Module Title: MODULE TITLE
Exam Diet (Dec/April/Aug): EXAM DIET 2004
Brief notes on answers:

1. (a) **AI Planning**: Planning is an area that has been researched in Artificial Intelligence for a long time now. What do we mean by "planning"? What is being studied in "AI planning"? Why do people usually only plan when this is necessary? [*5 marks*]

   Answer:

   - "planning":
     - explicit deliberation process that chooses and organizes actions by anticipating their outcomes (reasoning about actions)
     - aims at achieving some pre-stated objectives
   - "AI planning":
     - computational study of this deliberation process
   - Humans only plan when necessary because planning is complicated and time-consuming (trade-off: cost vs. benefit).

   (b) **State-Space Search**: Consider the planning domain and problem defined in appendix 1. The first step in the forward state-space search algorithm computes the set of applicable actions. What are the applicable actions for the given initial state? [*5 marks*]

   Answer:

   - `(op1 C C A )`
   - `(op1 C B A )`
   - `(op2 C C C )`
   - `(op2 A C C )`
   - `(op2 B C C )`
   - `(op2 C C A )`
   - `(op2 A C A )`
   - `(op2 B C A )`

   Note that there are other instances of `op1` that have their preconditions satisfied in the initial state. However, they have inconsistent effects.

   (c) **Hierarchical Planning**: When planning with task networks, the aim is not to achieve some goal state, but to accomplish a given set of tasks. Describe, in pseudo-code, the ground partial-order forward decomposition algorithm (Ground-PFD) that takes as input an initial state s, an initial task network $w = (U, E)$, a set of primitive operators O, and a set of methods M. [*5 marks*]

   Answer:

function Ground-PFD(s, w, O, M)

if $w.U = \emptyset$ return $\langle\rangle$

$task \leftarrow \{t \in U | t$ has no predecessors in $w.E\}$.chooseOne()

if $task$.isPrimitive() then

$actions \leftarrow \{(a, \sigma) | a = \sigma(task)$ and a applicable in $s\}$

if $actions$.isEmpty() then return failure

$(a, \sigma) \leftarrow actions$.chooseOne()

$plan \leftarrow$ Ground-PFD($\gamma(s, a), \sigma(w - \{task\}), O, M$)

if $plan =$ failure then return failure

else return $\langle a \rangle \bullet plan$

else

$methods \leftarrow \{(m, \sigma) | \sigma(m)$ is relevant for $task$ and m is applicable in $s\}$

if $methods$.isEmpty() then return failure

$(m, \sigma) \leftarrow methods$.chooseOne()

return Ground-PFD($s, \delta(w, task, m, \sigma), O, M$)

(d) **Graphplan**: The planning problem in appendix 1 can be translated into a propositional problem by grounding it, i.e. by replacing all variables with all possible combinations of objects defined in the problem. This results in 12 proposition symbols for states (`SA`, `SB`, `SC`, `RAA`, `RAB`, ..., `RCB`, `RCC`) and up to 54 possible actions (`op1AAA`, `op1AAB`, ..., `op2CCB`, `op2CCC`). Which of the following actions are (pairwise) independent?

- `op1ABC`
- `op1CBA`
- `op2CBA`

[*5 marks*]

Answer:

- `op1ABC` and `op1CBA` are dependent (not independent) because `op1ABC` deletes `SB` which is a precondition of `op1CBA`
- `op1ABC` and `op2CBA` are dependent (not independent) because `op2CBA` deletes `SC` which is also a positive effect of `op1ABC`
- `op1CBA` and `op2CBA` are dependent (not independent) because `op1CBA` deletes `SC` which is a precondition of `op2CBA`

(e) **Scheduling**: To apply local neighbourhood search to job shop scheduling problems, a schedule can be encoded as a sequence of action-machine pairs. Consider the following scheduling problem:

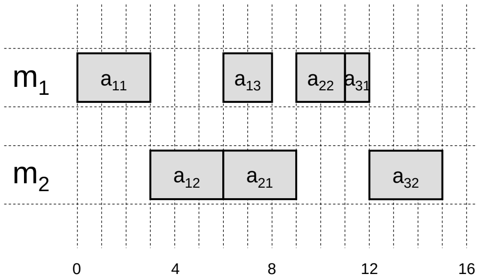
- $j_1 : \langle r_1(3), r_2(3), r_1(2) \rangle$
- $j_2 : \langle r_2(3), r_1(2) \rangle$
- $j_3 : \langle r_1(1), r_2(3) \rangle$

What schedule is encoded by the following sequence of action-machine pairs, given two machines m_1 and m_2 of resource types r_1 and r_2 respectively, and using the earliest assignable times to create the schedule?

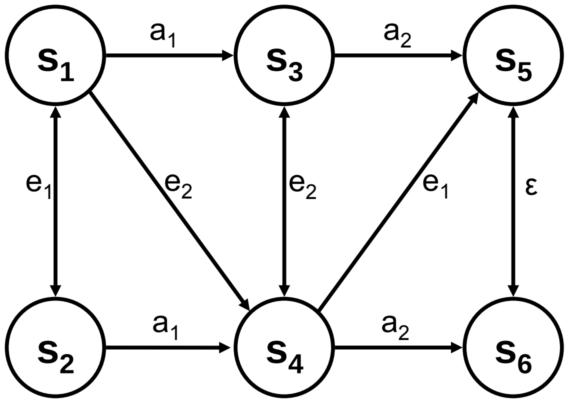
$$\langle (a_{11}, m_1)(a_{12}, m_2)(a_{21}, m_2)(a_{13}, m_1)(a_{22}, m_1)(a_{31}, m_1)(a_{32}, m_2) \rangle$$

[*5 marks*]

Answer:


m1
m2
a11
a13
a22
a31
a12
a21
a32
0
4
8
12
16

2. (a) **AI Planning**: The following graph represents a state-transition system $\Sigma = (S, A, E, \gamma)$ where a_1 and a_2 denote actions, e_1 and e_2 denote events, and ε denotes no event or action taking place. Define this state-transition system formally. Is this system deterministic? What state will the system be in after the action and event sequence $\langle a_1 e_2 a_2 \rangle$ assuming the initial state is s_1?



[*5 marks*]
Answer:

- $S = \{s_1, s_2, s_3, s_4, s_5, s_6\}$
- $A = \{a_1, a_2\}$
- $E = \{e_1, e_2\}$
-

γ	a_1	a_2	e_1	e_2	ε
s_1	$\{s_3\}$	$\{\}$	$\{s_2\}$	$\{s_4\}$	$\{\}$
s_2	$\{s_4\}$	$\{\}$	$\{s_1\}$	$\{\}$	$\{\}$
s_3	$\{\}$	$\{s_5\}$	$\{\}$	$\{s_4\}$	$\{\}$
s_4	$\{\}$	$\{s_6\}$	$\{s_5\}$	$\{s_3\}$	$\{\}$
s_5	$\{\}$	$\{\}$	$\{\}$	$\{\}$	$\{s_6\}$
s_6	$\{\}$	$\{\}$	$\{\}$	$\{\}$	$\{s_5\}$

The system is effectively non-deterministic because of the ε transition. $\langle a_1 e_2 a_2 \rangle$ will get the system into s_5 or s_6.

(b) **State-Space Search**: In state-space search the nodes in the search space are world states. Consider the planning domain and problem defined in appendix 1. What state will the world be in after execution of the following plan π in the initial state defined in the problem?

$\pi = \langle$ `(op1 C B A)`, `(op1 A A C)`, `(op2 C C C)` $\rangle$

[*5 marks*]
Answer:

- apply: `(op1 C B A)`
- result: { `(S A) (R A C) (R C A) (R C C)` }
- apply: `(op1 A A C)`
- result: { `(S C) (R A C) (R C A) (R C C)` }
- apply: `(op2 C C C)`

- result: { `(R A C) (R C A) (R C C)` }

(c) **Plan-Space Search**: In plan-space search the nodes in the search space are partial-order plans which contain explicit causal links between the different actions in the plan. Thus, planners that perform a plan-space search must find the new threats in a partial plan when the plan is refined. For which types of refinement do the threats need to be detected? For each of these describe in pseudo-code how the detection is performed. [*5 marks*]

Answer:

- in the initial plan π_0: no threats
- when adding an action a_{new} to $\pi = (A, \prec, B, L)$:

  for every causal link $\langle a_i \xrightarrow{p} a_j \rangle \in L$
  if ($a_{new} \prec a_i$) or ($a_j \prec a_{new}$) then next link
  else for every effect q of a_{new}
  if ($\exists\sigma : \sigma(p) = \sigma(\neg q)$) then q of a_{new} threatens $\langle a_i \xrightarrow{p} a_j \rangle$

- when adding a causal link $\langle a_i \xrightarrow{p} a_j \rangle$ to $\pi = (A, \prec, B, L)$:

  for every action $a_{old} \in A$
  if ($a_{old} \prec a_i$) or ($a_j = a_{old}$) or ($a_j \prec a_{old}$) then next action
  else for every effect q of a_{old}
  if ($\exists\sigma : \sigma(p) = \sigma(\neg q)$) then q of a_{old} threatens $\langle a_i \xrightarrow{p} a_j \rangle$

(d) **Graphplan**: The planning graph developed by the Graphplan planner quickly grows from layer to layer until the proposition layers contain all those propositions that are eventually achievable. However, propositions that occur in the same layer may not be achievable simultaneously. Define these so-called mutex relations for propositions and for actions. [*5 marks*]

Answer:

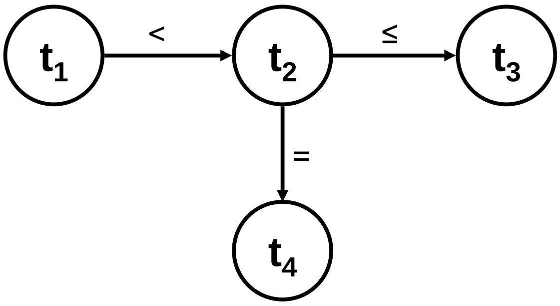
Two propositions p and q in the jth proposition layer P_j are mutex (mutually exclusive) if:

- every action in the preceding action layer A_j that has p as a positive effect (incl. no-op actions) is mutex with every action in A_j that has q as a positive effect, and
- there is no single action in A_j that has both, p and q, as positive effects.

Two actions a_1 and a_2 in action layer A_j are mutex if:

- a_1 and a_2 are dependent, or
- a precondition of a_1 is mutex with a precondition of a_2.

(e) **Temporal Planning**: The following graph represents a time point network consisting of four time points $t_1 \ldots t_4$ and three explicit relations between these time points. What are the implied constraints in this network? How can we tell that this network is consistent?



[*5 marks*]

Answer:

The following table lists all the relations in the time point network. Each cell contains the relation R that must hold between time point x and y, i.e. xRy must hold.

$x \backslash y$	t_1	t_2	t_3	t_4
t_1	$=$	$<$	$<$	$<$
t_2	$>$	$=$	$\leq$	$=$
t_3	$>$	$\geq$	$=$	$\geq$
t_4	$>$	$=$	$\leq$	$=$

The network is consistent because no two time points are related by the empty set (of primitive relations).

3. (a) **Situation Calculus**: In the situation calculus actions are denoted by function terms. The definition of these functions is given by a set of axioms, namely the applicability axiom, the effect axiom, and the frame axioms. Define these axioms for the action `op1` defined as part of the domain in appendix 1. [*5 marks*]

Answer:

- applicability:
  $\forall x_1, x_2, x_3, s : \mathit{applicable}(op1(x_1, x_2, x_3), s) \leftrightarrow S(x_1, s) \wedge S(x_2, s) \wedge R(x_3, X_1, s)$
- effect axiom:
  $\forall x_1, x_2, x_3, s : \mathit{applicable}(op1(x_1, x_2, x_3), s) \rightarrow S(x_3, \mathit{result}(op1(x_1, x_2, x_3), s)) \wedge R(x_1, x_3, \mathit{result}(op1(x_1, x_2, x_3), s))$
- frame axiom:
  $\forall x', x_1, x_2, x_3, s : S(x', s) \wedge (x' \neq x_1 \vee x' \neq x_2) \rightarrow S(x', \mathit{result}(op1(x_1, x_2, x_3), s))$
  $\forall x', y', x_1, x_2, x_3, s : R(x', y', s) \rightarrow R(x', y', \mathit{result}(op1(x_1, x_2, x_3), s))$

There is no need for frame axioms that carry forward negative conditions as all preconditions are positive in this domain.

(b) **State-Space Search**: In classical planning, a planning problem is solved by searching for a solution plan. Define, in pseudo-code, the non-deterministic ground backward state-space search algorithm for a given statement of a STRIPS planning problem $P = (O, s_i, g)$. [*5 marks*]

Answer:

function $\mathit{groundBwdSearch}(O, s_i, g)$
  $\mathit{subgoal} \leftarrow g$
  $\mathit{plan} \leftarrow \langle\rangle$
  loop
    if $s_i.\mathit{satisfies}(\mathit{subgoal})$ then return plan
    $\mathit{applicables} \leftarrow$ {ground instances from O relevant for $\mathit{subgoal}$}
    if $\mathit{applicables.isEmpty}()$ then return failure
    $\mathit{action} \leftarrow \mathit{applicables.chooseOne}()$
    $\mathit{subgoal} \leftarrow \gamma^{-1}(\mathit{subgoal}, \mathit{action})$
    $\mathit{plan} \leftarrow \langle \mathit{action} \rangle \bullet \mathit{plan}$

(c) **Plan-Space Search**: In plan-space search the nodes in the search space are partial plans that are refined until a solution plan is found. Partial plans that are not solutions may contain threats. Construct an example that explains the concept of a threat using the operators given in the planning domain defined in appendix 1. [*5 marks*]

Answer:

- partially instantiated actions in the plan:
  - `(op1 ?x1 ?x2 A)`
  - `(op1 A ?x1 ?x2)`
  - `(op2 A ?x1 ?x2)`
- with `(op1 ?x1 ?x2 A)` before `(op1 A ?x1 ?x2)`
- and causal link `(op1 ?x1 ?x2 A)` $\xrightarrow{(SA)}$ `(op1 A ?x1 ?x2)`

Then the effect `(not (S A))` of `(op2 A ?x1 ?x2)` constitutes a threat to the causal link. That is, if this operator was executed between the two connected by the causal link, it would undo the condition protected by the causal link.

(d) **SAT-Based Planning**: SAT-based planning is quite similar to the idea used in the situation calculus: the planning problem is reformulated as a theorem proving problem. Show how the actions of a propositional planning problem $P = (A, s_i, g)$ can be represented as part of the SAT encoding of a bounded planning problem, i.e. what are the propositional formulas that encode the preconditions, effects and frame axioms? [*5 marks*]

Answer:

Let F (the fluents) be the set of all the proposition symbols that occur in $P = (A, s_i, g)$ (in preconditions, positive and negative effects, in the initial state, or in the goal) and let n be the length of the plan sought (bounded planning problem). Then the actions can be represented by formulas for every $a \in A$ and $0 \leq i \leq n-1$:

$a_i \rightarrow (\bigwedge_{f \in precond(a)} f_i) \wedge \bigwedge_{f \in effects^+(a)} f_{i+1}) \wedge \bigwedge_{f \in effects^-(a)} \neg f_{i+1})$

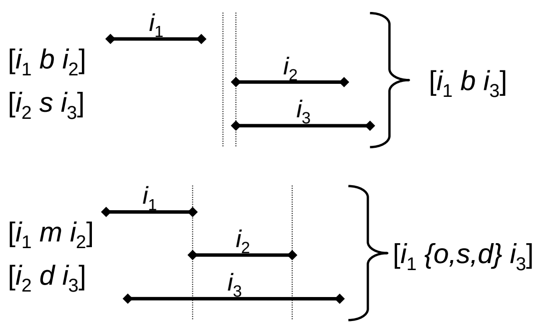
and the frame axioms can be represented (using explanation closure axioms) by formulas for every fluent $f \in F$ and $0 \leq i \leq n-1$:

$(f_i \wedge \neg f_{i+1}) \rightarrow (\bigvee_{a \in A \wedge f \in effects^-(a)} a_i) \wedge (\neg f_i \wedge f_{i+1}) \rightarrow (\bigvee_{a \in A \wedge f \in effects^+(a)} a_i)$

(e) **Temporal Planning**: In the Interval Algebra there are 13 different primitive relations that can relate two intervals. These can be combined using the composition operator ($\bullet$). For example, $(b \bullet b)$ can be simplified to b. What relations are expressed by $(b \bullet s)$ and $(m \bullet d)$? Explain your answers. [*5 marks*]

Answer:

- $i_1(b \bullet s)i_3$ can be simplified to $i_1 b i_3$ (b = before, s = starts)
- $i_1(m \bullet d)i_3$ can be simplified to $i_1\{o, s, d\}i_3$ (m = meets, d = during, o = overlaps)



Appendix 1

The following is definition of a planning domain and a statement of a planning problem in PDDL:

```
(define (domain random-domain)
  (:requirements :strips)

  (:action op1
    :parameters (?x1 ?x2 ?x3)
    :precondition
      (and (S ?x1) (S ?x2) (R ?x3 ?x1))
    :effect (and
      (S ?x3) (R ?x1 ?x3)
      (not (S ?x1)) (not (S ?x2))))
  (:action op2
    :parameters (?x1 ?x2 ?x3)
    :precondition
      (and  (S ?x1)(R ?x3 ?x2))
    :effect (and
      (R ?x3 ?x1) (R ?x3 ?x3)
      (not (S ?x1)))))
)


(define (problem problem1)
  (:domain random-domain)

  (:init
    (S A) (S B) (S C)
    (R A C) (R C C))

  (:goal
    (and (R C B) (R B A)))
)
```