Assignment 9
Semantics, WS 2011-2012

Prof. Dr. Gert Smolka, Dr. Chad Brown
www.ps.uni-saarland.de/courses/cl-ss11/

Read in the lecture notes: Chapter 5

Exercise 9.1

a) Prove that r is confluent if and only if *star r* satisfies the diamond property.
b) Prove that relations satisfying the diamond property are strongly confluent.
c) Prove that *star* preserves the diamond property.

Exercise 9.2 Prove the following goals stating two variants of the principle of well-founded induction.

```
Goal forall (r : rel) (p : X -> Prop) (x : X),
terminates r x ->
(forall x, (forall y, r x y -> p y) -> p x) ->
p x.
```

```
Goal forall (r : rel) (p : X -> Prop) (x : X),
terminates r x ->
(forall x, terminates r x -> (forall y, r x y -> p y) -> p x) ->
p x.
```

Exercise 9.3 Size induction generalizes complete induction to arbitrary types by employing a size function. Prove the following lemma providing for proofs by size induction.

```
Lemma size_induction {X : Type} (f : X -> nat) (p: X ->Prop) (x : X) :
(forall x, (forall y, f y < f x -> p y) -> p x) -> p x.
```

Hint: Follow the proof script for complete induction. Before doing the induction insert *remember* $(f\,x)$ *as* n so that you can do induction on n.

Exercise 9.4 Prove the following lemma, which says that a relation terminates if each step decreases the size of a node.

```
Lemma size_termination {X : Type} (r : rel X) (f : X -> nat) :
(forall x y, r x y -> f x > f y) -> terminating r.
```

Exercise 9.5 The **lexical product** of two relations is defined as follows.

```
Definition lex {X Y : Type} (r : rel X) (s : rel Y) : rel (X * Y) :=
fun p q => let (x,y) := p in let (x',y') := q in
r x x' \/ x=x' /\ s y y'.
```

a) Prove that the lexical product of two terminating relations is terminating.

```
Lemma lex_terminates {X Y : Type} (r : rel X) (s : rel Y) x y :
terminates r x -> terminating s -> terminates (lex r s) (x,y).
```

b) Find an example that shows that the lemma is unprovable if the termination of s is only required for y.

Exercise 9.6 Consider the following type of infinitely branching trees and *subtree* relation.

```
Inductive tree : Type :=
| treeL : tree
| treeN : (nat -> tree) -> tree.

Definition subtree : rel tree :=
fun s t => match s with
| treeL => False
| treeN f => exists n, f n = t
end.
```

a) Prove *subtree* terminates.

b) Prove *treeL* is normal.

c) Prove *treeL* is the normal form of any tree.

d) Prove *subtree* is confluent.

e) Prove *subtree* does not have the diamond property.

Exercise 9.7 We consider arithmetic expressions

$$e ::= O \mid Se \mid e + e$$

a) Define an abstract syntax as an inductive type *exp*.

b) Define a semantics $eval : exp \to nat$.

c) Define an inductive predicate $step : rel\ exp$ representing the rewrite rules

$$\begin{aligned} 0 + e &\to e \\ Se_1 + e_2 &\to S(e_1 + e_2) \end{aligned}$$

d) Prove that *step* is sound.

e) Define a size function for *exp*.

f) Prove that *step* is terminating.

g) Give an inductive definition *red* : *exp*− > *Prop* characterizing reducible expressions.

h) Prove *red* agrees with *reducible step*.

i) Give an inductive definition *norm* : *exp*− > *Prop* characterizing normal expressions.

j) Prove exhaustiveness of *red* and *norm*. (That is, every expression satisfies *red* or *norm*.)

k) Prove disjointness of *red* and *norm*. (That is, no expression satisfies both *red* and *norm*.)

l) Prove *norm* agrees with *normal step*.

m) Prove that *reducible step* is decidable.

n) Prove that *step* is complete.

o) Prove that *step* is normalizing.

p) Prove that *step* is confluent.

q) Prove that two expressions are convertible (by step) if and only if they evaluation to the same natural number.

r) **Challenge:** Define a function *nf:exp −> exp* that computes the normal form of an expression and prove it correct.

Exercise 9.8 Give the invariants for the following verification problems.

a) $\{P\}$ while true do skip $\{Q\}$

b) $\{X \le 3\}$ while $X \le 2$ do inc X $\{X = 3\}$

c) $\{X = x \wedge Z = z\}$ while $X \ne 0$ do dec Z; dec X $\{Z = z - x\}$

d) $\{X = x\}$ $Y := 0$; while $X \ne 0$ do inc Y; dec X $\{Y = x\}$

e) $\{X = x\}$ $Y := 0$; while $X \ne 0$ do $Y := 1 - Y$; dec X $\{Y = 0 \leftrightarrow \text{even}\, x\}$

Exercise 9.9 Prove the following in Coq.

a) $\forall PQ.$Hoare P (while true do skip) Q

b) $\forall PQ.$hoare P (while true do skip) Q