## Determinization

another method for planning in non-deterministic domains

turn the non-deterministic problem into a deterministic one

employ a planner for *deterministic* problems
efficient ones exist

## determinizing a domain

nondeterministic action: two (or more) possible outcomes

turn it into two (or more) deterministic actions


## problem vs. determinized problem


original problem
- a may lead to 1 or 2
- nondeterminism chooses the outcome

determinized problem
- $a_1$ leads to 1
- $a_2$ leads to 2
- we choose the outcome

# planning in the determinization

weak solution = may reach the goal or not
"may" = depends on the outcome of the nondeterministic actions

if we could choose these outcome:
always reaches the goal

determinization = we can choose the outcome
choose $a_1$ or $a_2$


plan for the determinized problem = weak solution for the original problem

# planning by determinizing

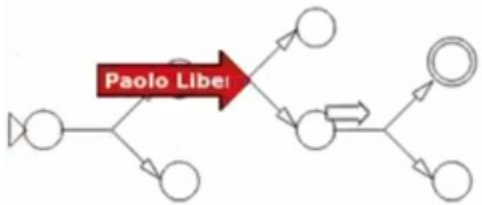
a step more:

weak solution =
    plan for the determinized problem
strong cyclic solution =
    weak solution that is also a weak solution for every state reachable by the policy

find a weak solution (by determinizing)
also find a weak solution for all other states reachable by the policy

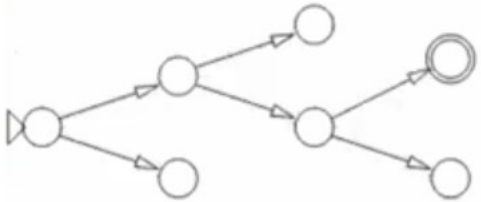
## determinization: example (1)

original problem



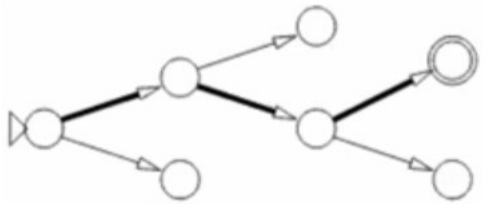

## determinization: example (2)

determinized problem



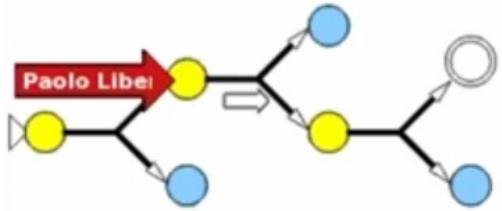
## determinization: example (3)

find a plan
(this is a deterministic problem)



## determinization: example (4)

carry the plan back to the original problem




weak solution

not a strong cyclic solution
no plan for the states in the frontier

**repeat for the states in the frontier**

for each (cyan states), find a weak solution
how: again, a plan in the determinization

# find a strong cyclic solution by determinization

summary:

- find a plan in the determinization
- carry it back to the original problem
- for each state in the frontier:
  find a plan in the determinization, etc.

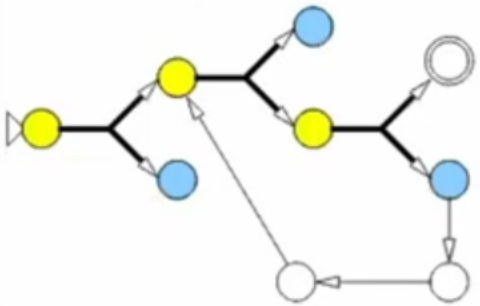
missing:

1. simplified example, tree-like
2. goal reachable from all states

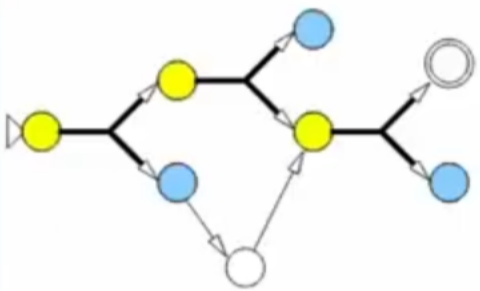
# cycles, convergence

two cases to keep into account

case 1: the deterministic plan from a state in the frontier leads back to a state in the domain (yellow):



case 2: the plan leads to another state in the domain (yellow):



neither is a problem:
states in the domain (yellow) already have a weak solution
the rest of the deterministic plan can be ignored

## cycles in the deterministic plan

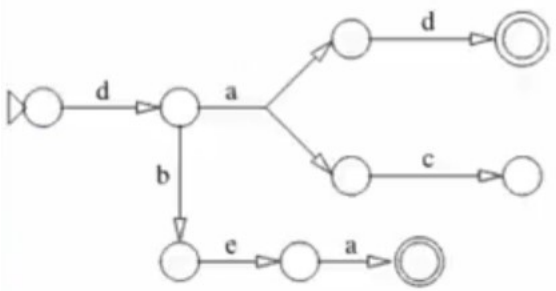
cut the cycles off

## dead ends

states where the goal is unreached

a strong (cyclic) policy never reaches them

## dead ends (1)

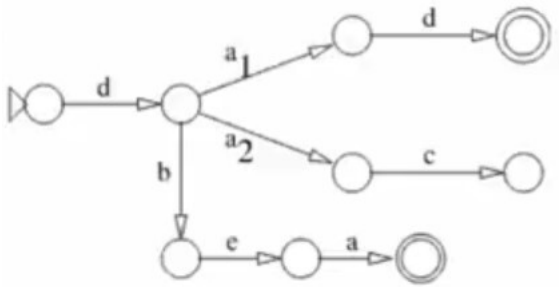



best policy is to execute `d,b,e,a`

planning by determinization may not realize it

## dead ends (2)

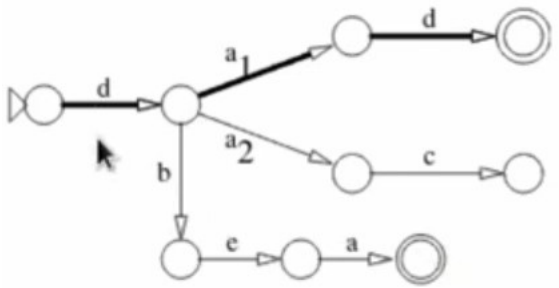
first, determinize:




## dead ends (3)

find a plan

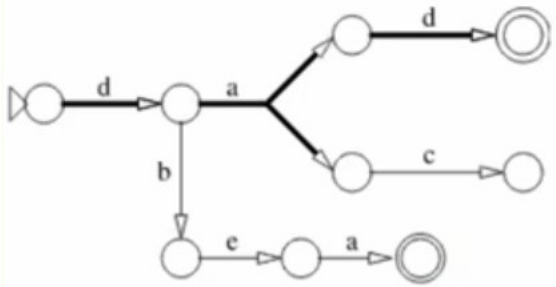
the shortest plan is:




plan `d,a1,d` has length 3
the other plan `d,b,e,a` has length 4

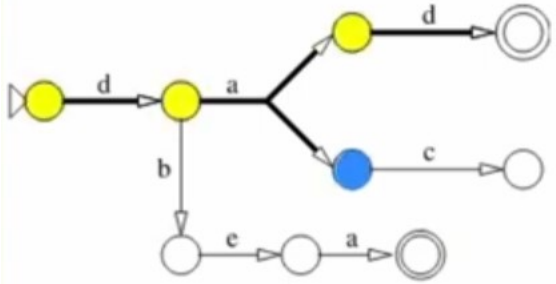
# dead ends (4)

carry the plan back to the nondeterministic problem:



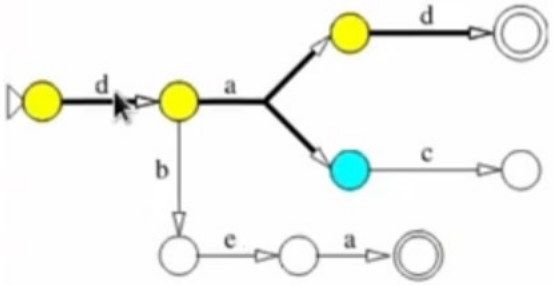

# dead ends (5)

determine the new frontier:




goal is unreachable from the cyan state!
this policy can never be turned into a strong solution

# avoid the dead ends




goal cannot be reached from state
⇓
never go the state

not just for this policy
do not go there, period

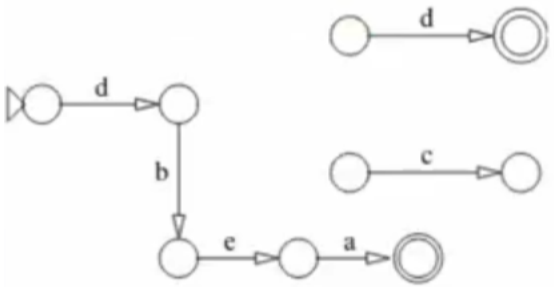
# cutting before the dead ends

action a in state s leads to a dead end

⇓

make a inexecutable in s

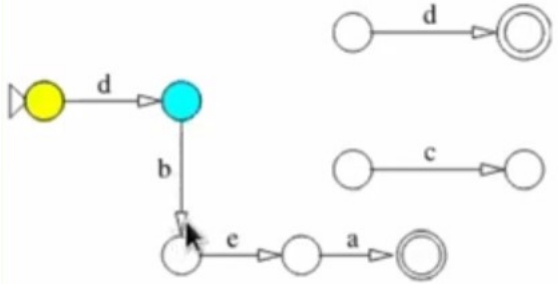
problem after the cut




note: a still executable in the other states with the same effects

# after the cut

recompute the frontier




continue: for each node in the frontier:

- search for a plan in the determinization
- carry it back to the nondeterministic problem
- …

# planning by determinization: summary

principle:

- a strong (cyclic) solution is a weak solution for every state reachable by the policy
- a weak solution can be found by determinizing

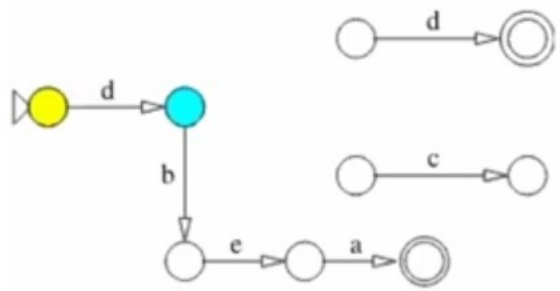
implementation:

- at the beginning, the frontier comprises the initial state only
- for each state in the frontier:
  - search for a plan in the determinization
  - if no plan exists, cut all actions leading to the state
  - otherwise, add the actions of the plan to the policy
  - in both cases, recompute the frontier

# the roads not taken

planning by determinization may look like it only adds actions
what about the roads not taken?
(= the actions not chosen)

in fact, it may consider them
after cutting actions from states, the frontier retreats
the algorithm has to find other paths from the states just before the cut




action b was neglected before
now is the only choice