Assignment 2

Out: Tuesday, 14 September 2010
Due: Tuesday, 28 September 2010

Total: 100 points + 40 bonus points

Exercise 1 (30 points) Let us use the following datatypes \texttt{term} and \texttt{term1} for representing untyped (pure) \(\lambda\)-terms and their \(\alpha\)-normal forms, respectively:

\[
\begin{align*}
\text{datatype} \ \text{term} &= \ \\
&= \ \text{TmVar of string} \ | \ \text{TmLam of (string, term)} \ | \ \text{TmApp of (term, term)} \ \\
\text{datatype} \ \text{term1} &= \ \\
&= \ \text{TmVar1 of string} \ | \ \text{TmInd1 of int} \ | \ \text{TmLam1 of term1} \ | \ \text{TmApp1 of (term1, term1)}
\end{align*}
\]

• (10 points) Please implement a function \texttt{nf_alpha} that translates a given \(\lambda\)-term into its alpha-normal form:

\[
\text{extern fun nf_alpha (t: term): term1}
\]

• (20 points) Please implement a function \texttt{subst} of the following type:

\[
\text{extern fun subst (t: term, x: string, s: term): term}
\]

Given a term \(t\), a variable \(x\) and a term \(s\), \texttt{subst}(\(t, x, s\)) should return \(t[x \mapsto s]\).

Exercise 2 (20 points) Please prove in ATS that \((2n+1)^2 - 1\) is a multiple of \(8\) for every natural number \(n\).

Exercise 3 (30 points) The following declared dataprop \texttt{F91} encodes the MacCarthy’s 91-function:

\[
\text{dataprop} \ \text{F91 (int, int) =}
\begin{align*}
| \ F91\text{def1 (91, 91)} \\
| \ {i: int | i <= 100; i <> 91} \ {r1, r2: int} \\
& \ F91\text{def2 (i, r2) of (F91 (i+11, r1), F91 (r1, r2))} \\
| \ {i: int | 101 <= i} \ {r: int} \\
& \ F91\text{def3 (i, r) of F91 (i-10, r)}
\end{align*}
\]

Given integers \(i\) and \(r\), if a proof value can be assigned the type \texttt{F91}(\(i, r\)), then \(f91(i) = r\), where \(f91\) is formally defined as follows:

\[
f91(i) = \begin{cases} 
                f91(i - 10) & \text{if } i \geq 101; \\
                f91(f91(i + 11)) & \text{if } i \leq 100 \wedge i \neq 91; \\
                91 & \text{otherwise.}
          \end{cases}
\]

Please construct a proof function \texttt{F91istot} in ATS that is declared as follows:
Exercise 4 (40 bonus points) The definition of Braun trees is encoded into the following declared dataprop isBraun:

```
datasort bt = B of (bt, bt) | E of ()

dataprop isBraun (bt) =
  | {t1,t2:bt} {s1,s2:nat | s2 <= s1; s1 <= 1 + s2} isBraun_B (B (t1, t2)) of (isBraun t1, isBraun t2, btsz (t1, s1), btsz (t2, s2))
  | isBraun_E (E ()) of ()
```

Please construct a proof function brauntree_height_lemma in ATS that is declared as follows:

```
prfun brauntree_height_lemma {t1,t2:bt} {h,h1:nat}
  (pf0: isBraun (B (t1, t2)), pf1: btht (B (t1, t2), h), pf2: btht (t1, h1))
  : [h == h1+1] void
```

Please also construct a proof function brauntree_size_height_lemma in ATS that is declared as follows:

```
prfun brauntree_size_height_lemma {t:bt} {s,h,n:nat}
  (pf0: isBraun (t), pf1: btsz (t, s), pf2: btht (t, h), pf3: POW2 (h, n))
  : [n <= s + s + 1] void
```

Note that the dataprops btsz, btht and POW2 are all declared in a file available on-line.