Programming Languages — CPSC 604, spring 2009
Assignment 1

February 12, 2009

General rules

You can return your answers via *CSNet*, or **preferably** bring your answers to class. Hand-written answers are fine, answers typeset with, say, LaTeX are even finer. This is a perfect excuse to learn LaTeX if you do not know it already. If you submit via CSNet, submit just one file `answers.pdf`. If you really insist, `answers.doc` is OK. Returing your answer late has a penalty, computed as follows: $(0, 24h]$ late, penalty is 5%; $(24h, 48h]$ late, penalty is 10%; $(48h, 72h]$ late, penalty is 20%; $(72h, 96h]$ late, penalty is 40%; $(96h, \infty)$ late, penalty is 100%. Deadline is announced in class and/or on the class web pages.

Enjoy!

Assignment

1. Consider the following conjecture:

   Conjecture 1 (Consensus among Aggies)**.** *For any set of Aggies wearing a T-shirt, all Aggies in that set wear a T-shirt of the same color.*

   Is either of the two attempted proofs below correct? Explain what is wrong with Attempt 2.

   Attempt 1. They all wear Maroon anyway, thus the conjecture follows. □

   [**Answer**: This was a joke. Next time there will be a smiley after one :) – **end of answer**]

   Attempt 2. By induction on the number of Aggies (n) in the set, $n \geq 0$. $P(n) \doteq$ " For any set of n Aggies wearing a T-shirt, all Aggies in that set wear a T-shirt of the same color."

   Base cases: $n = 0, n = 1$. Trivial.

   Induction step: Consider the set $A = \{a_1, a_2, \ldots, a_{n+1}\}$ of $n+1$ Aggies, for $n \geq 1$. Let's denote the color of the T-shirt of Aggie a_i as $C(a_i)$. Let $A' = A \setminus \{a_{n+1}\}$. By induction hypothesis $P(|A'|)$, and thus $C(a_1) = C(a_n)$. Let $A'' = A \setminus \{a_1\}$. By induction hypothesis $P(|A''|)$, and thus $C(a_n) = C(a_{n+1})$. As $a_n \in A'$ and $a_n \in A''$, the color of the T-shirts in both sets A' and A'' is the same. Since $A = A' \cup A''$, this is also the color of all T-shirts in A. □

   [**Answer**: The base cases are proved for $|A| < 2$. Induction step should then take care of the case where $|A| = 2$. However, the induction step takes away one element of A and still assumes that the size of the remaining set is at least 2. – **end of answer**]

2. Consider the language $\mathcal{NB}$ we used in class:

`t`	::=		*terms:*
		`v`	*value*
		`if` $\mathtt{t_1}$ `then` $\mathtt{t_2}$ `else` $\mathtt{t_3}$	*conditional*
		`succ t`	*successor*
		`pred t`	*predecessor*
		`iszero t`	*test for zero*
`v`	::=		*values:*
		`true`	*constant true*
		`false`	*constant false*
		`nv`	*numeric value*
`nv`	::=		*numeric values:*
		`0`	*zero value*
		`succ nv`	*successor value*

and its evaluation relation:

E-Value
$$\mathtt{v} \Downarrow \mathtt{v}$$

E-IszeroZero
$$\frac{t \Downarrow \mathtt{0}}{\mathtt{iszero}\ t \Downarrow \mathtt{true}}$$

E-IszeroSucc
$$\frac{t \Downarrow \mathtt{succ}\ nv}{\mathtt{iszero}\ t \Downarrow \mathtt{false}}$$

E-PredZero
$$\frac{t \Downarrow \mathtt{0}}{\mathtt{pred}\ t \Downarrow \mathtt{0}}$$

E-PredSucc
$$\frac{t \Downarrow \mathtt{succ}\ nv}{\mathtt{pred}\ t \Downarrow nv}$$

E-Succ
$$\frac{t \Downarrow nv}{\mathtt{succ}\ t \Downarrow \mathtt{succ}\ nv}$$

E-IfTrue
$$\frac{t_1 \Downarrow \mathtt{true} \qquad t_2 \Downarrow v_2}{\mathtt{if}\ t_1\ \mathtt{then}\ t_2\ \mathtt{else}\ t_3\ \Downarrow v_2}$$

E-IfFalse
$$\frac{t_1 \Downarrow \mathtt{false} \qquad t_3 \Downarrow v_3}{\mathtt{if}\ t_1\ \mathtt{then}\ t_2\ \mathtt{else}\ t_3\ \Downarrow v_3}$$

Draw the derivation trees for the evaluation of the terms:

(a) `if (iszero (succ 0)) then 0 else succ 0`

(b) `pred if iszero 0 then (succ 0) else 0`

[**Answer**:

$$\cfrac{\cfrac{\mathtt{succ\ 0} \Downarrow \mathtt{succ\ 0}\ \textsc{E-Value}}{\mathtt{iszero\ (succ\ 0)} \Downarrow \mathtt{false}}\ \textsc{E-IszeroSucc} \qquad \mathtt{succ\ 0} \Downarrow \mathtt{succ\ 0}\ \textsc{E-Value}}{\mathtt{if\ (iszero\ (succ\ 0))\ then\ 0\ else\ succ\ 0} \Downarrow \mathtt{succ\ 0}}\ \textsc{E-IfFalse}$$

$$\cfrac{\cfrac{\cfrac{\mathtt{0} \Downarrow \mathtt{0}\ \textsc{E-Value}}{\mathtt{iszero\ 0} \Downarrow \mathtt{true}}\ \textsc{E-IszeroZero} \qquad \mathtt{succ\ 0} \Downarrow \mathtt{succ\ 0}\ \textsc{E-IsValue}}{\mathtt{if\ iszero\ 0\ then\ (succ\ 0)\ else\ 0} \Downarrow \mathtt{succ\ 0}}\ \textsc{E-IfTrue}}{\mathtt{pred\ if\ iszero\ 0\ then\ (succ\ 0)\ else\ 0} \Downarrow \mathtt{0}}\ \textsc{E-PredSucc}$$

– **end of answer**]

Write a term

(a) that contains `iszero` and gets stuck (that is, does not evaluate to any term)

(b) that contains an `if` statement and gets stuck

[**Answer**:

`iszero false`

`if 0 then 0 else 0`

– **end of answer**]

Write a term that does not get stuck, but contains, as subterms, one or both of the terms you wrote for the previous exercise (that do get stuck).

[**Answer**:

`if true then 0 else iszero false`

– **end of answer**]

3. Consider the language $\mathcal{B}$ which we defined in class, consisting of booleans and a conditional expression. Replace the rules B-IfTrue and B-IfFalse with the following rules:

$$\textsc{B-IfTrue'}\quad \frac{t_1 \Downarrow \texttt{true} \qquad t_2 \Downarrow t_2' \qquad t_3 \Downarrow t_3'}{\texttt{if } t_1 \texttt{ then } t_2 \texttt{ else } t_3 \ \Downarrow t_2'} \qquad\qquad \textsc{B-IfFalse'}\quad \frac{t_1 \Downarrow \texttt{false} \qquad t_2 \Downarrow t_2' \qquad t_3 \Downarrow t_3'}{\texttt{if } t_1 \texttt{ then } t_2 \texttt{ else } t_3 \ \Downarrow t_3'}$$

Does the evaluation relation stay the same? Prove or disprove.

[**Answer**: Yes, the relation stays the same. Let's call the modified language $\mathcal{B}'$ and the denote the evaluation relation in it with $\Downarrow'$. We need to prove the following:

Theorem 1. *$t \Downarrow t'$ in language $\mathcal{B}$ if and only if $t \Downarrow' t'$ in language $\mathcal{B}'$.*

Proof.

- From left to right: By induction on the derivation $t \Downarrow t'$. Holds vacuously if the last rule used was B-True or B-False. If the last rule used was B-IfTrue, then $t = \texttt{if true then } t_2 \texttt{ else } t_3 \Downarrow t_2'$ and $t_2 \Downarrow t_2'$. By induction hypothesis $t_2 \Downarrow' t_2'$, and thus we need to show that $t_3 \Downarrow t_3'$ for some t_3'. It does, because $\Downarrow$ is total (we showed that in class), and thus $t \Downarrow' t'$ by the rule B-IfFalse'. The case of B-IfFalse similarly.
- From right to left: By induction on the derivation $t \Downarrow' t'$. Holds vacuously if the last rule used was B-True or B-False. If the last rule used was B-IfTrue', then $t = \texttt{if true then } t_2 \texttt{ else } t_3 \Downarrow' t_2'$ and $t_2 \Downarrow' t_2'$ and $t_3 \Downarrow' t_3'$. By induction hypothesis $t_2 \Downarrow t_2'$, and thus $t \Downarrow t'$ by the rule B-IfFalse. The case of B-IfFalse similarly.

□

– **end of answer**]

4. Consider the language $\mathcal{NB}$ and apply the same modifications as in exercise 3 to $\mathcal{NB}$; that is, replace the rules E-IfTrue and E-IfFalse with the following rules:

$$\textsc{E-IfTrue'}\quad \frac{t_1 \Downarrow \texttt{true} \qquad t_2 \Downarrow v_2 \qquad t_3 \Downarrow v_3}{\texttt{if } t_1 \texttt{ then } t_2 \texttt{ else } t_3 \ \Downarrow v_2} \qquad\qquad \textsc{E-IfFalse'}\quad \frac{t_1 \Downarrow \texttt{false} \qquad t_2 \Downarrow v_2 \qquad t_3 \Downarrow v_3}{\texttt{if } t_1 \texttt{ then } t_2 \texttt{ else } t_3 \ \Downarrow v_3}$$

Does the evaluation relation stay the same? Prove or disprove.

[**Answer**: No, it does not stay the same. Let $t =$ `if true then 0 else (iszero true)`. Using again $\Downarrow'$ for the new evaluation relation, $t \Downarrow$ `0` but $t \not\Downarrow'$ `0`; t is a stuck term in the new evaluation relation. – **end of answer**]