1 a Consider the following program Π:

$$
\begin{aligned}
p &\leftarrow \text{not } q, r. \\
p &\leftarrow \text{not } r. \\
q &\leftarrow \text{not } r. \\
r &\leftarrow \text{not } q. \\
s &\leftarrow \text{not } p, r. \\
s &\leftarrow p, s.
\end{aligned}
$$

Consider the sets:

$$
\begin{aligned}
S_1 &= \{\} \\
S_2 &= \{p\} \\
S_3 &= \{p, q\} \\
S_4 &= \{p, r, s\} \\
S_5 &= \{p, q, r, s\}
\end{aligned}
$$

i) For each set S_i, state whether or not it is a *model* of Π. If S_i is not a model of Π, provide a rule in Π that is violated by S_i.

ii) Which of the models in (i) are *supported*? For each model that is unsupported, briefly explain why.

iii) Which, if any, of the supported models in (ii) are *unfounded*? Briefly justify your answer.

b i) Provide a *splitting set* for Π that is non-trivial (i.e. it is neither the empty set $\{\}$ nor the set of all atoms $\{p, q, r, s\}$).

ii) List all the stable models of Π.

c Let Π be a ground normal logic program. Let X and Y be sets of ground atoms. Assume $X \subset Y$ and that Y is a stable model of Π. Show that there is a clause γ in Π such that $X \not\models \gamma$.

The three parts carry, respectively, 40%, 30%, and 30% of the marks.

2 a The *set cover* problem is a classic NP-complete problem. We are given a universe U of objects, and a collection S of subsets of U. The task is to find the smallest number of sets in S that together cover all of U.

For example, let $U = \{1,2,3,4,5\}$, and $S = \{\{1,2,3\},\{2,4\},\{3,4\},\{4,5\}\}$. Here, $\{1,2,3\}$ and $\{4,5\}$ together cover U.

You are given the following predicates:

- $universe(E)$ means that element E is part of the universe U
- $set(X)$ means that set X is in set S
- $in(E,X)$ means that element E is in set X

The example above can be represented using the atoms:

$universe(1)$	$universe(2)$	$universe(3)$	$universe(4)$	$universe(5)$
$set(s_1)$	$set(s_2)$	$set(s_3)$	$set(s_4)$	
$in(1,s_1)$	$in(2,s_1)$	$in(3,s_1)$		
$in(2,s_2)$	$in(4,s_2)$			
$in(3,s_3)$	$in(4,s_3)$			
$in(4,s_4)$	$in(5,s_4)$			

Here, $s_1 = \{1,2,3\}, s_2 = \{2,4\}, s_3 = \{3,4\}, s_4 = \{4,5\}$.

i) Write an ASP program defining a predicate $include(X)$, indicating that set X should be included in the collection of sets that cover U. Each answer set of your program should represent one possible solution to the set covering problem. For example, one answer set of your program should contain the atoms $includes(s_1)$ and $includes(s_4)$, since these two sets together cover U.

ii) Modify your ASP program so that it finds the *smallest number of sets* that together cover U.

b Consider the following:

- r_1: students usually get up late
- r_2: athletes usually get up early
- r_3: sick people usually get up late
- r_2 is an exception to r_1
- r_3 is an exception to r_2

i) Express these defeasible rules, and the exceptions between them, as an extended logic program. Be sure to express the incompatibility between "getting up early" and "getting up late".

ii) Suppose Alice is a student and also an athlete. Suppose Bob is an athlete, but he is also sick. What does your program conclude (in terms of *cautious* entailment) about Alice and Bob?

If you remove the exceptions between the rules, what does your program conclude (in terms of *cautious* entailment) about Alice and Bob? How many answer sets are there altogether, if you remove the exceptions between the rules?

The two parts carry equal marks.

3 a Suppose the following $\mathcal{C}+$ action signature $(\sigma^{\mathrm{f}}, \sigma^{\mathrm{a}})$:

$$\begin{aligned} \sigma^{\mathrm{f}} &= \{p, q\} \\ \sigma^{\mathrm{a}} &= \{a, b\} \\ dom(q) &= \{1,2,3\} \\ dom(b) &= \{1,2,3\} \end{aligned}$$

Domains are Boolean unless explicitly stated to be different.

For each of the following, state: *first*, which category of causal law (static, action dynamic, or fluent dynamic) it is—or if none, give a reason; and *secondly*, whether it could occur in a definite action description, and if not, give a reason.

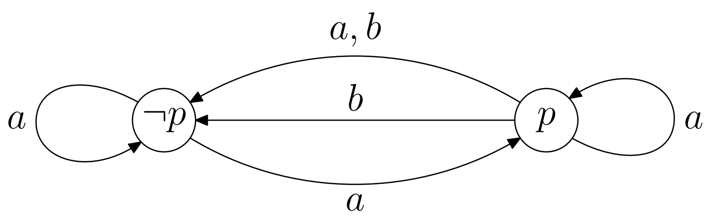
i) **default** p **after** $p \wedge (q = 1)$

ii) $(b = 1) \wedge q$ **may cause** $\top$ **if** $\neg p$

b Let $(\sigma^{\mathrm{f}}, \sigma^{\mathrm{a}})$ be the signature:

$$\begin{aligned} \sigma^{\mathrm{f}} &= \{p\} \\ \sigma^{\mathrm{a}} &= \{a, b\} \end{aligned}$$

All constants are Boolean. Write down a $\mathcal{C}+$ action description D which defines the following LTS:



c Consider the following knowledge, K:

> Two people, Alice and Bob, can each hold or put down either end of a plank. They can also do this to the same end.

Let $(\sigma^{\mathrm{f}}, \sigma^{\mathrm{a}})$ be the following $\mathcal{C}+$ action signature. (P is $\{alice, bob\}$, and E is $\{left, right\}$.)

$$\begin{aligned} \sigma^{\mathrm{f}} &= \{holds(P)\} & (P \in \mathsf{P}) \\ \sigma^{\mathrm{a}} &= \{take(P), release(P)\} & (P \in \mathsf{P}) \\ dom(c) &= \mathsf{E} \cup \{\mathbf{f}\} & (c \in \sigma^{\mathrm{f}} \cup \sigma^{\mathrm{a}}) \end{aligned}$$

Here, for $P \in \mathsf{P}$ and $E \in \mathsf{E}$:

$holds(P) = E$ means P is holding end E (left or right) of the plank;
$take(P) = E$ means that P picks up end E of the plank;
$release(P) = E$ means that P drops end E of the plank.

Let D be the following causal laws:

inertial $holds(P)$	$(P \in \mathsf{P})$
exogenous f	$(f \in \sigma^{a})$
$take(P) = E$ **causes** $holds(P) = E$	$(P \in \mathsf{P}; E \in \mathsf{E})$
nonexecutable $take(P) = E$ **if** $holds(P) = E'$	$(P \in \mathsf{P}; E, E' \in \mathsf{E}, E \neq E')$
$release(P) = E$ **causes** $holds(P) = \mathbf{f}$	$(P \in \mathsf{P}; E \in \mathsf{E})$
nonexecutable $release(P) = E$ **if** $\neg(holds(P) = E)$	$(P \in \mathsf{P}; E \in \mathsf{E})$

D formalizes part of the knowledge above.

i) Suppose the following is learned about the domain:

> A cup can be placed on the plank by Alice or Bob and will stay there, unless the plank is crooked. (It's crooked if only one end is held.)

Specify additions to (σ^{f}, σ^{a}) and D which formalize the new knowledge.

ii) Add further additions, supplementing those of (i), to formalize the following:

> The wind can blow the cup off the plank.

The three parts carry, respectively, 20%, 35%, and 45% of the marks.

4 a Assume Cn is a classical consequence operator; let K be a set of propositional formulas, containing background knowledge. Define Cn_K as follows (where A is any set of propositional formulas):

$$\mathrm{Cn}_K(A) =_{\mathrm{df}} \mathrm{Cn}(A \cup K)$$

Prove that Cn_K satisfies Monotony and Closure.

b Let $\mathit{models}(A)$ be the set of all interpretations in which every member of A is true. Note that $\alpha \in \mathrm{Th}(A)$ iff $\mathit{models}(A) \subseteq \mathit{models}(\{\alpha\})$.

Let a consequence relation $\mathrm{Cn}_{\mathrm{pref}}$ be such that $B \subseteq \mathrm{Cn}_{\mathrm{pref}}(A)$ iff $\mathit{models}_{\mathrm{pref}}(A) \subseteq \mathit{models}(B)$, where $\mathit{models}_{\mathrm{pref}}(A)$ represents a special 'preferred' subset of the models of A. (So $\mathit{models}_{\mathrm{pref}}(A) \subseteq \mathit{models}(A)$.)

Prove that $\mathrm{Cn}_{\mathrm{pref}}$ is supraclassical.

c A consequence operator Cn' (not necessarily classical) is said to be Cumulative if it satisfies the following property:

if $A \subseteq B \subseteq \mathrm{Cn}'(A)$ then $\mathrm{Cn}'(A) = \mathrm{Cn}'(B)$

A consequence relation Cn' (again, not necessarily classical) is said to be Reciprocal if it satisfies:

if $A \subseteq \mathrm{Cn}'(B)$ and $B \subseteq \mathrm{Cn}'(A)$ then $\mathrm{Cn}'(A) = \mathrm{Cn}'(B)$

Prove that, given Inclusion, then Cn' is Cumulative if and only if it is Reciprocal.

The three parts carry, respectively, 45%, 15%, and 40% of the marks.

1 a Consider the following program Π:

$$
\begin{aligned}
p &\leftarrow \text{not } q, r. \\
p &\leftarrow \text{not } r. \\
q &\leftarrow \text{not } r. \\
r &\leftarrow \text{not } q. \\
s &\leftarrow \text{not } p, r. \\
s &\leftarrow p, s.
\end{aligned}
$$

Consider the sets:

$$
\begin{aligned}
S_1 &= \{\} \\
S_2 &= \{p\} \\
S_3 &= \{p, q\} \\
S_4 &= \{p, r, s\} \\
S_5 &= \{p, q, r, s\}
\end{aligned}
$$

i) For each set S_i, state whether or not it is a *model* of Π. If S_i is not a model of Π, provide a rule in Π that is violated by S_i.

S_1 is not a model, since it violates $p \leftarrow \text{not } r$. S_2 is not a model, since it violates $q \leftarrow \text{not } r$. S_3 is a model. S_4 is a model. S_5 is a model.

Marks: 3

ii) Which of the models in (i) are *supported*? For each model that is unsupported, briefly explain why.

S_3 and S_4 are supported since, for both models, every atom is justified by some rule in Π that is satisfied in the model. S_5 is not supported since atom p lacks support.

Marks: 3

iii) Which, if any, of the supported models in (ii) are *unfounded*? Briefly justify your answer.

S_4 is unfounded. Let $U = \{s\}$ in the definition of an unfounded set. Then for each rule with s in the head, either S_4 does not satisfy the body, or s appears in the (positive) body.

Marks: 2

b i) Provide a *splitting set* for Π that is non-trivial (i.e. it is neither the empty set $\{\}$ nor the set of all atoms $\{p, q, r, s\}$).

$\{q, r\}$ is a suitable splitting set.

Marks: **2**

ii) List all the stable models of Π.

The stable models are $\{p, q\}$ and $\{p, r\}$.

Marks: **4**

c Let Π be a ground normal logic program. Let X and Y be sets of ground atoms. Assume $X \subset Y$ and that Y is a stable model of Π. Show that there is a clause γ in Π such that $X \not\models \gamma$.

Since Y is a stable model of Π, Y is a subset minimal model of Π. Hence, since $X \subset Y$, X is not a model of Π. Therefore, there is some clause γ such that $X \not\models \gamma$.

To show that stable models are subset minimal: assume Y is a stable model of Π, assume $Y' \subseteq Y$ and $Y' \models \Pi$. We must show $Y' = Y$:

- *Since $Y' \subseteq Y$, $\Pi^Y \subseteq \Pi^{Y'}$.*
- *Since $Y' \models \Pi$, $Y' \models \Pi^{Y'}$.*
- *Hence, since $\Pi^Y \subseteq \Pi^{Y'}$, $Y' \models \Pi^Y$.*
- *Since Y is a stable model of Π, $Y = M(\Pi^Y)$.*
- *Now, by definition, $M(\Pi^Y)$ is the* least *model of Π^Y.*
- *Hence $Y \subseteq Y'$.*
- *Hence $Y = Y'$.*

Marks: **6**

The three parts carry, respectively, 40%, 30%, and 30% of the marks.

2a The *set cover* problem is a classic NP-complete problem. We are given a universe U of objects, and a collection S of subsets of U. The task is to find the smallest number of sets in S that together cover all of U.

For example, let $U = \{1,2,3,4,5\}$, and $S = \{\{1,2,3\},\{2,4\},\{3,4\},\{4,5\}\}$. Here, $\{1,2,3\}$ and $\{4,5\}$ together cover U.

You are given the following predicates:

- $universe(E)$ means that element E is part of the universe U
- $set(X)$ means that set X is in set S
- $in(E,X)$ means that element E is in set X

The example above can be represented using the atoms:

$universe(1)$	$universe(2)$	$universe(3)$	$universe(4)$	$universe(5)$
$set(s_1)$	$set(s_2)$	$set(s_3)$	$set(s_4)$	
$in(1,s_1)$	$in(2,s_1)$	$in(3,s_1)$		
$in(2,s_2)$	$in(4,s_2)$			
$in(3,s_3)$	$in(4,s_3)$			
$in(4,s_4)$	$in(5,s_4)$			

Here, $s_1 = \{1,2,3\}, s_2 = \{2,4\}, s_3 = \{3,4\}, s_4 = \{4,5\}$.

i) Write an ASP program defining a predicate $include(X)$, indicating that set X should be included in the collection of sets that cover U. Each answer set of your program should represent one possible solution to the set covering problem. For example, one answer set of your program should contain the atoms $includes(s_1)$ and $includes(s_4)$, since these two sets together cover U.

One line to choose which sets to include:

```
{include(S)} :- set(S).
```

One line to define which elements are covered:

```
covered(E) :- include(S), in(E, S).
```

One line to constrain solutions to those in which every element in the universe is covered:

```
:- universe(E), not covered(E).
```

Marks: **8**

ii) Modify your ASP program so that it finds the *smallest number of sets* that together cover U.

Add one optimization statement:

```
:~ include(S). [1 @ 1, S]
```

Marks: **2**

b Consider the following:

- r_1: students usually get up late
- r_2: athletes usually get up early
- r_3: sick people usually get up late
- r_2 is an exception to r_1
- r_3 is an exception to r_2

i) Express these defeasible rules, and the exceptions between them, as an extended logic program. Be sure to express the incompatibility between "getting up early" and "getting up late".

$$\neg late(X) \leftarrow early(X).$$
$$\neg early(X) \leftarrow late(X).$$

$$sat(r_1(X)) \leftarrow student(X).$$
$$fires(r_1(X)) \leftarrow sat(r_1(X)), \text{not } \neg late(X), \text{not } \neg fires(r_1(X)).$$
$$late(X) \leftarrow fires(r_1(X)).$$

$$sat(r_2(X)) \leftarrow athlete(X).$$
$$fires(r_2(X)) \leftarrow sat(r_2(X)), \text{not } \neg early(X), \text{not } \neg fires(r_2(X)).$$
$$early(X) \leftarrow fires(r_2(X)).$$

$$sat(r_3(X)) \leftarrow sick(X).$$
$$fires(r_3(X)) \leftarrow sat(r_3(X)), \text{not } \neg late(X), \text{not } \neg fires(r_3(X)).$$
$$late(X) \leftarrow fires(r_3(X)).$$

$$\neg fires(r_1(X)) \leftarrow sat(r_2(X))$$
$$\neg fires(r_2(X)) \leftarrow sat(r_3(X))$$

Marks: **8**

ii) Suppose Alice is a student and also an athlete. Suppose Bob is an athlete, but he is also sick. What does your program conclude (in terms of *cautious* entailment) about Alice and Bob?

If you remove the exceptions between the rules, what does your program conclude (in terms of *cautious* entailment) about Alice and Bob? How many answer sets are there altogether, if you remove the exceptions between the rules?

Initially, it concludes that Alice gets up early, and that Bob gets up late. If you remove the exceptions between the rules, the program concludes nothing about Alice or Bob. Without the exceptions between the rules, there are four answer sets.

Marks: **2**

The two parts carry equal marks.

3 a Suppose the following $\mathcal{C}+$ action signature $(\sigma^{\mathrm{f}}, \sigma^{\mathrm{a}})$:

$$\begin{aligned} \sigma^{\mathrm{f}} &= \{p, q\} \\ \sigma^{\mathrm{a}} &= \{a, b\} \\ dom(q) &= \{1,2,3\} \\ dom(b) &= \{1,2,3\} \end{aligned}$$

Domains are Boolean unless explicitly stated to be different.

For each of the following, state: *first*, which category of causal law (static, action dynamic, or fluent dynamic) it is—or if none, give a reason; and *secondly*, whether it could occur in a definite action description, and if not, give a reason.

i) **default** p **after** $p \wedge (q = 1)$

ii) $(b = 1) \wedge q$ **may cause** $\top$ **if** $\neg p$

i) *This is not a causal law—the* **after** *would have to be an* **if**.

ii) *This is not a causal law, as* $(b = 1) \vee q$ *is not an action formula—it contains the fluent constant* q.

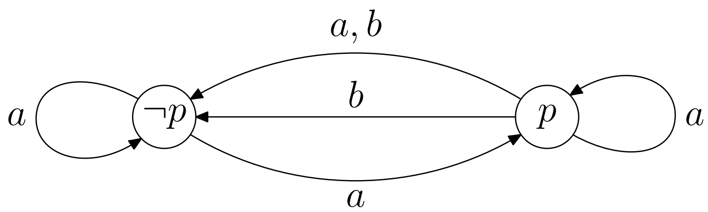
[Marking scheme*: two marks per part—one for the ans., one for the reason.]*

Marks: $\overline{4}$

b Let $(\sigma^{\mathrm{f}}, \sigma^{\mathrm{a}})$ be the signature:

$$\begin{aligned} \sigma^{\mathrm{f}} &= \{p\} \\ \sigma^{\mathrm{a}} &= \{a, b\} \end{aligned}$$

All constants are Boolean. Write down a $\mathcal{C}+$ action description D which defines the following LTS:



inertial p
exogenous a
exogenous b
a ***may cause*** p ***if*** $\neg p$
b ***causes*** $\neg p$ ***if*** p
nonexecutable $\neg a \wedge \neg b$
nonexecutable b ***if*** $\neg p$

Marks: $\overline{\mathbf{7}}$

c Consider the following knowledge, K:

> Two people, Alice and Bob, can each hold or put down either end of a plank. They can also do this to the same end.

Let $(\sigma^{\mathrm{f}}, \sigma^{\mathrm{a}})$ be the following $\mathcal{C}+$ action signature. (P is $\{alice, bob\}$, and E is $\{left, right\}$.)

$$\sigma^{\mathrm{f}} = \{holds(P)\} \qquad (P \in \mathsf{P})$$
$$\sigma^{\mathrm{a}} = \{take(P), release(P)\} \qquad (P \in \mathsf{P})$$
$$dom(c) = \mathsf{E} \cup \{\mathbf{f}\} \qquad (c \in \sigma^{\mathrm{f}} \cup \sigma^{\mathrm{a}})$$

Here, for $P \in \mathsf{P}$ and $E \in \mathsf{E}$:

> $holds(P) = E$ means P is holding end E (left or right) of the plank;
> $take(P) = E$ means that P picks up end E of the plank;
> $release(P) = E$ means that P drops end E of the plank.

Let D be the following causal laws:

inertial $holds(P)$ $\qquad (P \in \mathsf{P})$
exogenous f $\qquad (f \in \sigma^{\mathrm{a}})$
$take(P) = E$ **causes** $holds(P) = E$ $\qquad (P \in \mathsf{P}; E \in \mathsf{E})$
nonexecutable $take(P) = E$ **if** $holds(P) = E'$ $\qquad (P \in \mathsf{P}; E, E' \in \mathsf{E}, E \neq E')$
$release(P) = E$ **causes** $holds(P) = \mathbf{f}$ $\qquad (P \in \mathsf{P}; E \in \mathsf{E})$
nonexecutable $release(P) = E$ **if** $\neg(holds(P) = E)$ $\qquad (P \in \mathsf{P}; E \in \mathsf{E})$

D formalizes part of the knowledge above.

i) Suppose the following is learned about the domain:

> A cup can be placed on the plank by Alice or Bob and will stay there, unless the plank is crooked. (It's crooked if only one end is held.)

Specify additions to $(\sigma^{\mathrm{f}}, \sigma^{\mathrm{a}})$ and D which formalize the new knowledge.

ii) Add further additions, supplementing those of (i), to formalize the following:

> The wind can blow the cup off the plank.

---

i) *One way: add a single Boolean fluent constant cup_on to* σ^{f}, *and an action constant putcup with* $dom(putcup) = \mathsf{P} \cup \{\boldsymbol{f}\}$. *The laws:*

$$\textbf{inertial } cup\_on$$
$$\textbf{exogenous } putcup$$
$$putcup = P \textbf{ causes } cup\_on \qquad (P \in \mathsf{P})$$
$$\textbf{caused } \neg cup\_on \textbf{ if } holds(P) = E \wedge \neg(holds(P') = E')$$
$$(P, P \in \mathsf{P}, P \neq P; E, E' \in \mathsf{E}, E \neq E')$$
$$\textbf{nonexecutable } putcup = P \textbf{ if } cup\_on \qquad (P \in \mathsf{P})$$

ii) *Add a further, Boolean action constant blow_off, and the laws:*

$$\textbf{exogenous } blow\_off$$
$$blow\_off \textbf{ causes } \neg cup\_on \textbf{ if } cup\_on$$
$$\textbf{nonexecutable } blow\_off \textbf{ if } \neg cup\_on$$

*[*Marking scheme*: six marks for the first part; three for the second.]*

Marks: $\overline{\mathbf{9}}$

The three parts carry, respectively, 20%, 35%, and 45% of the marks.

4 a Assume Cn is a classical consequence operator; let K be a set of propositional formulas, containing background knowledge. Define Cn_K as follows (where A is any set of propositional formulas):

$$\mathrm{Cn}_K(A) =_{\mathrm{df}} \mathrm{Cn}(A \cup K)$$

Prove that Cn_K satisfies Monotony and Closure.

There are many ways to do this. Here's one.

To be classical Cn must satisfy:

Inclusion	$A \subseteq Cn(A)$
Closure	$Cn(Cn(A)) \subseteq Cn(A)$
Monotony	*if* $A \subseteq B$ *then* $Cn(A) \subseteq Cn(B)$

For Monotony:
We have to show that if $A \subseteq B$, then $Cn_K(A) \subseteq Cn_K(B)$. Clearly, if $A \subseteq B$ then $(A \cup K) \subseteq (B \cup K)$ (by basic set theory). Since Cn is classical it satisfies Monotony, so if $(A \cup K) \subseteq (B \cup K)$, then $Cn(A \cup K) \subseteq Cn(B \cup K)$. By transitivity of $\subseteq$, then: if $A \subseteq B$, then $Cn(A \cup K) \subseteq Cn(B \cup K)$. The latter is just $Cn_K(A) \subseteq Cn_K(B)$, so we are done.

For Closure:
We need to show $Cn_K(Cn_K(A)) \subseteq Cn_K(A)$. This, by definition of Cn_K, is $Cn(Cn_K(A) \cup K) \subseteq Cn(A \cup K)$, which again by definition of Cn_K, is (a) $Cn(Cn(A \cup K) \cup K) \subseteq Cn(A \cup K)$. Now, $(A \cup K) \subseteq Cn(A \cup K)$ by Inclusion of Cn; and since $K \subseteq (A \cup K)$ then by transitivity of $\subseteq$ we have $K \subseteq Cn(A \cup K)$. Thus $Cn(A \cup K) \cup K = Cn(A \cup K)$. Substituting in (a), we get that we have to show $Cn(Cn(A \cup K)) \subseteq Cn(A \cup K)$. This is immediate from Cn's satisfying Closure (it's classical).

[Marking scheme*: four points for Monotony; five points for the proof of Closure.]*

Marks: **9**

b Let $models(A)$ be the set of all interpretations in which every member of A is true. Note that $\alpha \in \mathrm{Th}(A)$ iff $models(A) \subseteq models(\{\alpha\})$.

Let a consequence relation $\mathrm{Cn}_{\mathrm{pref}}$ be such that $B \subseteq \mathrm{Cn}_{\mathrm{pref}}(A)$ iff $models_{\mathrm{pref}}(A) \subseteq models(B)$, where $models_{\mathrm{pref}}(A)$ represents a special 'preferred' subset of the models of A. (So $models_{\mathrm{pref}}(A) \subseteq models(A)$.)

Prove that $\mathrm{Cn}_{\mathrm{pref}}$ is supraclassical.

We have to show $Th(A) \subseteq Cn_{pref}(A)$. So assume $\alpha \in Th(A)$. This means that $models(A) \subseteq models(\{\alpha\})$. But then, since $models_{pref}(A) \subset models(A)$, we have $models_{pref}(A) \subseteq models(\{\alpha\})$, i.e., $\alpha \in Cn_{pref}(A)$, as desired.

[Marking scheme*: points off for fallings-away from a correct proof.]*

Marks: $\overline{\mathbf{3}}$

---

c A consequence operator Cn' (not necessarily classical) is said to be Cumulative if it satisfies the following property:

if $A \subseteq B \subseteq \mathrm{Cn}'(A)$ then $\mathrm{Cn}'(A) = \mathrm{Cn}'(B)$

A consequence relation Cn' (again, not necessarily classical) is said to be Reciprocal if it satisfies:

if $A \subseteq \mathrm{Cn}'(B)$ and $B \subseteq \mathrm{Cn}'(A)$ then $\mathrm{Cn}'(A) = \mathrm{Cn}'(B)$

Prove that, given Inclusion, then Cn' is Cumulative if and only if it is Reciprocal.

To show Reciprocal, note that if $A \subseteq Cn'(B)$ then $B \subseteq Cn'(B)$ (from Inclusion) gives us $B \subseteq (A \cup B) \subseteq Cn'(B)$. Similarly, if $B \subseteq Cn'(A)$ then by Inclusion we get $A \subseteq (A \cup B) \subseteq Cn'(A)$. Apply Cumulative to each of these, to get $Cn'(B) = Cn'(A \cup B)$ and $Cn'(A) = Cn'(A \cup B)$, respectively. So $Cn'(A) = Cn'(B)$.

To show Inclusion, suppose $A \subseteq B \subseteq Cn'(A)$. Then $B \subseteq Cn'(A)$, trivially. Similarly, $A \subseteq B \subseteq Cn'(B)$ using inclusion, so $A \subseteq Cn'(B)$. So $Cn'(A) = Cn'(B)$ by Reciprocal.

[Marking scheme*: Four marks for each direction.]*

Marks: $\overline{\mathbf{8}}$

The three parts carry, respectively, 45%, 15%, and 40% of the marks.