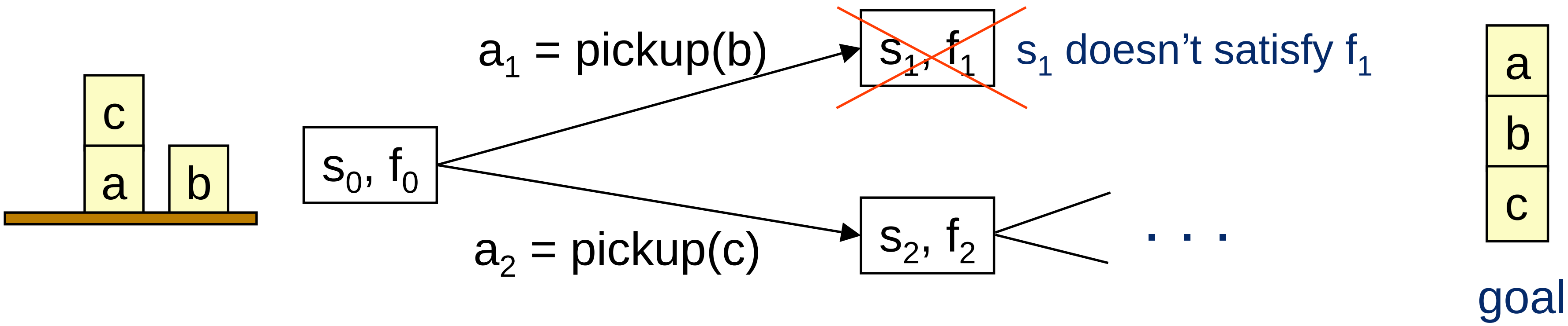


HTN Planning (Chapter 11)



- Problem reduction
  - *Tasks* (activities) rather than goals
  - *Methods* to decompose tasks into subtasks
  - Enforce constraints, backtrack if necessary
- Real-world applications
- Noah, Nonlin, O-Plan, SIPE, SIPE-2, SHOP, SHOP2

Planning with Control Formulas (Chapter 10)



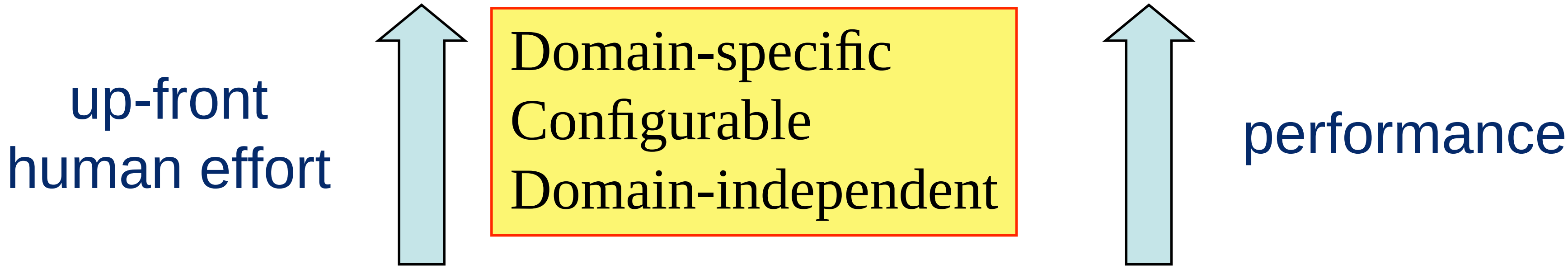
- At each state s_i we have a *control formula* f_i in temporal logic

$$\boxed{ontable(x) \wedge \neg\exists[y{:}\mathrm{GOAL}(on(x,y))] \Rightarrow \bigcirc(\neg holding(x))}$$

"never pick up *x* from table unless *x* needs to be on another block"

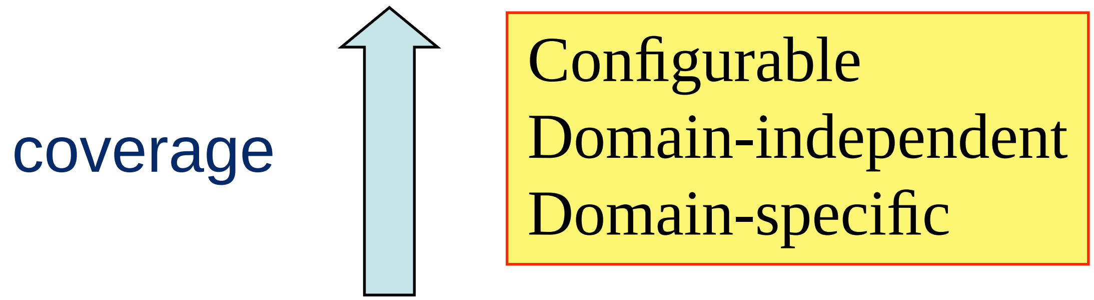
- For each successor of *s*, derive a control formula using *logical progression*
- Prune any successor state in which the progressed formula is false
    - TLPlan [Bacchus & Kabanza]
    - TALplanner [Kvarnstrom & Doherty]

Comparisons



- Domain-specific planner
  - Write an entire computer program - lots of work
  - Lots of domain-specific performance improvements
- Domain-independent planner
  - Just give it the basic actions - not much effort
  - Not very efficient

Comparisons



- A domain-specific planner only works in one domain
- **In principle**, configurable and domain-independent planners should both be able to work in any domain
- **In practice**, configurable planners work in a larger variety of domains
  - Partly due to efficiency
  - Partly due to expressive power

Example

- The planning competitions
  - All of them included domain-independent planners
- In addition, AIPS 2000 and *IPC* 2002 included configurable planners
- The configurable planners
  - Solved the most problems
  - Solved them the fastest
  - Usually found better solutions
  - Worked in many non-classical planning domains that were beyond the scope of the domain-independent planners

AIPS 1998 Planning Competition

AIPS 2000 Planning Competition

IPC 2002

IPC 2004

IPC 2006

But Wait …

- *IPC* 2004 and *IPC* 2006 included ***no*** configurable planners.
  - Why not?

AIPS 1998 Planning Competition

AIPS 2000 Planning Competition

IPC 2002

IPC 2004

IPC 2006

But Wait …

- *IPC* 2004 and *IPC* 2006 included ***no*** configurable planners.
  - Why not?
- Hard to enter them in the competition
  - Must write all the domain knowledge yourself
  - Too much trouble except to make a point
  - The authors of TLPlan, TALplanner, and SHOP2 felt they had already made their point

AIPS 1998 Planning Competition

AIPS 2000 Planning Competition

IPC 2002

IPC 2004

IPC 2006

But Wait …

- *IPC* 2004 and *IPC* 2006 included ***no*** configurable planners.
  - Why not?
- Hard to enter them in the competition
  - Must write all the domain knowledge yourself
  - Too much trouble except to make a point
  - The authors of TLPlan, TALplanner, and SHOP2 felt they had already made their point
- Why not provide the domain knowledge?

AIPS 1998 Planning Competition

AIPS 2000 Planning Competition

IPC 2002

IPC 2004

IPC 2006

But Wait …

- *IPC* 2004 and *IPC* 2006 included ***no*** configurable planners.
  - Why not?
- Hard to enter them in the competition
  - Must write all the domain knowledge yourself
  - Too much trouble except to make a point
  - The authors of TLPlan, TALplanner, and SHOP2 felt they had already made their point
- Why not provide the domain knowledge?
  - Drew McDermott proposed this at *ICAPS-05*
  - Many people didn't like this idea
    - » Cultural bias against it

AIPS 1998 Planning Competition

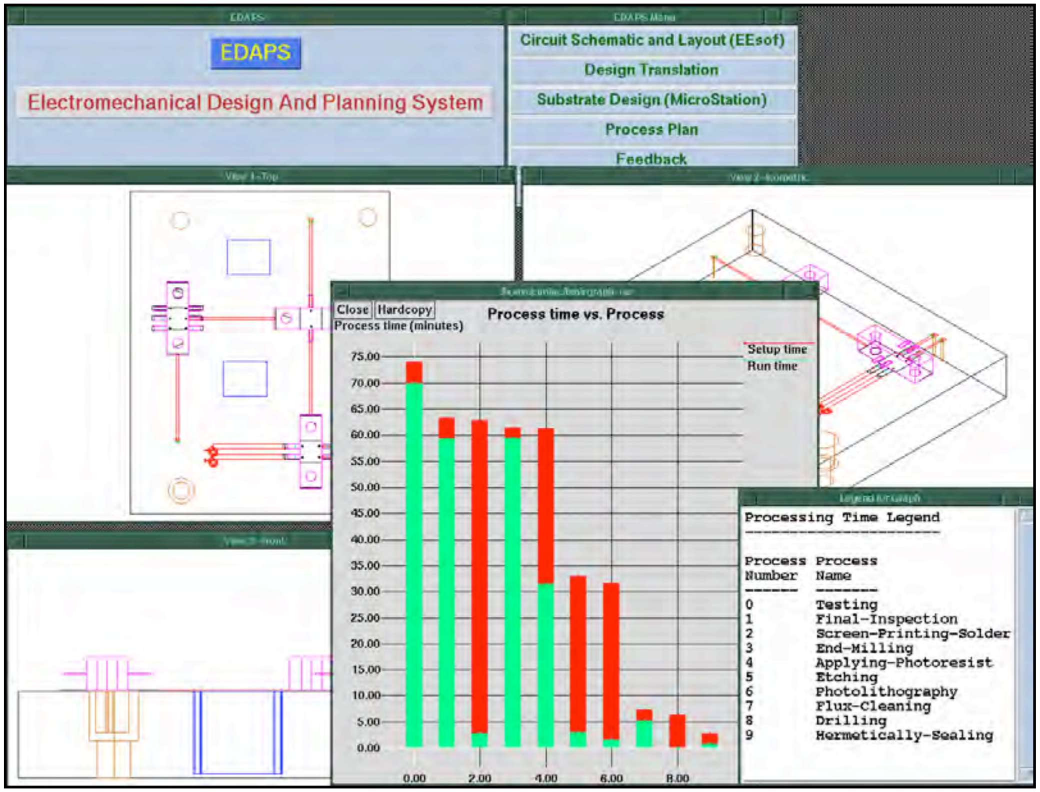
AIPS 2000 Planning Competition

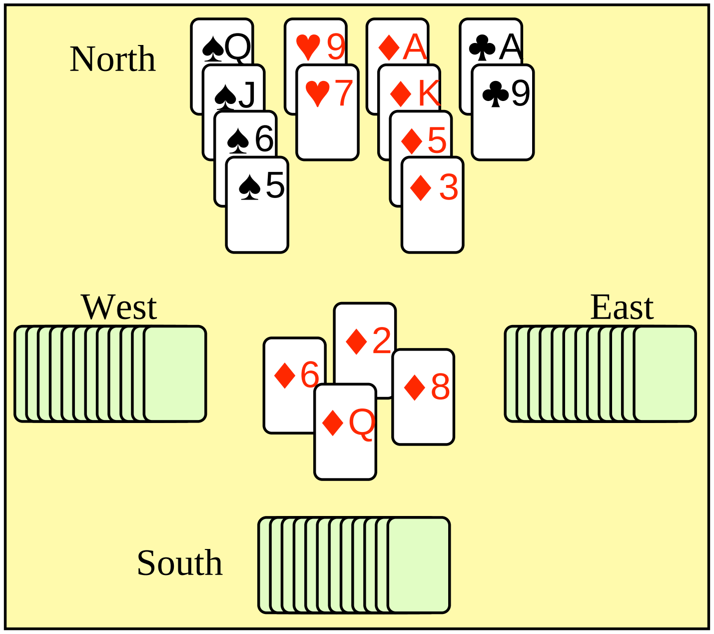
Cultural Bias

- Most automated-planning researchers feel that using domain knowledge is "cheating"
- Researchers in other fields have trouble comprehending this
  - Operations research, control theory, engineering, …
  - Why would anyone *not* want to use the knowledge they have about a problem they're trying to solve?
- In the past, the bias has been very useful
  - Without it, automated planning wouldn't have grown into a separate field from its potential application areas
- But it's less useful now
  - The field has matured
  - The bias is too restrictive

Example

- Typical characteristics of application domains
  - Dynamic world
  - Multiple agents
  - Imperfect/uncertain info
  - External info sources
    - » users, sensors, databases
  - Durations, time constraints, asynchronous actions
  - Numeric computations
    - » geometry, probability, etc.
- Classical planning excludes all of these





In Other Words …

- We **like** to think classical planning is domain-independent planning
- **But it isn't!**
  - Classical planning only includes domains that satisfy some **very** specific restrictions
  - Classical planners depend heavily on those restrictions
- This is fine for the **blocks world**
- **Not** so fine for the **real world**

Good News, Part 1

- We're already moving away from classical planning
- Example: the planning competitions
  - AIPS 1998, AIPS 2000, *IPC* 2002, *IPC* 2004
- Increasing divergence from classical planning
  - 1998, 2000: classical planning
  - 2002: added elementary notions of time durations, resources
  - 2004: added inference rules, derived effects, and a separate track for planning under uncertainty
  - 2006: added soft goals, trajectory constraints, preferences, plan metrics

AIPS 1998 Planning Competition

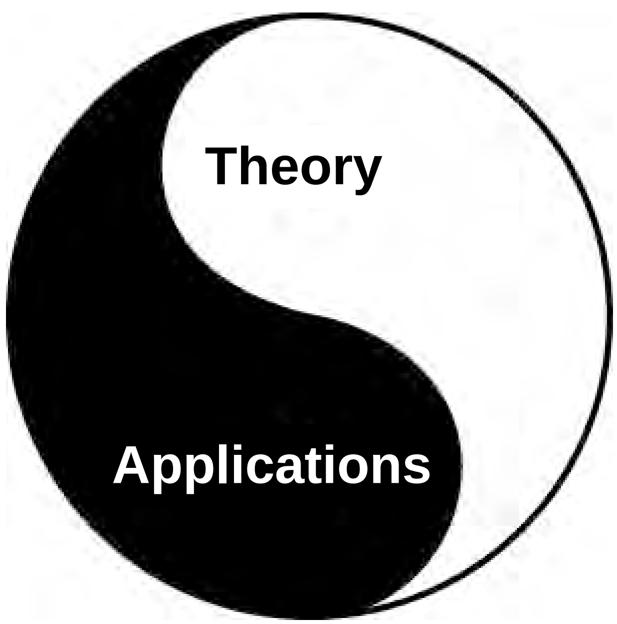
AIPS 2000 Planning Competition

Good News, Part 2

- Success in high-profile applications
  - A success like the Mars rovers is a big deal
  - Creates excitement about building planners that work in the real world

Good News, Part 3

- These successes provide opportunities for synergy between theory and practice
  - Understanding real-world planning leads to better theories
  - Better theories lead to better real-world planners

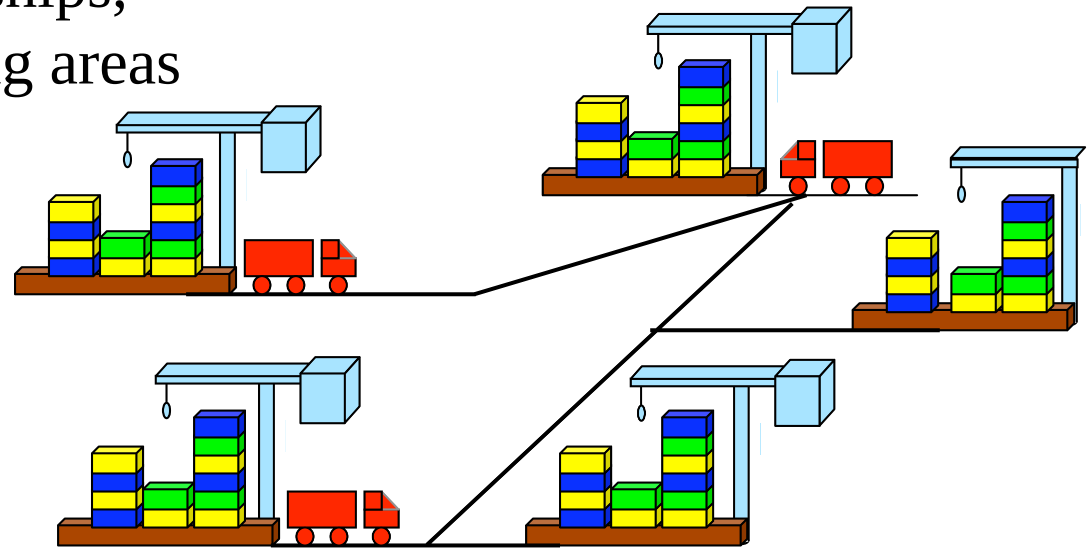


Good News, Part 4

- Classical planning research has produced some very powerful techniques for reducing the size of the search space
- We can generalize these techniques to work in non-classical domains
- Examples:
  - Partial order planning has been extended to do temporal planning
    - » Mars rovers
  - HTN planning has lots of applications
  - Classical planners can be extended to do planning under uncertainty
    - » I'll discuss this later in the semester if there's time

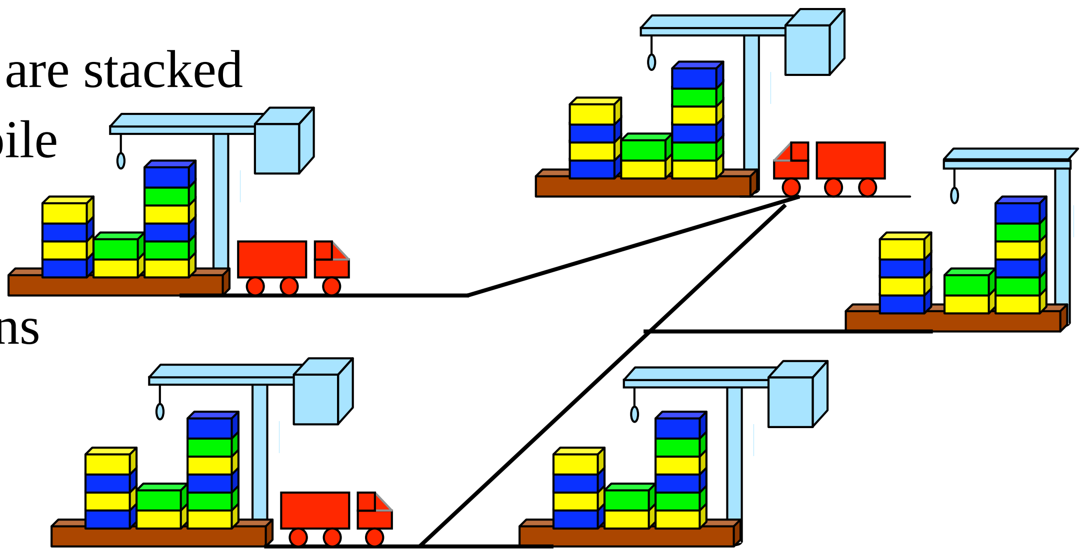
A running example: Dock Worker Robots

- Generalization of the earlier example
  - A harbor with several locations
    - » e.g., docks, docked ships, storage areas, parking areas
  - Containers
    - » going to/from ships
  - Robot carts
    - » can move containers
  - Cranes
    - » can load and unload containers



A running example: Dock Worker Robots

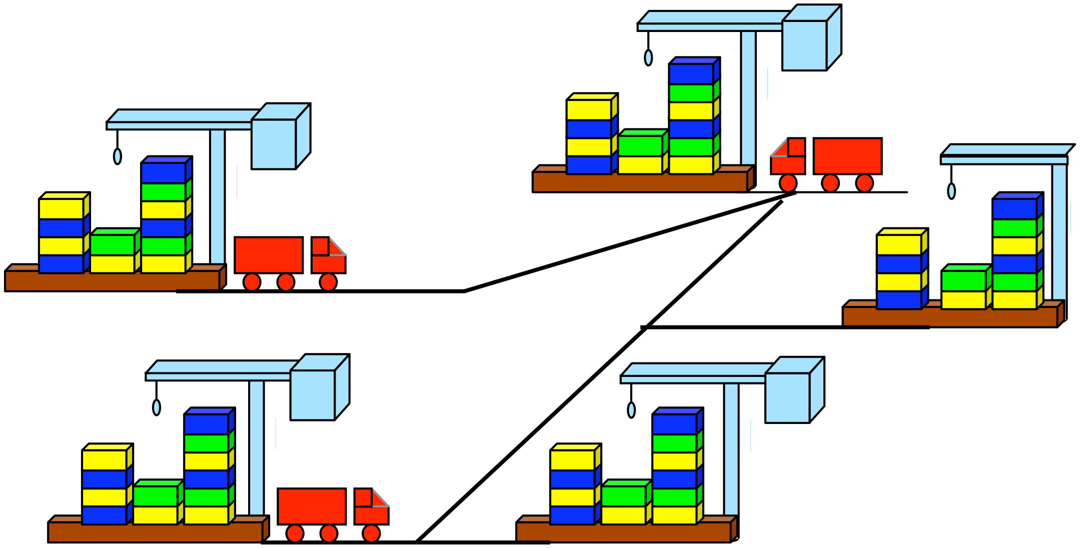
- **Locations**: l1, l2, …
- **Containers**: c1, c2, …
  - can be stacked in piles, loaded onto robots, or held by cranes
- **Piles**: p1, p2, …
  - fixed areas where containers are stacked
  - pallet at the bottom of each pile
- **Robot carts**: r1, r2, …
  - can move to adjacent locations
  - carry at most one container
- **Cranes**: k1, k2, …
  - each belongs to a single location
  - move containers between piles and robots
  - if there is a pile at a location, there must also be a crane there



A running example: Dock Worker Robots

- Fixed relations: same in all states
  adjacent(l,l')   attached(p,l)   belong(k,l)

- Dynamic relations: differ from one state to another
  occupied(l)   at(r,l)
  loaded(r,c)   unloaded(r)
  holding(k,c)   empty(k)
  in(c,p)   on(c,c')
  top(c,p)   top(pallet,p)



- Actions:
  take(c,k,p)   put(c,k,p)
  load(r,c,k)   unload(r)   move(r,l,l')

Planning the *free-flight* UAV

OPPA European Social Fund
Prague & EU: We invest in your future.