Principles of AI Planning

2. Transition systems

Malte Helmert

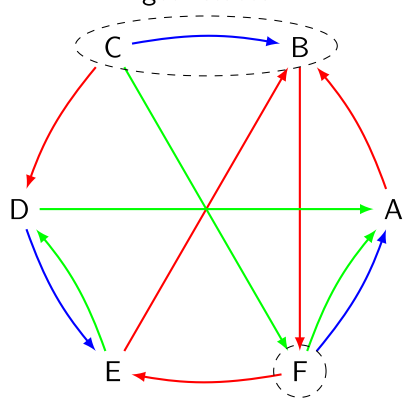
Albert-Ludwigs-Universität Freiburg

October 24th, 2008

Principles of AI Planning

October 24th, 2008 — 2. Transition systems

Transition systems



Transition systems

Formalization of the dynamics of the world/application

Definition (transition system)

A transition system is $\langle S, I, \{a_1, \dots, a_n\}, G\rangle$ where

- S is a finite set of states (the state space),
- $I \subseteq S$ is a finite set of initial states,
- every action $a_i \subseteq S \times S$ is a binary relation on S,
- $G \subseteq S$ is a finite set of goal states.

Definition (applicable action)

An action a is applicable in a state s if sas' for at least one state s'.

Transition systems

Deterministic transition systems

A transition system is deterministic if there is only one initial state and all actions are deterministic. Hence all future states of the world are completely predictable.

Definition (deterministic transition system)

A deterministic transition system is $\langle S, I, O, G \rangle$ where

- S is a finite set of states (the state space),
- $I \in S$ is a state,
- actions $a \in O$ (with $a \subseteq S \times S$) are partial functions,
- $G \subseteq S$ is a finite set of goal states.

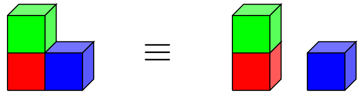
Successor state wrt. an action

Given a state s and an action a so that a is applicable in s, the successor state of s with respect to a is s' such that sas', denoted by $s' = app_a(s)$.

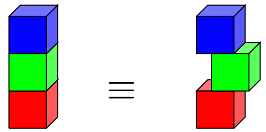
Blocks world

The rules of the game

Location on the table does not matter.



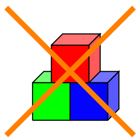
Location on a block does not matter.



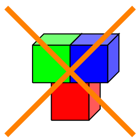
Blocks world

The rules of the game

At most one block may be below a block.

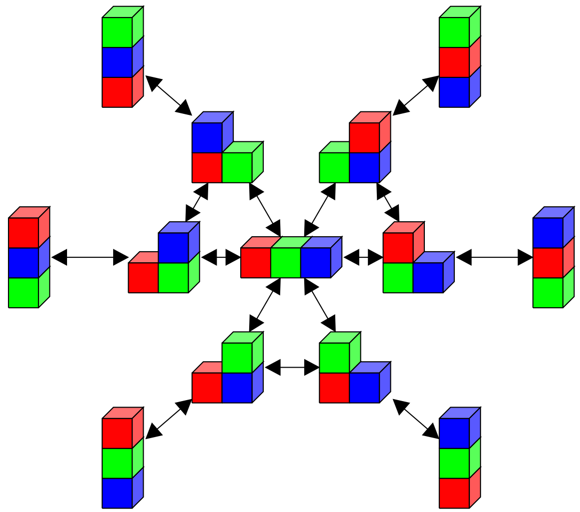


At most one block may be on top of a block.



Blocks world

The transition graph for three blocks



Blocks world

Properties

blocks	states
1	1
2	3
3	13
4	73
5	501
6	4051
7	37633
8	394353
9	4596553
10	58941091

1. Finding a solution is polynomial time in the number of blocks (move everything onto the table and then construct the goal configuration).
2. Finding a shortest solution is NP-complete (for a compact description of the problem).

Deterministic planning: plans

Definition (plan)

A plan for $\langle S, I, O, G\rangle$ is a sequence $\pi = o_1, \dots, o_n$ of operators such that $o_1, \dots, o_n \in O$ and $s_0, \dots, s_n$ is a sequence of states (the execution of π) so that

1. $s_0 = I$,
2. $s_i = app_{o_i}(s_{i-1})$ for every $i \in \{1, \dots, n\}$, and
3. $s_n \in G$.

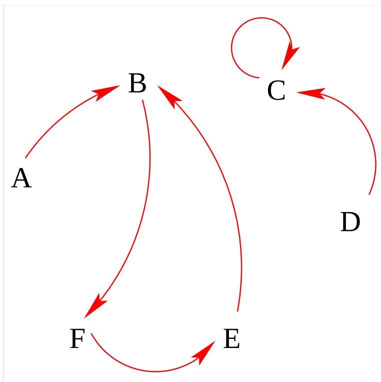
This can be equivalently expressed as

$$app_{o_n}(app_{o_{n-1}}(\dots app_{o_1}(I)\dots)) \in G$$

Transition relations as matrices

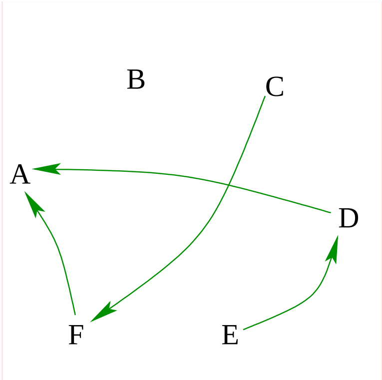
1. If there are n states, each action (a binary relation) corresponds to an $n \times n$ matrix: The element at row i and column j is 1 if the action maps state i to state j, and 0 otherwise.
   For deterministic actions there is at most one non-zero element in each row.
2. Matrix multiplication corresponds to sequential composition: taking action M_1 followed by action M_2 is the product $M_1 M_2$. (This also corresponds to the join of the relations.)
3. The unit matrix $I_{n \times n}$ is the NO-OP action: every state is mapped to itself.

Example



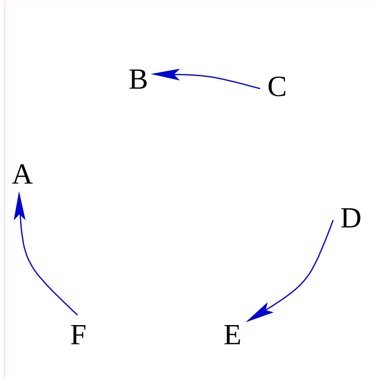
	A	B	C	D	E	F
A	0	1	0	0	0	0
B	0	0	0	0	0	1
C	0	0	1	0	0	0
D	0	0	1	0	0	0
E	0	1	0	0	0	0
F	0	0	0	0	1	0

Example



	A	B	C	D	E	F
A	0	0	0	0	0	0
B	0	0	0	0	0	0
C	0	0	0	0	0	1
D	1	0	0	0	0	0
E	0	0	0	1	0	0
F	1	0	0	0	0	0

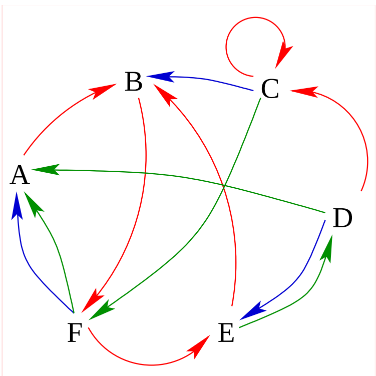
Example



	A	B	C	D	E	F
A	0	0	0	0	0	0
B	0	0	0	0	0	0
C	0	1	0	0	0	0
D	0	0	0	0	1	0
E	0	0	0	0	0	0
F	1	0	0	0	0	0

Sum matrix $M_R + M_G + M_B$

Representing one-step reachability by any of the component actions



	A	B	C	D	E	F
A	0	1	0	0	0	0
B	0	0	0	0	0	1
C	0	1	1	0	0	1
D	1	0	1	0	1	0
E	0	1	0	1	0	0
F	1	0	0	0	1	0

We use addition $0 + 0 = 0$ and $b + b' = 1$ if $b = 1$ or $b' = 1$.

Sequential composition as matrix multiplication

$$\begin{pmatrix} 0&1&0&0&0&0 \\ \hline 0&0&0&0&0&\mathbf{1} \\ \hline 0&1&1&0&0&1 \\ 1&0&1&0&1&0 \\ 0&1&0&1&0&0 \\ 1&0&0&0&1&0 \end{pmatrix} \times \left(\begin{array}{cccc|c|c} 0&1&0&0&0&0 \\ 0&0&0&0&0&1 \\ 0&1&1&0&0&1 \\ 1&0&1&0&1&0 \\ 0&1&0&1&0&0 \\ 1&0&0&0&\mathbf{1}&0 \end{array}\right) = \begin{pmatrix} 0&0&0&0&0&1 \\ 1&0&0&0&\mathbf{1}&0 \\ 1&1&1&0&1&1 \\ 0&1&1&1&0&0 \\ 0&0&1&0&1&1 \\ 0&1&0&1&0&0 \end{pmatrix}$$

E is reachable from B by two actions
because
F is reachable from B by one action and
E is reachable from F by one action.

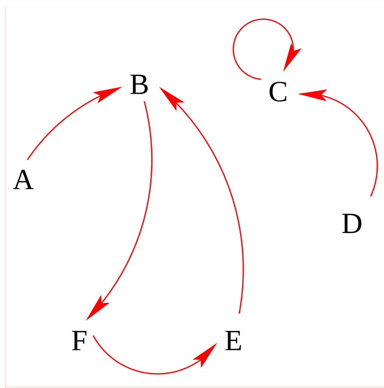
Reachability

Let M be the $n \times n$ matrix that is the (Boolean) sum of the matrices of the individual actions. Define

$$
\begin{aligned}
R_0 &= I_{n\times n} \\
R_1 &= I_{n\times n} + M \\
R_2 &= I_{n\times n} + M + M^2 \\
R_3 &= I_{n\times n} + M + M^2 + M^3 \\
&\vdots
\end{aligned}
$$

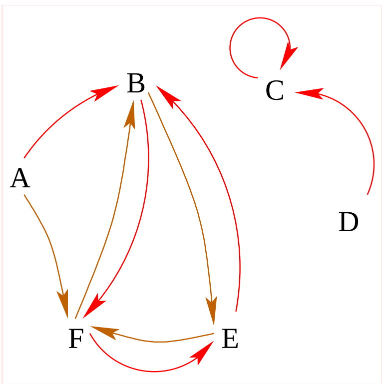
R_i represents reachability by i actions or less. If s' is reachable from s, then it is reachable with $\leq n-1$ actions: $R_{n-1} = R_n$.

Reachability: example, M_R



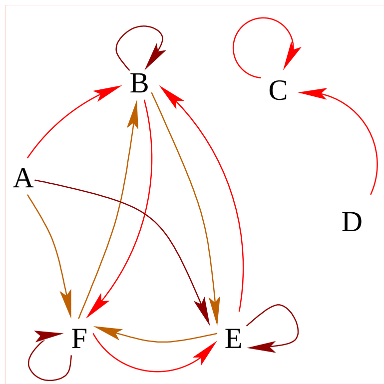
	A	B	C	D	E	F
A	0	1	0	0	0	0
B	0	0	0	0	0	1
C	0	0	1	0	0	0
D	0	0	1	0	0	0
E	0	1	0	0	0	0
F	0	0	0	0	1	0

Reachability: example, $M_R + M_R^2$



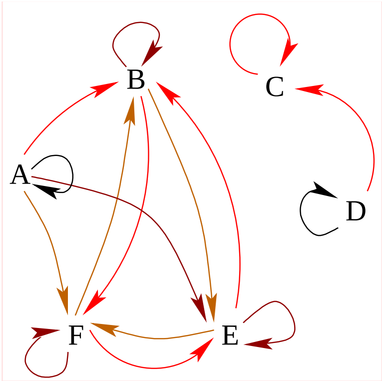
	A	B	C	D	E	F
A	0	1	0	0	0	1
B	0	0	0	0	1	1
C	0	0	1	0	0	0
D	0	0	1	0	0	0
E	0	1	0	0	0	1
F	0	1	0	0	1	0

Reachability: example, $M_R + M_R^2 + M_R^3$



	A	B	C	D	E	F
A	0	1	0	0	1	1
B	0	1	0	0	1	1
C	0	0	1	0	0	0
D	0	0	1	0	0	0
E	0	1	0	0	1	1
F	0	1	0	0	1	1

Reachability: example, $M_R + M_R^2 + M_R^3 + I_{6\times 6}$



	A	B	C	D	E	F
A	1	1	0	0	1	1
B	0	1	0	0	1	1
C	0	0	1	0	0	0
D	0	0	1	1	0	0
E	0	1	0	0	1	1
F	0	1	0	0	1	1

Relations and sets as matrices

Row vectors as sets of states

Row vectors S represent sets of states.
SM is the set of states reachable from S by M.

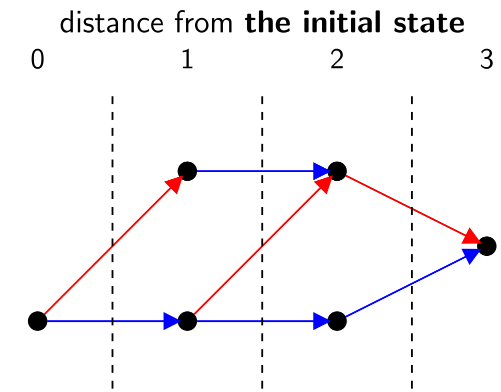
$$\begin{pmatrix}1\\0\\1\\0\\0\\0\end{pmatrix}^T \times \begin{pmatrix}1&1&0&0&1&1\\0&1&0&0&1&1\\0&0&1&0&0&0\\0&0&1&1&0&0\\0&1&0&0&1&1\\0&1&0&0&1&1\end{pmatrix} = \begin{pmatrix}1\\1\\1\\0\\1\\1\end{pmatrix}^T$$

A simple planning algorithm

- We next present a simple planning algorithm based on computing distances in the transition graph.
- The algorithm finds shortest paths less efficiently than Dijkstra's algorithm; we present the algorithm because we later will use it as a basis of an algorithm that is applicable to much bigger state spaces than Dijkstra's algorithm directly.

A simple planning algorithm

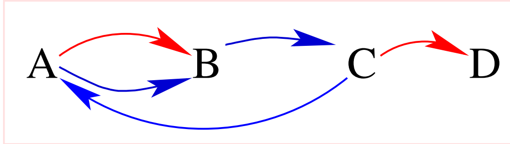
Idea



A simple planning algorithm

1. Compute the matrices $R_0, R_1, R_2, \ldots, R_n$ representing reachability with $0, 1, 2, \ldots, n$ steps with all actions.
2. Find the smallest i such that a goal state s_g is reachable from the initial state according to R_i.
3. Find an action (the last action of the plan) by which s_g is reached with one step from a state $s_{g'}$ that is reachable from the initial state according to R_{i-1}.
4. Repeat the last step, now viewing $s_{g'}$ as the goal state with distance $i - 1$.

Example



	A	B	C	D
A	0	1	0	0
B	0	0	0	0
C	0	0	0	1
D	0	0	0	0

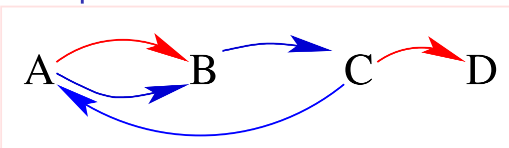
$+$

	A	B	C	D
A	0	1	0	0
B	0	0	1	0
C	1	0	0	0
D	0	0	0	0

$=$

	A	B	C	D
A	0	1	0	0
B	0	0	1	0
C	1	0	0	1
D	0	0	0	0

Example



$R_0 =$

	A	B	C	D
A	1	0	0	0
B	0	1	0	0
C	0	0	1	0
D	0	0	0	1

$R_1 =$

	A	B	C	D
A	1	1	0	0
B	0	1	1	0
C	1	0	1	1
D	0	0	0	1

$R_2 =$

	A	B	C	D
A	1	1	1	0
B	1	1	1	1
C	1	1	1	1
D	0	0	0	1

$R_3 =$

	A	B	C	D
A	1	1	1	1
B	1	1	1	1
C	1	1	1	1
D	0	0	0	1