Department of Computer Science, Australian National University
COMP2600 — Formal Methods in Software Engineering
Semester 2, 2011

Week 12 Tutorial

Miscellaneous topics

Question 1

(1) Consider the following Prolog program:

```
alts([A|Xs],[A,B|Ys]) :- alts(Xs,Ys).     % (A1)
alts([B|Xs],[A,B|Ys]) :- alts(Xs,Ys).     % (A2)
alts([],[]).                               % (A3)
alts([A],[A]).                             % (A4)
```

(a) For the query `alts(Ls,[1,2,3,4,5]).`, what values of `Ls` result, and in what order?

(b) One of its solutions is `Ls = [1,4,5]`.

Explain in detail the sequence of steps which leads to the the solution `Ls = [1,4,5]`.

For each of these steps include the following information:

- The subgoal being attempted (such as, eg, "solve `alts(Ls,[1,3,4])`")
- Which rule (of (A1) to (A4)) is matched to that subgoal, in the course of producing the solution
- What solution is immediately and ultimately obtained for the variable (eg, `Ls`) in that subgoal

(c) How would the answer to (a) change if we changed the order of the clauses?

(d) What happens if we try the goal `alts(As, Bs).`? How does this change if we reorder the clauses?

(2) In the course of executing a Prolog program, the Prolog engine needs to unify the following two terms:

```
arr(A, (arr(B, D), G))
arr(arr(int, G), (A, bool))
```

Unify these two terms, so as to obtain the most general unifier. Show clearly the steps in the algorithm you use.

Question 2

We want to infer types for each of the following λ-calculus terms. For each of them

- draw the parse tree, indicating, with type variables, types for each subterm
- generate the constraints relating these types
- solve all the constraints to find the most general type of the whole term

(1) $\lambda f.\ \lambda x.\ f\ x\ x$

(2) $\lambda x.\ \lambda y.\ \lambda z.\ x\ z\ (y\ z)$

(3) $\lambda f.\ \lambda x.\ \lambda y.\ f\ y\ x$

(4) $\lambda g.\ \lambda f.\ \lambda x.\ g\ (f\ x)$

Question 3

In lectures (the lecture on Soundness and Completeness, slides 12 and 13) we have tables giving the relevant rules which correspond to the calculations done in constructing a row of the truth-table, for the $\wedge$ and $\rightarrow$ boolean operators. These tables are also shown below.

p	q	$p \rightarrow q$	corresponding "rule"
T	T	T	$\dfrac{p \quad q}{p \rightarrow q}$
T	F	F	$\dfrac{p \quad \neg q}{\neg(p \rightarrow q)}$
F	T	T	$\dfrac{\neg p \quad q}{p \rightarrow q}$
F	F	T	$\dfrac{\neg p \quad \neg q}{p \rightarrow q}$

p	q	$p \wedge q$	corresponding "rule"
T	T	T	$\dfrac{p \quad q}{p \wedge q}$
T	F	F	$\dfrac{p \quad \neg q}{\neg(p \wedge q)}$
F	T	F	$\dfrac{\neg p \quad q}{\neg(p \wedge q)}$
F	F	F	$\dfrac{\neg p \quad \neg q}{\neg(p \wedge q)}$

Construct similar tables for the $\vee$ and $\neg$ operators. Make sure you know how to prove the corresponding rules.