T-79.5102 **Autumn 2007**
Special Course in Computational Logic
Tutorial 12

1. Given a supported model M of a normal logic program P, prove that

(a) if P is tight on M, then there is a level numbering $\lambda : M \to \mathbb{N}$, and

(b) the converse does not hold in general.

2. Recall the primitive operations on binary counters $c[1 \dots n]$ and $d[1 \dots n]$ involved in the translation $\mathrm{Tr}_{\mathrm{AT}}(P)$:

(a) The program $\mathrm{NXT}(c[1 \dots n], d[1 \dots n])$ sets the value of $d[1 \dots n]$ as the successor of the value of $c[1 \dots n]$ in binary representation.

(b) The program $\mathrm{FIX}(c[1 \dots n], v)$ sets a fixed value v for $c[1 \dots n]$.

(c) The program $\mathrm{LT}(c[1 \dots n], d[1 \dots n])$ checks whether the value of the counter $c[1 \dots n]$ is lower than that of $d[1 \dots n]$.

(d) The program $\mathrm{EQ}(c[1 \dots n], d[1 \dots n])$ tests whether the values of the counters $c[1 \dots n]$ and $d[1 \dots n]$ are the same.

Encode these programs using only *atomic rules* of the form $a \leftarrow {\sim}C$.

3. The correctness proof of $\mathrm{Tr}_{\mathrm{AT}}(P)$ treats the translation in the respective parts $\mathrm{Tr}_{\mathrm{SUPP}}(P)$, $\mathrm{Tr}_{\mathrm{CTR}}(P)$, $\mathrm{Tr}_{\mathrm{MAX}}(P)$, and $\mathrm{Tr}_{\mathrm{MIN}}(P)$, which can be viewed as *program modules*.

(a) Define the module interfaces for these four parts of $\mathrm{Tr}_{\mathrm{AT}}(P)$.

(b) Is the *join* of the respective modules defined?

(c) Explain in which way the *module theorem* can alleviate the proof of $P \equiv_{\mathrm{v}} \mathrm{Tr}_{\mathrm{AT}}(P)$.