T-79.5102 **Autumn 2007**
Special Course in Computational Logic
Tutorial 9

1. Consider the following decision problem (formulated as a language):

ENTAILMENT: the set of pairs $\langle S, c\rangle$ where S is a finite set of clauses and $c \in \mathrm{Hb}(S)$ an atom such that $S \models c$.

Show that ENTAILMENT is coNP-complete.

2. Prove the following properties for normal programs.

(a) For a program P and an interpretation $M \subseteq \mathrm{Hb}(P)$,

$$M \models P \iff M \models P^M.$$

(b) For any programs P and Q such that $P \subseteq Q$, $\mathrm{SE}(Q) \subseteq \mathrm{SE}(P)$.

(c) For a program P and a rule $r \in P$,

$$\begin{aligned} & \mathrm{SE}(P \setminus \{r\}) \subseteq \mathrm{SE}(P) \\ \iff \quad & \forall \langle N, M\rangle \in \mathrm{SE}(P \setminus \{r\})\colon M \models r \text{ and } N \models \{r\}^M. \end{aligned}$$

3. A number of *program transformations* have been proposed in the literature. Some of them preserve strong equivalence. Prove that this is the case for the following principles of rule deletion described in terms of strong equivalence. You may consider the class of normal programs for simplicity.

TAUT:	$\{a \leftarrow B, \sim C.\ \} \equiv_{\mathrm{s}} \emptyset$	$(a \in B)$
CONTRA:	$\{a \leftarrow B, \sim C.\ \} \equiv_{\mathrm{s}} \emptyset$	$(B \cap C \neq \emptyset)$
RED$^-$:	$\{a \leftarrow B, \sim C.\ \ c.\ \} \equiv_{\mathrm{s}} \{c.\ \}$	$(c \in C)$
NONMIN:	$\{a \leftarrow B_1, \sim C_1.\ \ a \leftarrow B_2, \sim C_2.\ \}$ $\equiv_{\mathrm{s}} \{a \leftarrow B_1, \sim C_1.\ \}$	$(B_1 \subseteq B_2$ and $C_1 \subseteq C_2)$

In which sense are these results applicable to normal programs?

4. Recall our formalization of coffee orders as an `smodels` program P:

$\{\mathsf{Coffee}, \mathsf{Tea}, \mathsf{Cookie}, \mathsf{Cake}, \mathsf{Cognac}\}.$
$\{\mathsf{Cream}, \mathsf{Sugar}\} \leftarrow \mathsf{Coffee}.$
$\mathsf{Cognac} \leftarrow \mathsf{Coffee}.$
$\{\mathsf{Milk}, \mathsf{Lemon}, \mathsf{Sugar}\} \leftarrow \mathsf{Tea}.$
$\mathsf{Mess} \leftarrow \mathsf{Milk}, \mathsf{Lemon}.$
$\mathsf{Happy} \leftarrow 1\,\{\mathsf{Cookie}, \mathsf{Cake}, \mathsf{Cognac}\}.$
$\mathsf{Broke} \leftarrow 6\,[\mathsf{Coffee} = 1, \mathsf{Tea} = 1, \mathsf{Cookie} = 1, \mathsf{Cake} = 2, \mathsf{Cognac} = 4].$
$\mathsf{OK} \leftarrow \mathsf{Happy}, \sim\mathsf{Broke}, \sim\mathsf{Mess}.$
$\mathsf{f} \leftarrow \sim\mathsf{OK}, \sim\mathsf{f}.$

Study the effect of dropping an individual rule r from this program by finding potential counter examples to $P \equiv P \setminus \{r\}$, e.g., using the translation $\mathrm{EQT}(P, P \setminus \{r\})$. Is some of the rules redundant in this sense?