Logic Programming 2013–14
Assignment 2: Theory

Model solution

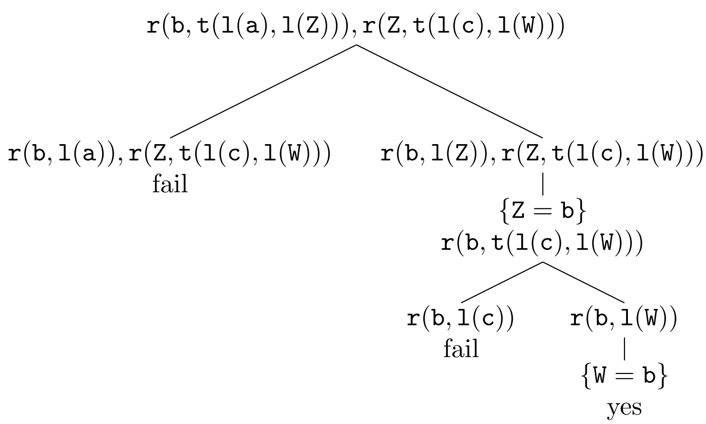
1. (a) Program:

$$\forall \mathtt{X}.\ \mathtt{r(X,l(X))}$$
$$\forall \mathtt{X,Y,Z}.\ \mathtt{r(X,Y)} \ \rightarrow \ \mathtt{r(X,t(Y,Z))}$$
$$\forall \mathtt{X,Y,Z}.\ \mathtt{r(X,Z)} \ \rightarrow \ \mathtt{r(X,t(Y,Z))}$$

Query:

$$\exists Z, W.\ \mathtt{r(b,t(l(a),l(Z)))} \ \wedge \ \mathtt{r(Z,t(l(c),l(W)))}$$

(b) The search tree is:



The response of Prolog is yes with $\mathtt{Z} = \mathtt{b}, \mathtt{W} = \mathtt{b}$

(c) i. Prolog's response to `\+ r(a,l(b))` is yes.
This is because the search for `r(a,l(b))` fails (since it dose not unify with the head of any program clause).

ii. $\neg \mathtt{r(a,l(b))}$ not a logical consequence of the program.
To see this we give a model of the program in which $\mathtt{r(a,l(b))}$ is true. Take the structure whose universe U is the 1-element set $\{0\}$, which determines the interpretation of constants and function symbols. Interpet $r(t_1, t_2)$ as true for all ground terms t_1, t_2. This is easily seen to be a model and, by definition, $\mathtt{r(a,l(b))}$ is true.

iii. $\neg \mathtt{r(a,l(b))}$ is true in the minimum Herbrand model.
A ground atomic formula is true in the minimum Herbrand model if and only if it is provable from the program using a complete proof system (e.g., SLD resolution). $\mathtt{r(a,l(b))}$ is not provable since it does not resolution-combine with any program clause. (Thus $\mathtt{r(a,l(b))}$ is false in the minimum Herbrand model hence $\neg \mathtt{r(a,l(b))}$ is true.)

2. (a) In this part, marks were given for correctness only, since justifications were not requested. Nevertheless, justifications are provided below for their explanatory benefit.

f_1 is not monotonic, it has no fixed points and no least fixed point. (To see this, since every monotonic function has a least fixed point and every function with a least fixed point *a fortiori* has a least fixed point, it is enough to show f_1 has no fixed point. Assume for contradiction that Y is a fixed point. Then $0 \in Y$ iff $0 \notin f(Y)$ iff $0 \notin Y$ (because $Y = f(Y)$), which is indeed a contradiction.)

f_2 is not-monotonic, its fixed points are $\{0\}$, $\{1\}$ and $\{2\}$, but it has no least fixed point.
(Justification: For each $Y \subseteq X$, it holds that $f(Y)$ is a singleton set (i.e., contains exactly one element). So the only possible fixed points are $\{0\}$, $\{1\}$ and $\{2\}$, and these are indeed all fixed points. Since none of the fixed points is a subset of another there is no least fixed point. Hence the function is not monotone.)

f_3 is not-monotonic, its only fixed point is $\{1\}$, this is a least fixed point.
(Justification: by the definition of the function, any fixed point must contain 1 and cannot contain 0 or 2. The only possibility is thus $\{1\}$, which is indeed a fixed point. Since the fixed point is unique it is trivially least. The function is, however, not-monotonic since, e.g., $\{0,1\} \subseteq \{0,1,2\}$ but $f_3(\{0,1\}) = \{1,2\} \not\subseteq \{1\} = f_3(\{0,1,2\})$.)

f_4 is monotonic, its fixed points are $\emptyset$, $\{0,2\}$, $\{1\}$, $\{0,1,2\}$, the least fixed point is $\emptyset$.
(Proof of monotonicity: Suppose $Y \subseteq Y'$. To prove $f_4(Y) \subseteq f_4(Y')$, suppose $z \in f_4(Y)$. Then $z = 2-y$ for some $y \in Y$. Since $Y \subseteq Y'$, we also have $y \in Y'$ hence $2 - y \in f_4(Y')$, i.e., $z \in f_4(Y')$ as required.)

(b) The function f is

$$\begin{aligned} f(Y) \ &= Y \cup \{\mathtt{a}\} \cup \{\mathtt{b} \mid \mathtt{a},\mathtt{c} \in Y\} \cup \{\mathtt{c} \mid \mathtt{b},\mathtt{d} \in Y\} \\ &\qquad \cup \{\mathtt{d} \mid \mathtt{a},\mathtt{e} \in Y\} \cup \{\mathtt{e} \mid \mathtt{a} \in Y\} \cup \{\mathtt{j} \mid \mathtt{b},\mathtt{c} \in Y\} \end{aligned}$$

This reaches a fixed-point via

$$\begin{aligned} f(\emptyset) \ &= \ \{\mathtt{a}\} \\ f^2(\emptyset) \ &= \ \{\mathtt{a},\mathtt{e}\} \\ f^3(\emptyset) \ &= \ \{\mathtt{a},\mathtt{d},\mathtt{e}\} \ = \ f^4(\emptyset) \end{aligned}$$

The least fixed point is $\{\mathtt{a},\mathtt{d},\mathtt{e}\}$.

(c) Given a propositional definite-clause program, the decision procedure constructs the function f and then calculates the iterates $f^n(\emptyset)$ until the least fixed point is reached. It then checks whether a query atom $\mathtt{q}$ belongs to the least fixed point and returns *yes* if it does and *no* otherwise.

(d) Calculating the least-fixed-point is not a possibility for predicate logic, where the analogous procedure would have to construct the set of all ground atomic formulas true in the minimum Herbrand model, which is infinite. Proof search, in contrast, applies to predicate logic as well as to propositional logic.

(Other answers are acceptable here. E.g., proof search gives the programmer control over search and can be adapted to interact with imperative programming features.)