Logic Programming 2012–13
Tutorial 3: Logical Consequence

For discussion during Week 5 (Oct. 15–19)

1. Here are three small propositional Prolog programs.

```
p :- q, r.          p :- q, r.          p :- q, r.
q :- s.             q :- s.             q :- p.
q :- t.             q :- t.             q :- t.
r :- t.             r :- p.             r :- t.
t.                  t.                  t.
```

Program 1 **Program 2** **Program 3**

   (a) Rewrite **Program 1** in standard logical notation.

   (b) For each of the three programs, draw the search tree resulting from the initial query `?- p.` In each case, say whether Prolog search (as described in Theory Lecture 2) will answer `yes`, `no` or go into a loop.

   (c) The Prolog inference procedure is *incomplete*. Say what this means. Do any of the three examples above illustrate this incompleteness?

   (d) What is a *decision procedure* for the problem of deciding if a given atom is a logical consequence of a given theory? Describe how the Prolog depth-first top-down inference procedure could be adapted to provide a decision procedure for inference from propositional definite clauses.

2. Consider the following Prolog program operating on unary natural numbers (cf. Programming Lecture 3).

```
lessthan(z,s(_)).
lessthan(s(X),s(Y)) :- lessthan(X,Y).
```

   (a) Write this program in logical notation, with each line written as a sentence expressed as a universally quantified definite clause. (That is, the universal quantifiers should be explicitly written.)

   (b) Consider three structures $\mathcal{S}_1$, $\mathcal{S}_2$ and $\mathcal{S}_3$ defined as follows. The corresponding universes are U_1, U_2 and U_3 as specified below.

$$\begin{aligned} U_1 &= \mathbb{N} && \text{(i.e., } \{0, 1, 2, 3, \dots\}) \\ U_2 &= \mathbb{Z} && \text{(i.e., } \{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}) \\ U_3 &= \mathbb{R}^+ && \text{(i.e., } \{x \in \mathbb{R} \mid x \geq 0\}) \end{aligned}$$

The interpretation of the constant `z`, function symbol `s`. and predicate symbol `lessthan` are defined in all three structures by:

$$\begin{aligned} \texttt{z}^{\mathcal{S}} &= 0 \\ \texttt{s}^{\mathcal{S}}(x) &= x+1 \\ \texttt{lessthan}^{\mathcal{S}}(x,y) &\Leftrightarrow x < y \end{aligned}$$

Which of these structures are models of the program? Justify your answers.

(c) For each of the following predicate logic sentences, say which of the three structures it holds in, and whether or not it is a logical consequence of the example program. In each case, justify your answers.

i. `lessthan(z,z)`

ii. $\forall$`X.` `lessthan(z,s(s(X)))`

iii. $\forall$`X,Y.` `lessthan(X,Y)` $\rightarrow \exists Z.$ `lessthan(X,Z)` $\wedge$ `lessthan(Z,Y)`

iv. $\neg$ `lessthan(z,z)`

(The last of these is quite tricky.)

3. Suppose that we extend the language of propositional programs to allow negated atoms to appear in clauses (this is standard logical negation with the usual truth table, **not** Prolog negation, which is something else). Consider the two programs below.

¬`q :-` ¬`p.`	`p :-` ¬`p.`
`q.`	
Program 1	**Program 2**

(a) Write both these programs in logical notation.

(b) Consider the query `?- p.` For each program, say whether `p` is a logical consequence of the program, and justify your answer.

(c) What difficulties arise in extending Prolog proof search to allow logically negated atoms in program clauses?

We will look at Prolog's treatment of negation later in the course.