Universität Innsbruck **SS 2008**
Institut für Informatik **Exercise Sheet 4**
Prof. Dr. Clemens Ballarin **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.

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

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

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

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

The induction rule consists of two bases cases and two induction steps. For $(xs, ys) \in \textbf{Perm} \Rightarrow P(xs, ys)$ for some property P one needs to show:

- Base cases:

  Nil: $P([], [])$

  Swap: $\forall x, y, l.\ P(x \cdot y \cdot l, y \cdot x \cdot l)$

- Induction steps

  Cons: $\forall xs, ys, z.\ P(xs, ys) \Rightarrow P(z \cdot xs, z \cdot ys)$

  Trans: $\forall xs, ys, zs.\ P(xs, ys) \wedge P(ys, zs) \Rightarrow P(xs, zs)$

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

   To be shown: $(xs, ys) \in \textbf{Perm} \Rightarrow |xs| = |ys|$.
   Applying the induction rule yields four statements to be shown:

   Nil: $|[]| = |[]|$ $\surd$

   Swap: $|x \cdot y \cdot l| = |y \cdot x \cdot l|$ $\surd$

Cons: Induction hypothesis: $|xs| = |ys|$

From this the following is immediate: $|z{\cdot}xs| = |xs|{+}1 \stackrel{\text{IH}}{=} |ys|{+}1 = |z \cdot ys|$.

Trans: Induction hypothesis: $|xs| = |ys|, |ys| = |zs|$

By transitivity of equality: $|xs| = |zs|$

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

By rule induction we again obtain four statements, which are to be shown:

Nil: There is a permutation π with $[]_i = []_{\pi(i)}$. This is trivial since the list is empty.

Swap: There is a permuation π with $(x \cdot y \cdot l)_i = (y \cdot x \cdot l)_{\pi(i)}$. This holds for $\pi = (12)$.

Cons: The induction hypothesis says that there exists a permutation π with $xs_i = ys_{\pi(i)}$ for all i (from 1 to $|xs|$).

From this we obtain a permuation τ by setting $\tau(1) := 1$ and $\tau(i) := \pi(i-1) + 1$ für $i > 1$. We have $(z \cdot xs)_i = (z \cdot ys)_{\tau(i)}$.

Trans: Induction hypothesis: there is a permutation π with $xs_i = ys_{\pi(i)}$ for all i and a permutation τ with $ys_i = zs_{\tau(i)}$ for all i.

Then $\tau \circ \pi$ is also a permutation, and $xs_i = zs_{\tau(\pi(i))}$.

2 Rule Induction

Formalise part of the lecture on inductive sets in Isabelle.

▷ 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).

```
lemma closed_int:
  "⟦ mono f; closed f A; closed f B ⟧ ⟹ closed f (A ∩ B)"
  by (unfold closed_def mono_def) blast
```

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

```
definition lfpt :: "('a set ⇒ 'a set) ⇒ 'a set"
  where "lfpt f ≡ ⋂ {B. closed f B}"
```

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

```
lemma lfpt_lower: "closed f B ⟹ lfpt f ⊆ B"
  by (unfold lfpt_def) auto

lemma lfpt_greatest:
  assumes A_smaller: "⋀B. closed f B ⟹ A ⊆ B"
  shows "A ⊆ lfpt f"
  by (unfold lfpt_def) (blast dest: A_smaller)

lemma 1:
  "mono f ⟹ f (lfpt f) ⊆ lfpt f"
  apply (rule lfpt_greatest)
  apply (rule subset_trans)
   apply (erule monoD)
   apply (erule lfpt_lower)
  apply (unfold closed_def)
  apply assumption
  done

lemma 2:
  "mono f ⟹ lfpt f ⊆ f (lfpt f)"
  apply (rule lfpt_lower)
  apply (unfold closed_def)
  apply (rule monoD, assumption)
  apply (rule 1, assumption)
  done

lemma lfpt_fixpoint:
  "mono f ⟹ f (lfpt f) = lfpt f"
  by (blast intro!: 1 2)
```

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

```
lemma lfpt_least:
  assumes A: "A = f A"
  shows "lfpt f ⊆ A"
proof -
  from A have "closed f A" by (unfold closed_def) blast
  then show "lfpt f ⊆ A" by (rule lfpt_lower)
qed
```

▷ 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"
```

▷ 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`.

```
lemma monoRhat: "mono Rhat"
```

```
  by (unfold mono_def Rhat_def) blast
```

Soundness of *rule induction* means that if some predicate `P` can be verified by rule induction, then `P` holds for all elements of the set (constructed as least fixed point).

```
lemma soundness:
  assumes hyp: "∀(H,x) ∈ R. ((∀h ∈ H. P h) ⟶ P x)"
  shows "∀x ∈ lfpt Rhat. P x"
proof -
  from hyp have "closed Rhat {x. P x}"
    by (unfold closed_def Rhat_def) blast
  then have "lfpt Rhat ⊆ {x. P x}" by (rule lfpt_lower)
  then show ?thesis by blast
qed
```

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, $S = T$.

The inductive definitions are

$$\varepsilon \in S \quad (\text{S1}) \qquad \frac{w \in S}{(w) \in S} \quad (\text{S2}) \qquad \frac{v \in S \qquad w \in S}{vw \in S} \quad (\text{S3})$$

and

$$\varepsilon \in T \quad (\text{T1}) \qquad \frac{v \in T \qquad w \in T}{v(w) \in T} \quad (\text{T23})$$

In order to show $S = T$ we show that S is contained in T and T in S. The latter is simpler, hence it is shown first.
In order to show $T \subseteq S$ we show that for any x, $x \in T \Longrightarrow x \in S$ by rule induction for the set T.

T1: $\varepsilon \in S \quad \surd$

T23: Induction hypothesis: $v \in S, w \in S$.

We need to show that $v(w) \in S$, which follows from the induction hypothesis by the following inference:

$$\dfrac{v \in S \qquad \dfrac{w \in S}{(w) \in S}\,(\text{S2})}{v(w) \in S}\,(\text{S3})$$

For the direction $S \subseteq T$ we use the lemma (shown below).

$$\dfrac{v \in T \qquad w \in T}{vw \in T}\,(\text{T3})$$

Similar to the before, we show that for any x, $x \in S \Longrightarrow x \in T$, this time by rule induction for the set S.

S1: $\varepsilon \in T \quad \surd$

S2: Induction hypothesis: $w \in T$.

Show that $(w) \in T$. This follows from the induction hypothesis by (T23) where $v = \varepsilon$.

S3: Induction hypothesis: $v \in T, w \in T$.

Show that $vw \in T$. Immediate with (T3).

Proof of Lemma (T3).

Following the scheme of the lecture, the induction rule for T is the theorem

$$\dfrac{x \in T \qquad P\,\varepsilon \qquad \dfrac{P\,v \qquad P\,w}{P\,v(w)}}{P\,x}$$

By setting $P\,x \;\equiv\; x \in T \wedge Q\,x$ for an arbitrary predicate Q we obtain this stronger version of the induction rule:

$$\dfrac{x \in T \qquad Q\,\varepsilon \qquad \dfrac{v \in T \qquad Q\,v \qquad w \in T \qquad Q\,w}{Q\,v(w)}}{Q\,x}$$

This is the rule that Isabelle derives. This rule is used in the proof of (T3). We will use the additional induction hypotheses in the proof of (T3). We show $vw \in T$ by rule induction on the second premise $w \in T$:

T1 $w = \varepsilon$

Show $v\varepsilon \in T$. This follows from the first premise $v \in T$.

T23 $w = v'(w')$

Show $vv'(w') \in T$. By induction hypothesis $vv' \in T, vw' \in T$. By induction hypothesis of the stronger induction rule also $v' \in T, w' \in T$. The goal is shown by (T23) from $vv' \in T$ and $w' \in T$.