

# Logic Programming, Fall 2014

## Tutorial 8: sample answers

1.

- (a) i. Add a to program. So program is:
  - a.
  - and solve b => a
- ii. For this, add b to program. So program is:
  - a.
  - b.
  - and solve a. This succeeds.
- (b) i. Add a => (b => c) to program. So program is:
  - c :- a,b.
  - (see slide in Lec 9 about right associativity of bracketing)
  - and solve (a => b) => (a => c)
- ii. For this, add (a => b) to the program, so program is
  - c :- a,b.
  - b :- a.
  - and solve a => c
- iii. For this, add a to the program, so program is
  - c :- a,b.
  - b :- a.
  - a.
  - and solve c. This succeeds.

2.

- i. The initial goal is sterile j.
- ii. This matches the head of the first program clause, and we get the new goal
  - pi x\ ((bug x) => (in x j) => (dead x))
- iii. We instantiate x with a fresh constant, a say, so the new goal is:
  - (bug a) => (in a j) => (dead a)
- iv. To solve this, add (bug a) as a new line 4 to the program.
  - The new goal is (in a j) => (dead a)
- v. For this, add (in a j) as a new line 5 to the program.
  - The new goal is (dead a)
- vi. This matches the head of the second program clause, so the new goal is:
  - heated Y, in a Y, bug a
- vii. The first subgoal heated Y matches with axiom 3, so we substitute Y = j leaving the goal:
  - in a j, bug a
- viii. in a j is solved by the axiom we added as line 5, leaving the goal
  - bug a
- ix. bug a is solved by the axiom we added as line 4.
  - Thus the query succeeds.

3. One can write the phrase as:

```
The artist formerly known as "the artist formerly known as prince"
is now known as "the artist"
```

The words in quotes are mentioned. The other ones used. For this example, it is also legitimate too consider the two occurrences of "the" as being used rather than mentioned. That is, there is possible ambiguity in the location of the use/mention borderline, resulting from the ambiguities of natural language. Finally, note that when the mentioned phrase

```
the artist formerly known as prince
```

is itself interpreted, the occurrence of "prince" in this phrase is used rather than mentioned.

4. The declarative reading interprets comma as logical conjunction, which is a commutative operation. That is, the order in which it is written ("A and B" versus "B and A") does not affect the meaning.

5. The call `flatten(t,Y)`, for a term `t`, returns, as the value of `Y`, a list of elements each of which is not itself a list. This list is obtained by accumulating as its elements all terms in `t` that are not themselves lists, but which may nonetheless be buried within nested lists in `t`.

E.g. the following hold

```
flatten([1,[2,[3,[4]]]], [1,2,3,4])
flatten(f(a), [f(a)])
```

The last of these results from the last clause in the program. However, the first clause is responsible for the behaviour

```
?- flatten(X,Y)
Y = [X]
```

which holds even though its substitution instance `flatten([],[])` fails. The variable `X` is mentioned in in the first clause but used in the last.