Homework

An exercise that we will do for each of HOL Light, Mizar, and Coq, is to prove the statement:

$$\forall n.(\sum_{i=0}^{n} i)^2 = \sum_{i=0}^{n} i^3 \tag{1}$$

For instance, for $n = 3$ we have $(0+1+2+3)^2 = 6^2 = 36 = 0+1+8+27 = 0^3+1^3+2^3+3^3$. The goal is to make you via this exercise acquainted with the basics of each of HOL Light, Mizar, and Coq, so you can make your own decision later during the course which one you prefer to do the final assignment with.

1. Read the HOL Light tutorial up to and including Section 8. In particular, work through the example in Section 8.2 step by step (this result is used in our proof of (1)).

2. Replay the proof `ADD_0` of the fact that $\forall m.m+0 = m$ in `arith.ml`, step by step. Then try to prove the fact that $\forall m.m = m+0$ in the same way. What is the problem? What solutions can you think of to work around the problem?

3. During the PS we have given a step-by-step paper-proof of (1), splitting it into several lemmas (most of them simple algebraic laws on addition and multiplication like the one of the previous exercise). Write down these lemmas, their dependencies, and formulate them in HOL Light (try to come up with systematic names for them).

4. Prove (1) by successively proving your lemmas of the previous item in HOL Light.
(Although powerful tactics can be used, try using only simple ones in the beginning, to learn how to manipulate statements/theorems.)

5. In the second part of the PS a proof of strong normalisation (termination) of β-reduction for simply type λ-calculus (due to Tait) was given, see the note on the next page.[1]

   - Spell out the details of why the identity substitution is strongly computable.
   - Explain in what way the predicate SC is inductively defined.
   (How would its definition fail to be inductive for the untyped λ-calculus?)

[1] Beware that in the note $\vec{R}$ is used both as a vector of terms (when stand-alone $\vec{R} = R_1, \ldots, R_n$) and as a repeated application (when applied $M\vec{R} = (\ldots(MR_1)\ldots R_n)$, i.e. a sries of applications).

Simply typed lambda calculus is strongly normalizing

The following proof of strong normalization of $\Lambda^{\rightarrow}$, using induction loading on the predicate SN to deal with creation of redexes and substitutions to deal with contraction of existing ones, is a standard strong computibility-style proof. The predicate SC on $\Lambda^{\rightarrow}$ is defined by:

$$M \in SC =^{\text{def}} \forall \vec{R} \in SC.M\vec{R} \in \Lambda^{\rightarrow} \text{ is SN}$$

From the definition, $SC \implies \text{SN}$, SC is closed under β, and SC holds for every variable x, since $\vec{R} \in SC \implies \vec{R} \in \text{SN} \implies x\vec{R} \in \text{SN}$.

THEOREM 1. $\Lambda^{\rightarrow}$ *is* SN.

PROOF We prove *for all terms M and type preserving substitutions σ mapping free variables of M to terms in SC, $M^{\sigma} \in SC$*, by induction on the derivation of $M \in \Lambda^{\rightarrow}$, from which the theorem follows by setting σ to the identity.

(var) $x^{\sigma} = \sigma(x) \in SC$, by assumption,

(app) Let $\vec{R} \in SC$, then $(MN)^{\sigma}\vec{R} = (M^{\sigma}N^{\sigma})\vec{R} = M^{\sigma}N^{\sigma}\vec{R} \in \text{SN}$, by ih for M (and N),

(abs) Let $\vec{R} \in SC$. $(\lambda x.M)^{\sigma}\vec{R} = (\lambda x.M^{\sigma})\vec{R}$. By ih and assumption, M^{σ} and $\vec{R}$ are SN, so an infinite reduction must be of the form $(\lambda x.M^{\sigma})P\vec{Q} \twoheadrightarrow_{\beta} (\lambda x.M')P'\vec{Q} \rightarrow_{\beta} M'^{[x:=P']}\vec{Q'} \rightarrow_{\beta}^{\infty} \ldots$ with $\vec{R} = P\vec{Q}$, but $M'^{[x:=P']}\vec{Q'} \twoheadleftarrow_{\beta} M^{\sigma[x:=P]}\vec{Q} \in \text{SN}$ by ih. $\odot$