Planning as a CSP

- In forward planning, the nodes considered are constrained to be reachable, even if they don't lead to goal.

Planning as a CSP

- In forward planning, the nodes considered are constrained to be reachable, even if they don't lead to goal.
- In regression planning, the nodes considered are constrained to be ones from which we can achieve the goal, even if they are not reachable.

Planning as a CSP

- In forward planning, the nodes considered are constrained to be reachable, even if they don't lead to goal.
- In regression planning, the nodes considered are constrained to be ones from which we can achieve the goal, even if they are not reachable.
- When representing planning as a CSP, we can constrain the states by the starting state and the goal.

Planning as a CSP

- In forward planning, the nodes considered are constrained to be reachable, even if they don't lead to goal.
- In regression planning, the nodes considered are constrained to be ones from which we can achieve the goal, even if they are not reachable.
- When representing planning as a CSP, we can constrain the states by the starting state and the goal.
  . . . but we can only do this if we know the number of steps.

Planning as a CSP

- In forward planning, the nodes considered are constrained to be reachable, even if they don't lead to goal.
- In regression planning, the nodes considered are constrained to be ones from which we can achieve the goal, even if they are not reachable.
- When representing planning as a CSP, we can constrain the states by the starting state and the goal.
  . . . but we can only do this if we know the number of steps.
- Search over planning horizons (number of time steps).

Planning as a CSP

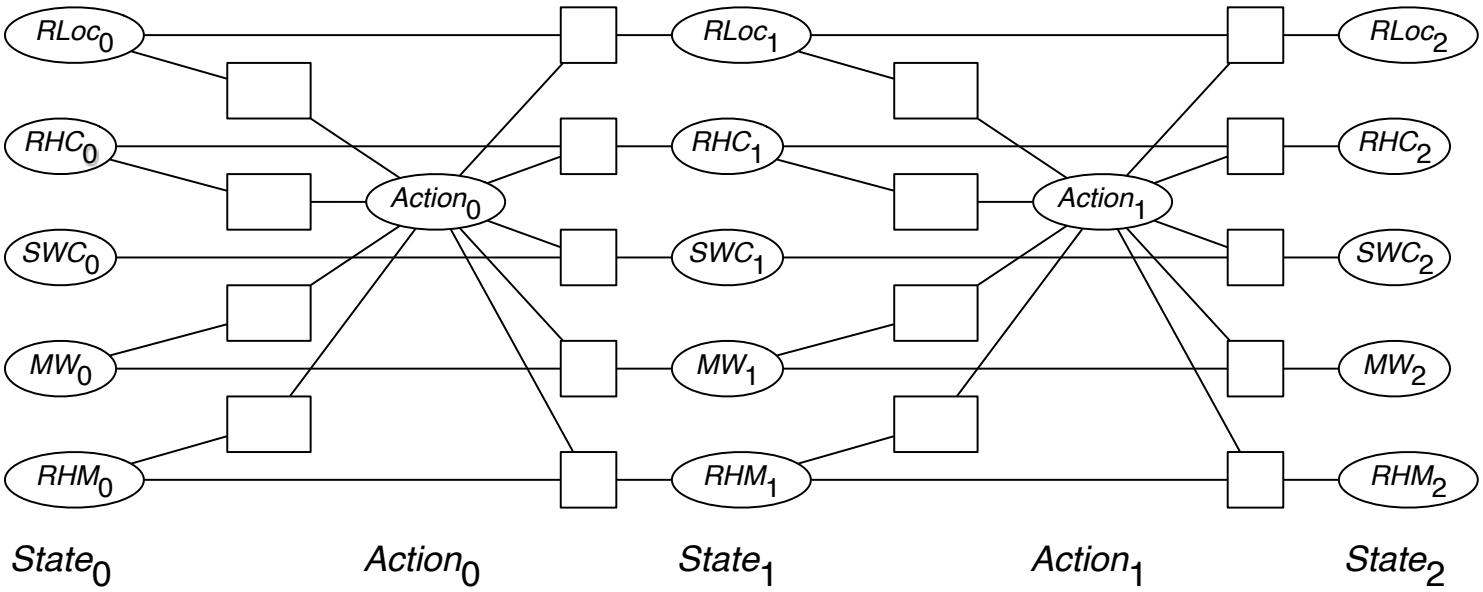
- In forward planning, the nodes considered are constrained to be reachable, even if they don't lead to goal.
- In regression planning, the nodes considered are constrained to be ones from which we can achieve the goal, even if they are not reachable.
- When representing planning as a CSP, we can constrain the states by the starting state and the goal.
  . . . but we can only do this if we know the number of steps.
- Search over planning horizons (number of time steps).
- For each planning horizon, create a CSP constraining possible actions and features

CSP Variables

Choose a planning horizon k.

- Create a variable for each state feature and each time from 0 to k.
- Create a variable for the action for each time in the range 0 to $k-1$.

CSP for Delivery Robot for a planning horizon of 2



$RLoc_i$ — Rob's location
RHC_i — Rob has coffee
SWC_i — Sam wants coffee
MW_i — Mail is waiting
RHM_i — Rob has mail

$Move_i$ — Rob's move action
PUC_i — Rob picks up coffee
$DelC$ — Rob delivers coffee
PUM_i — Rob picks up mail
$DelM_i$ — Rob delivers mail

Constraints

- precondition constraints between state variables at time t and action variable at time t, specify constraints on what actions are available from a state.

Constraints

- precondition constraints between state variables at time t and action variable at time t, specify constraints on what actions are available from a state.
- effect constraints between state variables at time t, action variable at time t and state variables at time $t+1$ constrain the resulting state to be one that satisfies the effects.

Constraints

- precondition constraints between state variables at time t and action variable at time t, specify constraints on what actions are available from a state.
- effect constraints between state variables at time t, action variable at time t and state variables at time $t+1$ constrain the resulting state to be one that satisfies the effects.
- frame constraints among state variables at time t, action variables at time t, and state variables at time $t+1$ for values of variables that do not change.

Constraints

- precondition constraints between state variables at time t and action variable at time t, specify constraints on what actions are available from a state.
- effect constraints between state variables at time t, action variable at time t and state variables at time $t+1$ constrain the resulting state to be one that satisfies the effects.
- frame constraints among state variables at time t, action variables at time t, and state variables at time $t+1$ for values of variables that do not change.
- initial state constraints that are usually domain constraints on the initial state (at time 0).
- goal constraints that constrains the final state to be a state that satisfies the goals that are to be achieved.

Example Constraints

precondition constraint $RLoc_i = cs$ if $Action_i = puc$
is violated when

Example Constraints

precondition constraint $RLoc_i = cs$ if $Action_i = puc$
is violated when $RLoc_i \neq cs \wedge Action_i = puc$

Example Constraints

precondition constraint $RLoc_i = cs$ if $Action_i = puc$
is violated when $RLoc_i \neq cs \wedge Action_i = puc$

effect constraint rhc_{i+1} if $Act_i = puc$
is violated when

Example Constraints

precondition constraint $RLoc_i = cs$ if $Action_i = puc$
is violated when $RLoc_i \neq cs \wedge Action_i = puc$

effect constraint rhc_{i+1} if $Act_i = puc$
is violated when $RHC_{i+1} = false \wedge Act_i = puc$.

Example Constraints

precondition constraint $RLoc_i = cs$ if $Action_i = puc$
is violated when $RLoc_i \neq cs \wedge Action_i = puc$

effect constraint rhc_{i+1} if $Act_i = puc$
is violated when $RHC_{i+1} = false \wedge Act_i = puc$.

frame constraint Rob has mail at any time if it had mail before and the action wasn't

Example Constraints

precondition constraint $RLoc_i = cs$ if $Action_i = puc$
is violated when $RLoc_i \neq cs \wedge Action_i = puc$

effect constraint rhc_{i+1} if $Act_i = puc$
is violated when $RHC_{i+1} = false \wedge Act_i = puc$.

frame constraint Rob has mail at any time if it had mail before and the action wasn't to pickup mail or deliver mail:

$$RHM_{i+1} = RHM_i \text{ if } Act_i \notin \{pum, dm\}$$

violated when

Example Constraints

precondition constraint $RLoc_i = cs$ if $Action_i = puc$
is violated when $RLoc_i \neq cs \wedge Action_i = puc$

effect constraint rhc_{i+1} if $Act_i = puc$
is violated when $RHC_{i+1} = false \wedge Act_i = puc$.

frame constraint Rob has mail at any time if it had mail before and the action wasn't to pickup mail or deliver mail:

$$RHM_{i+1} = RHM_i \text{ if } Act_i \notin \{pum, dm\}$$

violated when
$RHM_{i+1} \neq RHM_i \wedge Act_i \neq pum \wedge Act_i \neq dm$