Homework 5: Semantics of Arithmetic Expressions

CIS 352: Programming Languages

18 February 2019, Version 2

Administrivia

- You **may** work in pairs on this assignment.
- *However*, to get some practice for future quizzes, everyone should work on the first three problems on their own.
- For Part I, hand written answers are fine, ***but make them readable.***
- For Part II, copy all the files in `http://www.cis.syr.edu/courses/cis352/code/L1/` and use `eval1.hs` as your starter file.
- **To turn Part I:** Place your papers in the CIS 352 box on the 4th floor of SciTech by the due date.
- **To turn Part II:** Submit via Blackboard *(i)* the source files for Part II, *(ii)* the transcripts of test runs, and *(iii)* the cover sheet.

Grading Criteria

- The homework is out of 100 points.
- Each programming problem is $\approx$ 70% correctness and $\approx$ 30% testing.
- Omitting your name(s) in the source code looses you 5 points.

Typo corrections in red.

Part I: Problems on Paper

Fair warning: Questions like Problems 1, 2, and 3 *will* show up on quizzes.

❖ *Problem 1 (18 points)* ❖

Give a complete big-step derivation of each of the following.[1]

(a) $(8 + (1 * 3)) \Downarrow 11$

(b) $((8 + 1) * 3) \Downarrow 27$

(c) $(((6 - 4) * (1 + 4)) - 8) \Downarrow 2.$

[1] The big-step evaluation rules are given in Appendices on page 3 and **Sample 1** gives a sample big-step derivation.

❖ *Problem 2 (18 points)* ❖

Give a complete small-step derivation of each transition below.[2]

(a) $((5 * 3) - 2) \;\rightarrow\; (15 - 2)$

(b) $((7 - (2 * 3))/4) \;\rightarrow\; ((7 - 6)/4)$

(c) $((((5 - 2) * 3) + 12) - (2 * 6)) \;\rightarrow\; (((3 * 3) + 12) - (2 * 6))$

[2] The small-step evaluation rules are given in Appendices on page 3 and **Sample 2** gives a sample small-step derivation.

❖ *Problem 3 (14 points)* ❖

Give a complete transition sequence to a value for each of the following expressions.[3]

(a) $(((7 * 2) - 6)/4) \;\rightarrow^*\; 2$

(b) $((((5 - 2) * 3) + 12) - (2 * 6)) \;\rightarrow^*\; 9$

[3] **Notes:**

- The small-step evaluation rules are given in Appendices on page 3 and **Sample 3** gives a sample complete small-step transition sequence.
- You *do not* need to give a small-step derivation for each step in the transition sequence. small-step derivation.
- **Reminder:** $\rightarrow^*$ means many $\rightarrow$-steps (including 0-steps).

❖ ***Problem 4 (20 points)*** ❖
Suppose we add the following new sort of arithmetic expression to our language:

$$(\mathrm{E}_1 \ ?\ \mathrm{E}_2 : \mathrm{E}_3)$$

This expression is based on the conditional expression from the C programming language, whose evaluation Kernighan and Ritchie describe as follows[4]:

> In the expression
>
> $$\mathit{expr}_1 \ ?\ \mathit{expr}_2 \ :\ \mathit{expr}_3$$
>
> the expression expr_1 is evaluated first. If it is non-zero . . . , then the expression expr_2 is evaluated, and that is the value of the conditional expression. Otherwise expr_3 is evaluated, and that is the value. Only one of expr_2 and expr_3 is evaluated.

[4] Brian W. Kernighan and Dennis M. Ritchie. *The C Programming Language*, 2nd ed. Prentice Hall Software Series, 1988, page 51.

Example:
$(10 \ ?\ 6 * 5 : 17) \rightsquigarrow 30$
$(\ 0 \ ?\ 6 * 5 : 17) \ \rightsquigarrow 17$

(a) Extend the definition of $\Downarrow$ to account for conditional expressions of this form.[5]

(b) Using your new rule(s), give a formal derivation of:

$$(\ (23+7)\ ?\ (9-4)\ :\ (3/0)\ ) \Downarrow 5$$

Explain why the evaluation does *not* cause a divide-by-zero error.

[5] **Giant hint:** Figure out how to fill in the blanks (i.e., the ???'s) in the partial definitions of $COND_0$ and $COND_1$ in Appendix A below.

Part II: Programming Problems

This part consists of two small extensions of `eval` in the `eval1.hs` file. You need to construct a reasonable set of tests for both extensions.

❖ ***Problem 5 (10 points)*** ❖

(a) ***(6 pts)*** Extend the definition of `eval` to handle division per the big step rules. Note that for `(Div a1 a2)`, if `a1` evaluates to v_1 and `a2` evaluates to $v_2 \neq 0$, then the value of `(Div a1 a2)` should be (`div` v_1 v_2) where `div` is the standard Haskell integer division function. In the case were you have a division by 0, supply your own error message.

(b) ***(4 pts)*** Devise and run a reasonable set of tests for this extension.

❖ ***Problem 6 (20 points)*** ❖

(a) ***(12 pts)*** Extend the definition of `eval` to handle conditional expressions per your answer to Problem 4 above. Be sure that no division by 0 error occurs when evaluating either of:

- `(1 ? 10 : (1/0))`
- `(0 ? (1/0) : 20))`

(b) ***(8 pts)*** Devise and run a reasonable set of tests for this extension.

Part III: Challenge Problems
No points, just glory.

✪ ***Challenge Problem 1: Craft rules.*** ✪
Provide reasonable small-step rules for conditional expressions.

✪ ***Challenge Problem 2: Automate.*** ✪
Automate the construction of small-step derivations and complete transition sequences.

Appendices

Key
a: an arithmetic expression
v: a numeric value

Reference: Big Step Rules

$$\textit{PLUS: } \frac{a_1 \Downarrow v_1 \quad a_2 \Downarrow v_2}{(a_1 + a_2) \Downarrow v} \ (v = v_1 + v_2)$$

$$\textit{MINUS: } \frac{a_1 \Downarrow v_1 \quad a_2 \Downarrow v_2}{(a_1 - a_2) \Downarrow v} \ (v = v_1 - v_2)$$

$$\textit{MULT: } \frac{a_1 \Downarrow v_1 \quad a_2 \Downarrow v_2}{(a_1 * a_2) \Downarrow v} \ (v = v_1 * v_2)$$

$$\textit{DIV: } \frac{a_1 \Downarrow v_1 \quad a_2 \Downarrow v_2}{(a_1 / a_2) \Downarrow v} \ \begin{pmatrix} v_2 \neq 0 \ \& \\ v = \lfloor v_1 / v_2 \rfloor \end{pmatrix}$$

$$\textit{NUM: } \frac{}{v \Downarrow v} \ (\mathcal{N}[\![n]\!] = v)$$

$$\textit{COND}_0\textit{: } \frac{??? \quad ???}{(a_1 ? a_2 : a_3) \Downarrow v} \ \left(??? \right)$$

$$\textit{COND}_1\textit{: } \frac{??? \quad ???}{(a_1 ? a_2 : a_3) \Downarrow v} \ \left(??? \right)$$

Sample 1: A big-step derivation

$$\textit{MULT: } \dfrac{\textit{PLUS: } \dfrac{\textit{NUM: } \dfrac{}{2 \Downarrow 2} \quad \textit{NUM: } \dfrac{}{5 \Downarrow 5}}{(2+5) \Downarrow 7} \ (2+5=7) \qquad \textit{NUM: } \dfrac{}{13 \Downarrow 13}}{((2+5) * 13) \Downarrow 91} \ (7 * 13 = 91)$$

Reference: Small Step Rules

$$\textit{plus}_1\textit{: } \frac{a_1 \to a_1'}{(a_1 + a_2) \to (a_1' + a_2)} \qquad \textit{plus}_2\textit{: } \frac{a_2 \to a_2'}{(v_1 + a_2) \to (v_1 + a_2')} \qquad \textit{plus}_3\textit{: } \frac{}{(v_1 + v_2) \to v} \ (v = v_1 + v_2)$$

$$\textit{minus}_1\textit{: } \frac{a_1 \to a_1'}{(a_1 - a_2) \to (a_1' - a_2)} \qquad \textit{minus}_2\textit{: } \frac{a_2 \to a_2'}{(v_1 - a_2) \to (v_1 - a_2')} \qquad \textit{minus}_3\textit{: } \frac{}{(v_1 - v_2) \to v} \ (v = v_1 - v_2)$$

$$\textit{mult}_1\textit{: } \frac{a_1 \to a_1'}{(a_1 * a_2) \to (a_1' * a_2)} \qquad \textit{mult}_2\textit{: } \frac{a_2 \to a_2'}{(v_1 * a_2) \to (v_1 * a_2')} \qquad \textit{mult}_3\textit{: } \frac{}{(v_1 * v_2) \to v} \ (v = v_1 * v_2)$$

$$\textit{div}_1\textit{: } \frac{a_1 \to a_1'}{(a_1 / a_2) \to (a_1' / a_2)} \qquad \textit{div}_2\textit{: } \frac{a_2 \to a_2'}{(v_1 / a_2) \to (v_1 / a_2')} \qquad \textit{div}_3\textit{: } \frac{}{(v_1 / v_2) \to v} \ \begin{pmatrix} v_1 \neq 0 \quad \& \\ v = \lfloor v_1 / v_2 \rfloor \end{pmatrix}$$

Sample 2: A small-step derivation

$$\textit{mult}_1\textit{: } \dfrac{\textit{plus}_2\textit{: } \dfrac{\textit{minus}_3\textit{: } \dfrac{}{(8-3) \to 5} \ (5 = 8 - 3)}{(6 + (8-3)) \to (6+5)}}{((6 + (8-3)) * (5-2)) \to ((6+5) * (5-2))}$$

Sample 3: A complete small-step transition sequence

$$((6 + (8-3)) * (5-2)) \ \to \ ((6+5) * (5-2)) \ \to \ 11 * (5-2) \ \to \ 11 * 3 \ \to \ 33$$