EXAM 2 April 29, 2011

This exam consists of five exercises. The available points for each item are written in the margin. You need at least 50 points to pass.

1 Consider the following questions concerning propositional logic.

[7] (a) Give a natural deduction proof of the sequent $\neg\neg p \vee q \vdash \neg p \to (\neg q \to p)$.

[6] (b) Transform the formula

$$(p \to q) \wedge (r \to \neg s) \wedge ((\neg p \wedge \neg r) \to \neg s) \wedge \neg(\neg s \vee \neg(q \to r))$$

into clausal form.

[7] (c) Use resolution to decide whether the formula in part (b) is satisfiable.

2 In the first three parts it is your task to define various ternary Prolog predicates for the exponentiation function. Here, we assume that all numbers involved are natural numbers without zero (i.e., numbers in $\{1, 2, 3, \dots\}$).

[2] (a) Define a predicate `pow/3` where `pow(B,E,N)` should be provable if and only if $\mathtt{B}^{\mathtt{E}} = \mathtt{N}$. You can assume that the first two arguments of `pow` are numbers.

[4] (b) Define a predicate `log/3` where `log(B,E,N)` should be provable if and only if $\mathtt{B}^{\mathtt{E}} = \mathtt{N}$. You can assume that the third argument of `log` is a number. Moreover, you may use the `clpfd`-library.

[6] (c) Define a predicate `exp/3` where `exp(B,E,N)` should be provable if and only if $\mathtt{B}^{\mathtt{E}} = \mathtt{N}$. Now you cannot make any assumptions on the input and you should link the predicates `pow` and `log` properly. If the arguments are not sufficiently instantiated, then your predicate should crash without giving any answer. Try to avoid redundant answers.

Examples:

- `?- exp(3,E,N)` crashes.
- `?- exp(B,1,N)` gives the answer `B = N`.
- `?- exp(B,E,4)` gives the answers `B = 2, E = 2` and `B = 4, E = 1`.

[8] (d) Compute the SLD tree and give all answer substitutions for the Prolog program

```
all([]).                        n(P) :- P, !, fail.
all([P|Ps]) :- P, all(Ps).      n(P).
q(a).
```

and the query `?- all([q(X),n(n(q(Y)))])`.

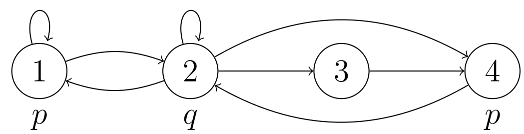
3 For each of the following sequents prove if they are valid or not.

[7] (a) $\exists x\, P(x), \forall x\, a = x \vdash \forall x\, P(x)$

[7] (b) $\exists x\, P(x) \rightarrow \forall x\, a = x \vdash \forall x\, P(x)$

[6] (c) $\forall x\, P(x), \forall x\, a = x \vdash \exists x\, P(x)$

4 Consider the CTL formulas $\phi = \mathsf{E}[(\mathsf{EG}\, p \vee \mathsf{AF}\, p)\, \mathsf{U} (\mathsf{AF}\, p \wedge \mathsf{AX}\, q)]$, $\psi = \mathsf{AG}\, p \vee \mathsf{A}[p\, \mathsf{U}\, p \wedge q]$, and $\chi = \neg \mathsf{E}[\neg q\, \mathsf{U}\, \neg p]$, and the following model $\mathcal{M}$:



[7] (a) Apply the CTL model checking algorithm to determine the states of $\mathcal{M}$ which satisfy ϕ.

[7] (b) Give a model which shows that ψ and χ are not equivalent.

[6] (c) In this part we extend CTL with following new binary temporal operator AR:

$$\begin{aligned} \mathcal{M}, s \models \mathsf{A}[\alpha\, \mathsf{R}\, \beta] \iff & \forall \text{ path } s = s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow \dots \\ & \forall\, i \geqslant 1\ \mathcal{M}, s_i \models \alpha, \text{ or} \\ & \exists\, i \geqslant 1\ \mathcal{M}, s_i \models \beta \text{ and } \forall\, j \leqslant i\ \mathcal{M}, s_j \models \alpha \end{aligned}$$

Extend the model checking algorithm with instructions for deciding which states should be labelled by formulas of the shape $\mathsf{A}[\alpha\, \mathsf{R}\, \beta]$.

[20] **5** Determine whether the following statements are true or false. Every correct answer is worth 2 points. For every wrong answer 1 point is subtracted, provided the total number of points is non-negative.

statement
Executing the Prolog query `?- X = 1+2.` produces the answer `X = 3`.
The term $f(g(x), z)$ is free for x in $\exists y\, P(x) \rightarrow \forall x\, (P(z) \vee \exists y\, Q(x))$.
Every adequate set of temporal CTL connectives contains EG or EF or AU.
The algebraic normal form of the boolean function $f(x, y) = \overline{x} \cdot \overline{y}$ is $1 \oplus x \oplus y \oplus xy$.
The clause $\{\neg p, r\}$ is a resolvent of $\{\neg p, q, r, \neg s\}$ and $\{\neg p, \neg q, s\}$.
$[\![\, \mathsf{EF}\, \phi\,]\!] = [\![\, \phi\,]\!] \cup \mathsf{pre}_{\forall}([\![\, \mathsf{EF}\, \phi\,]\!])$
The backtrack rule can simulate the backjump rule in abstract DPLL.
$\forall x\, (\phi \rightarrow \exists x\, \psi) \dashv\vdash \exists x\, \phi \rightarrow \exists x\, \psi$
$\mathrm{HWB}_3(x, x, x) = x$
The Prolog query `?- select(3,[3,1,3,X],[1|Y]).` admits two different answers.