Universität Innsbruck
Institut für Informatik
Prof. Dr. Clemens Ballarin

SS 2008
Exercise Sheet 4
Due 29 May 2008

Automatic Deduction — Introduction to Isabelle

LVA 703522

1 Permutations of Lists

In this exercise we consider lists (over an arbitrary element type). The cons operation is denoted by $x \cdot xs$, $|xs|$ is the length of xs and xs_i the ith element. Permutations of lists are defined inductively by the following four rules.

$$([], []) \in \mathbf{Perm} \quad \text{(Nil)} \qquad (x \cdot y \cdot l, y \cdot x \cdot l) \in \mathbf{Perm} \quad \text{(Swap)}$$

$$\frac{(xs, ys) \in \mathbf{Perm}}{(z \cdot xs, z \cdot ys) \in \mathbf{Perm}} \quad \text{(Cons)} \qquad \frac{(xs, ys) \in \mathbf{Perm} \qquad (ys, zs) \in \mathbf{Perm}}{(xs, zs) \in \mathbf{Perm}} \quad \text{(Trans)}$$

The defined set **Perm** contains pairs of lists. In each pair the lists only in the order of elements.

▷ State the induction rule and prove the following statements (on paper).

a) For $(xs, ys) \in \mathbf{Perm}$ holds: xs and ys have equal length.

b) For $(xs, ys) \in \mathbf{Perm}$ holds: there is a permutation π of numbers $1 \ldots |xs|$, such that $xs_i = ys_{\pi(i)}$ for all $i = 1 \ldots |xs|$.

2 Rule Induction

Formalise part of the lecture on inductive sets in Isabelle.

We define a predicate `closed f A`, where `f::'a set ⇒ 'a set` and `A::'a set`.

```
definition closed :: "('a set ⇒ 'a set) ⇒ 'a set ⇒ bool"
  where "closed f A ≡ f A ⊆ A"
```

▷ Show `closed f A ∧ closed f B ⟹ closed f (A ∩ B)` if `f` is monotone (the predicate `mono` is predefined).

▷ Define a function `lfpt` mapping `f` to the intersection of all `f`-closed sets.

▷ Show that `lfpt f` is a fixed point of `f` if `f` is monotone.

▷ Show that `lfpt f` is the least fixpoint of `f`.

We now declare a constant `R::('a set × 'a) set`. This is the set of rules, which will not be further specified here.

```
consts R :: "('a set × 'a) set"
```

Then we define `Rhat::'a set ⇒ 'a set` in terms of `R`.

```
definition Rhat :: "'a set ⇒ 'a set"
  where "Rhat B ≡ {x. ∃H. (H,x) ∈ R ∧ H ⊆ B}"
```

▷ Show soundness of rule induction using `R` and `lfpt Rhat`.

3 Two Grammars

The most natural definition of valid sequences of parentheses is this:

$$S \quad \rightarrow \quad \epsilon \quad | \quad '(' \; S \; ')' \quad | \quad S \; S$$

where ϵ is the empty word.

A second, somewhat unusual grammar is the following one:

$$T \quad \rightarrow \quad \epsilon \quad | \quad T \; '(' \; T \; ')'$$

▷ Model both grammars as inductive sets S and T and prove, on paper and using rule inducion, $S = T$.