CS 476 Homework #5 Due 10:45am on 11/3

Note: Answers to the exercises listed below should be handed to the instructor *in hardcopy* and in *typewritten form* (latex formatting preferred) by the deadline mentioned above. You should also email to Stephen Skeirik (`skeirik2@illinois.edu`) the Maude code for the exercises requiring that. Also, for any difficulties using any of the tools, you can send email to Stephen Skeirik asking for his help.

1. Two equational theories (Σ, E) and (Σ, E'), with same signature Σ and with E, E' sets of unconditional equations, are called *equivalent*, denoted $(\Sigma, E) \cong (\Sigma, E')$ iff:

   - for each equation $\varphi' \in E'$, $E \vdash \varphi'$, and
   - for each equation $\varphi \in E$, $E' \vdash \varphi$.

   Prove the following results:

   - If $(\Sigma, E) \cong (\Sigma, E')$, then $(\Sigma, E) \cong (\Sigma, E \cup E') \cong (\Sigma, E')$
   - If $(\Sigma, E) \cong (\Sigma, E')$, then $\mathcal{T}_{\Sigma/E} = \mathcal{T}_{\Sigma/E'}$

2. Given an order-sorted equational theory (Σ, E), prove that for every Σ-equation $u = v$, we have the equivalence:

   $$\mathcal{T}_{\Sigma/E} \models u = v \quad \Leftrightarrow \quad \mathcal{T}_{\Sigma/E} = \mathcal{T}_{\Sigma/E \cup \{u=v\}}$$

   Exhibit a theory (Σ, E) and an equation $u = v$ such that $\mathcal{T}_{\Sigma/E} \models u = v$, but $(\Sigma, E) \not\cong (\Sigma, E \cup \{u = v\})$.

3. Consider the following module defining the cardinality of multisets, that you can download from the course web page:

```
fmod CARD is
  sorts Natural MSet .
  subsort Natural < MSet .
  op 0 : -> Natural [ctor] .
  op s : Natural -> Natural [ctor] .
  op _+_ : Natural Natural -> Natural .
  op __ : MSet MSet -> MSet [ctor assoc comm] .
  op card : MSet -> Natural .
  vars N M : Natural .
  vars U V : MSet .
  eq N + 0 = N .
  eq N + s(M) = s(N + M) .
  eq card(N U) = s(card(U)) .
endfm
```

   Do the following:

   - Use the Maude Sufficient Completeness Checker (SCC) to show that this module is not sufficiently complete. Use the information given by the SCC to complete the module, giving the most natural possible definition of cardinality in the missing case so as to cover all cases.
   - Use the SCC again to check that your corrected definition is now sufficiently complete.
   - Use the ITP to state and prove that the following equality holds inductively in your corrected module:

```
((card(U:MSet V:MSet)) = (card(U:MSet) + card(V:MSet)))
```

Note the fact that both the above equation as a whole, as well as the left side and right side of the equation, are enclosed in parentheses. This is to help the, somewhat primitive, parser of the ITP, which may not be able to parse a goal unless the equation in the goal has enough parentheses.

4. Consider the definition of binary trees with quoted identifiers on the leaves given in Lecture 13. Complete the module given below (available in the course web page) by defining with confluent and terminating equations the two functions called `leaves` and `inner`, that count, respectively, the number of leaf nodes of a tree, and the number of nodes in a tree that are *not* leaf nodes. For example, for the tree `(('a # 'b) # 'c) # 'd` there are 4 leaf nodes (namely `'a`, `'b`, `'c`, and `'d`), and 3 inner nodes (corresponding to the 3 different occurrences of the `#` operator).

```
fmod TREE is
  protecting QID .
  sorts Natural Tree .
  subsort Qid < Tree .
  op 0 : -> Natural [ctor].
  op s : Natural -> Natural [ctor].
  op _+_ : Natural Natural -> Natural [assoc comm].
  op _#_ : Tree Tree -> Tree [ctor] .
  ops leaves inner : Tree -> Natural .
  var I : Qid .
  vars N M : Natural .
  vars T T' : Tree .
  eq N + 0 = N .
  eq N + s(M) = s(N + M) .
  *** add equations for leaves and inner here
endfm
```

Once you have defined and tested your definitions for `leaves` and `inner` do the following, *including screenshots for each tool used* in your hardcopy of the homework:

- check that it is sufficiently complete using the SCC tool
- state a theorem, in the form of a universally quantified equation, that gives a general law stating, for any tree `T`, the exact relation between the numbers `leaves(T)` and `inner(T)`
- give a mechanical proof of that theorem using the ITP