Homework # 4
due March 18, 13:00

As with all homeworks, please turn in the written part (§3) of your homework at lecture and send the SASyLF proofs (§2) in `sound.slf` by email to `scmalte@inf.ethz.ch` before 1 pm.

1 Reading

Please read Chapter 8 in your textbook. We are skipping Chapter 6 since deBruijn indices are not needed in SASyLF (unlike Coq). We are skipping Chapter 7 and all ML implementation chapters.

Please do the following problems from the book:

8.3.5, 8.3.6 (page 98)

Do *not* turn in answers; check against back of book.

2 Proofs

Prove the following statements in SASyLF: 8.3.1–3. Put the proofs in `sound.slf` with your name in a comment at the top. The canonical forms lemma (8.3.1) requires a new judgment:

```
judgment canonical: t value : T

----------------- canonical-true
true value : Bool

------------------ canonical-false
false value : Bool

t numvalue
------------- canonical-num
t value : Nat
```

Use this lemma by doing inversion or case analysis on the result it gives.

3 Discussion

1. The rule E-PredSucc has a condition above the line. What happens if we remove this condition? Can we still prove progress and preservation? Explain!

   > If we remove the condition then E-PredSucc can apply even if the inner term is not fully evaluated; this makes evaluation nondeterministic, but does not affect progress (it makes a program *easier* to run) nor preservation, because if the resulting term is not typed as a `Nat`, then T-Succ couldn't apply either, so the input wouldn't be typed either.

2. The rule T-Pred has a condition above the line. What happens if we remove this condition? Can we still prove progress and preservation? Explain!

> The condition only affects typing, not evaluation. Unfortunately, without the condition, progress is no longer true, since `pred true` can be typed as `Nat` by the broken T-Pred.
>
> Preservation however is still true: we need to update the proof for the evaluation rules for `pred` t expressions:
>
> **E-Pred** The result can be typed again by T-Pred.
>
> **E-PredZero** The result is `0` which can be typed using T-Zero.
>
> **E-PredSucc** This evaluation rule only applies to `pred succ` t and evaluates to t where t is a numeric value. But if t is a numeric value, it has type `Nat`.

(You might find it helpful to try out what happens using the SASyLF proofs of program and preservation.)

4 Literature Survey

Find three papers in the literature that prove progress and preservation (or "subject reduction"). If possible, please reuse papers from you Homework #2.

If you recall, type systems are used to prevent certain kinds of errors, for example attempting to add things that aren't numbers, or applying somthing that is not a function.

What sort of errors are excluded by each of your chosen systems? Give the URL and the list of errors that are avoided. (Later assignments will continue to build on your choices.)

> Many people found the original question unclear. (In this solution I have expanded it to make clear what I want(ed).) So I have scaled back the expectations for this question (and the points).