SIF8015 Logic

Exercise 8, Solutions
Logic Programming I

Task 1

(a) $\forall X p(X)$ is true iff for every $X \in U$ we have that $p(X) \in \Im$.
False: $c \in U$ but $p(c) \notin \Im$.

(b) $\forall X q(X)$ is true iff for every $X \in U$ we have that $q(X) \in \Im$.
True: $q(a) \in \Im, q(b) \in \Im, q(c) \in \Im$ and $q(d) \in \Im$.

(c) $\exists X(q(X) \wedge p(X))$ is true iff for some $X \in U$ we have that $q(X) \in \Im$ and $p(X) \in \Im$.
True: $a \in U, q(a) \in \Im$ and $p(a) \in \Im$.

(d) $\forall X(q(X) \rightarrow p(X))$ is true iff for every $X \in U$ we have that $q(X) \notin \Im$ or $p(X) \in \Im$.
False: $c \in U$ but $q(c) \in \Im$ and $p(c) \notin \Im$.

(e) $\forall X(p(X) \rightarrow q(X))$ is true iff for every $X \in U$ we have that $p(X) \notin \Im$ or $q(X) \in \Im$.
True: $q(a) \in \Im, q(b) \in \Im, p(c) \notin \Im$ and $p(d) \notin \Im$.

Task 2

(a) Let P denote the following program:

bigeyes(X) ← frog(X).
frog(bertram).
girl(juliett).
pretty(X) ← girl(X), bigeyes(X).
bigeyes(juliett).

(b) $U_P = \{bertram, juliett\}$

(c) $B_P = \{bigeyes(X), frog(X), girl(X), pretty(X) \mid X \in U_P\}$

(d) $M_P = T_P \uparrow \omega$, so we compute:

$T_P \uparrow 0 \quad \emptyset$
$T_P \uparrow 1 \quad \{frog(bertram), girl(juliett), bigeyes(juliett)\}$
$T_P \uparrow 2 \quad \{frog(bertram), girl(juliett), bigeyes(juliett), bigeyes(bertram), pretty(juliett)\}$
$T_P \uparrow 3 \quad T_P \uparrow 2$

Hence, $M_P = T_P \uparrow \omega = T_P \uparrow 2$.

Task 3

(a) The case numbers refer to the solved form algorithm on page 40 in *Logic, Programming and Prolog (Second Edition)* by Nilsson and Małuszyński.

$\{s_1 \doteq s_2\}$	
$\{f(g(X), g(c)) \doteq f(Y, X), Y \doteq g(g(Z)), c \doteq Z\}$	Case 1
$\{g(X) \doteq Y, g(c) \doteq X, Y \doteq g(g(Z)), c \doteq Z\}$	Case 1
$\{Y \doteq g(X), X \doteq g(c), Y \doteq g(g(Z)), Z \doteq c\}$	Case 4
$\{Y \doteq g(g(c)), X \doteq g(c), Y \doteq g(g(Z)), Z \doteq c\}$	Case 5b
$\{Y \doteq g(g(c)), X \doteq g(c), Y \doteq g(g(c)), Z \doteq c\}$	Case 5b
$\{Y \doteq g(g(c)), X \doteq g(c), Z \doteq c\}$	Elimination of duplicates

(b) $\text{mgu}(s_1, s_2) = \{X/g(c), Y/g(g(c)), Z/c\} = \theta$

$s_1\theta = r(f(g(g(c)), g(c)), g(g(c)), c) = s_2\theta$

Task 4

(a) Case 2 below applies to $a \doteq f(a)$.

$\{q(f(a), X) \doteq q(X, a)\}$	
$\{f(a) \doteq X, X \doteq a\}$	Case 1
$\{X \doteq f(a), X \doteq a\}$	Case 4
$\{a \doteq f(a), X \doteq a\}$	Case 5b
Failure	Case 2

(b) Case 5a below applies to $X \doteq g(f(X))$.

$\{p(f(X), X) \doteq p(Y, g(Y))\}$	
$\{f(X) \doteq Y, X \doteq g(Y)\}$	Case 1
$\{Y \doteq f(X), X \doteq g(Y)\}$	Case 4
$\{Y \doteq f(X), X \doteq g(f(X))\}$	Case 5b
Failure	Case 5a

Task 5

See Figure 1.

Task 6

See Figures 2 and 3. For definite programs, SLD-resolution computes the same answer substitutions independent of the computation rule.

Task 7

We define a helper predicate *insertelement*/3 that inserts an element into a sorted list, and use this to define *insertsort*/2.

$insertelement(E, [], [E]).$
$insertelement(E, [H|T], [E, H|T]) \leftarrow E \leq H.$
$insertelement(E, [H|T], [H|L]) \leftarrow E > H, insertelement(E, T, L).$

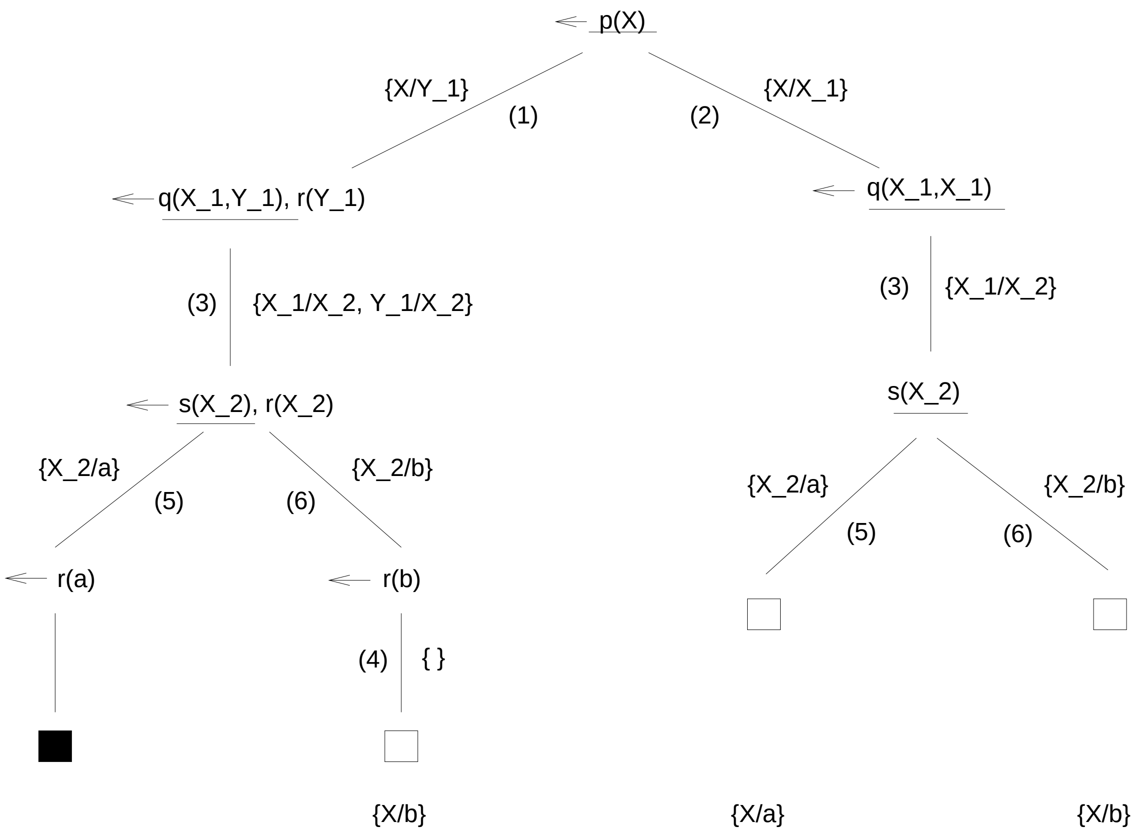

Figure 1: Solution to Task 5.

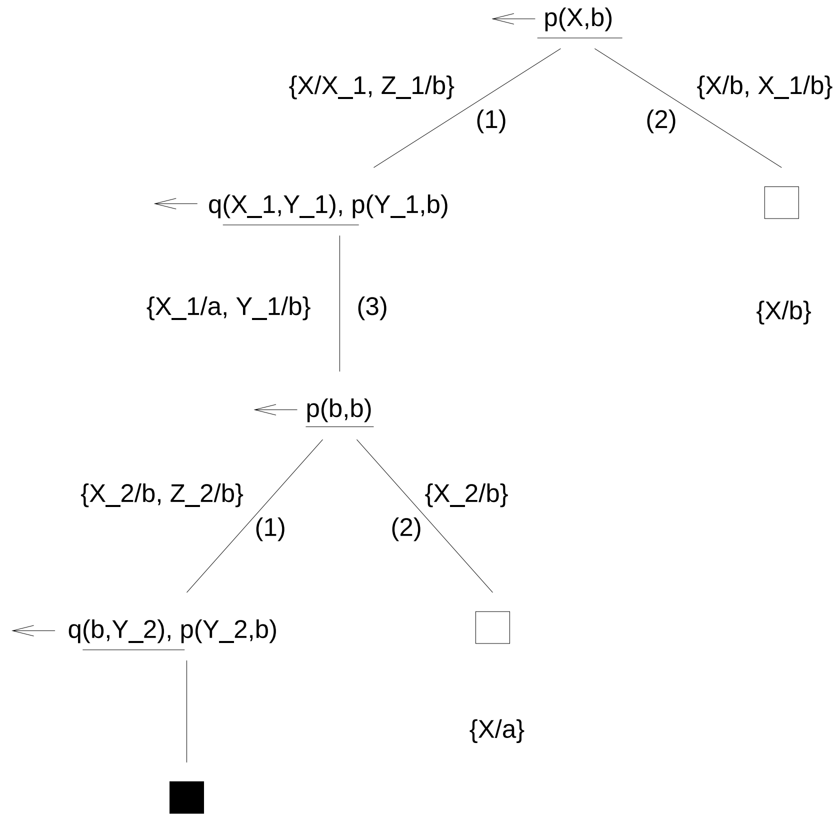

Figure 2: Solution to Task 6 (LM).

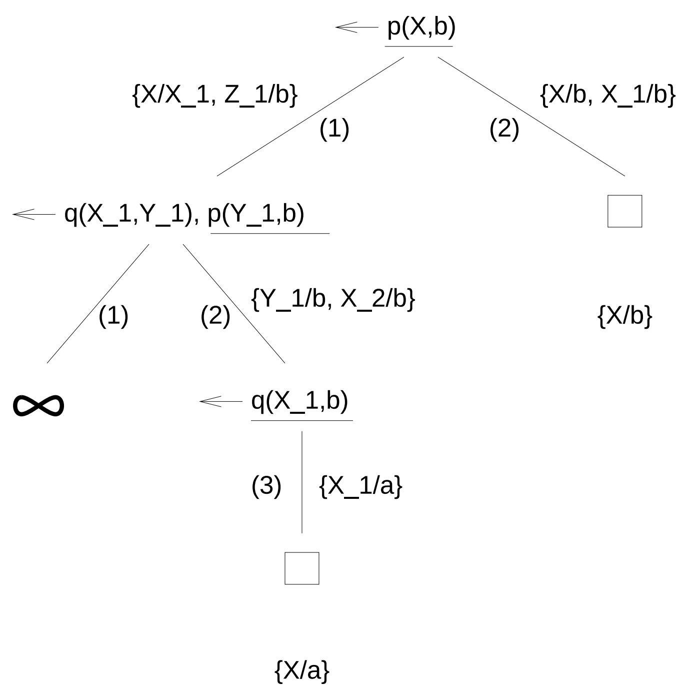

Figure 3: Solution to Task 6 (RM).

$insertsort([\,], [\,])$.
$insertsort([H|T], S_2) \leftarrow insertsort(T, S_1), insertelement(H, S_1, S_2)$.

Task 8

(a) The empty list has length 0. The length of a non-empty list is the length of its tail plus one:

$length([\,], 0)$.
$length([H|T], s(N)) \leftarrow length(T, N)$.

(b) Employ that $0! = 1$ and that $(l+1)! = l! \times (l+1)$:

$fact(0, s(0))$.
$fact(s(L), N) \leftarrow fact(L, M), times(M, s(L), N)$.