Assignment 7
Semantics, WS 2011-2012

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

Read in the lecture notes:

Read the new material in Chapter 4 of the lecture notes.

Exercise 7.1 Give the induction principles for the following inductive predicates defined in this chapter and check your results with Coq.

a) *And*

b) *Eq*

c) *EQ*

Exercise 7.2 Give the induction principle for the following inductive predicate.

```
Inductive le2 : nat -> nat -> Prop :=
| le2x : forall x, le2 x x
| le2S : forall x y, le2 x y -> le2 x (S y).
```

Exercise 7.3 Prove the following goal. Do not use a lemma.

```
Lemma starTR : forall x y z, star x y -> R y z -> star x z.
```

Exercise 7.4 Give the induction principle for the inductive predicate *star*.

Exercise 7.5 We can define a reflexive transitive closure predicate *star1* with a single proper argument.

```
Inductive star1 (x : X) : X -> Prop :=
| star1R : star1 x x
| star1T : forall y z, star1 x y -> R y z -> star1 x z.
```

a) Give the induction principle for *star1*.

b) Prove that *star1* is reflexive and transitive.

c) Prove $\forall x y.\ star\, x\, y \leftrightarrow star1\, x\, y$

Exercise 7.6 Prove that taking the reflexive transitive closure preserves invariants.

```
Definition invariant {X : Type} (p : X -> Prop) (R : X -> X -> Prop) : Prop :=
forall x y, R x y -> p x -> p y.
```

```
Goal forall (X : Type) (R : X -> X -> Prop) (p : X -> Prop),
invariant p R -> invariant p (star R).
```

Exercise 7.7 You may have seen $R^* := \bigcup_{n\in\mathbb{N}} R^n$ as a definition of the reflexive transitive closure. Using the function *iter*, we can express this definition in Coq.

```
Definition comp {X : Type} (R S : X -> X -> Prop) (x z : X) : Prop :=
exists y, R x y /\ S y z.
```

```
Definition stari {X : Type} (R : X -> X -> Prop) (x y : X) :=
exists n, iter n (comp R) (fun x y => x=y) x y.
```

Prove the equivalence of the inductive and the iterative definition.

```
Goal forall (X : Type) (R : X -> X-> Prop) (x y : X),
star R x y <-> stari R x y.
```

Exercise 7.8 Give three proofs for the following goal.

```
Goal forall r r' : rel, rap r r' -> functional r' -> functional r.
```

a) Use *firstorder*.

b) Use *eauto*.

c) Use *eapply* and *eassumption*.

Exercise 7.9 Download the Coq code from the lecture of November 30. Consider the correctness theorem for the compiler from commands to regular expressions.

```
Theorem correctness c :
req (sem c) (den (compile c)).
```

The proof has two directions. The proof of the first direction is given.

a) Rewrite the proof of the first direction so that it does not use *firstorder* or *eauto*.

b) Write the proof of the second direction.

Exercise 7.10 Prove the following goal by applying the induction principle *even_ind*. Do not use the induction tactic.

```
Goal forall n, even n -> even (S n) -> False.
```