Universität Innsbruck
Institut für Informatik
Prof. Dr. Clemens Ballarin

SS 2008
Exercise Sheet 1
Due 3 April 2008

Automatic Deduction — Introduction to Isabelle
LVA 703522

1 Lambda Calculus

1.1 α-Conversion

$\triangleright$ Which of the following λ-terms (if any) are α-equivalent?

(a) $\lambda x\, y.\, y\, x$	(e) $\lambda x.\, y\, x$
(b) $\lambda x\, y.\, x\, y$	(f) $\lambda x.\, x\, y$
(c) $\lambda y\, x.\, y\, x$	(g) $\lambda y.\, y\, x$
(d) $\lambda y\, x.\, x\, y$	(h) $\lambda y.\, x\, y$

1.2 β-Reduction

$\triangleright$ β-reduce the following λ-terms to their normal form. How many derivations of the normal form exist?

(a) $(\lambda x\, y.\, x)\, (\lambda x.\, x\, x)\, (\lambda x.\, x\, x)$

(b) $(\lambda x\, y.\, y)\, (\lambda x.\, x\, x)\, (\lambda x.\, x\, x)$

(c) $(\lambda x.\, f\, x\, x)\, ((\lambda f.\, f\, x)\, g)$

1.3 Church Numerals

Recall the encoding of natural numbers as λ-terms via *Church numerals*:

$$n \equiv \lambda f\, x.\, f^n x$$

$\triangleright$ Define a function $\mathrm{mult} \equiv \lambda m\, n.\, \ldots$ as a λ-term such that $\mathrm{mult}\, m\, n$ encodes $m \cdot n$, i.e. multiplication of Church numerals.

$\triangleright$ Define a function $\exp \equiv \lambda m\, n.\, \ldots$ as a λ-term such that $\exp m\, n$ encodes $m\, \hat{}\, n$, i.e. exponentiation of Church numerals.

1.4 $\alpha\beta\eta$-Equivalence in Isabelle

Equality in Isabelle is (usually) modulo $\alpha\beta\eta$-equivalence. Therefore the theorem `refl`: `t = t` is sufficient to prove the following lemmas:

```
lemma alpha: "(λx. x) = (λy. y)"
  by (rule refl)

lemma beta: "(λf. f x) f = f x"
  by (rule refl)

lemma eta: "(λx. f x) = f"
  by (rule refl)
```

▷ Prove these lemmas in Isabelle. What happens wrt. $\alpha\beta\eta$-equivalence?

1.5 Church Numerals in Isabelle

In this exercise we will use Isabelle to perform computations with Church numerals. Isabelle's *simplifier*, which is invoked on a proof state by `simp`, performs term rewriting. `unfolding` can be used to unfold definitions.

▷ Make yourself familiar with the `definition` command (Section 4.1.1 of the Isabelle/Isar Reference manual). Use it to define the functions `add`, `mult` and `exp` (cf. Exercise 1.3) which perform addition, multiplication and exponentiation, respectively, of Church numerals in Isabelle.

▷ Prove the following lemmas. Which values are computed for `?x`?

```
lemma "add (λf x. f (f x)) (λf x. f (f (f x))) = ?x"
  unfolding add_def  — we can unfold the definition first ...
  apply simp
  done

lemma "mult (λf x. f (f x)) (λf x. f (f (f x))) = ?x"
  by (simp add: mult_def)  — ... or give it to the simplifier directly
```

▷ Compute $2 + 2 \cdot (3 + 1)$ in a similar fashion.

▷ Show that $2 \,\hat{}\, 3 = 2 \cdot 2 \cdot 2$.