Homework 5: Semantics of Arithmetic Expressions

The big-step and small-step evaluation rules are given in Appendices A and B at the end of the writeup. Unlike the rules on some of the class slides, the rules Appendices A and B conflate numerals and integer-values to cut some clutter in derivations.

N.B. Questions like Problems 1, 2, and 3 will show up on quizzes.

Part I: Problems on Paper

Problem 1: (18 points).
Give a complete big-step derivation of each of the following.

(a) \((1 + (3 \times 5)) \downarrow 16\)
(b) \(((1 + 3) \times 5) \downarrow 20\)
(c) \((25 - ((4 + 3) \times (2 + 1))) \downarrow 4\).

Problem 2: (18 points).
Give a complete small-step derivation of each transition below.

(a) \(((10 - 7) \times (1 + 1)) \rightarrow (3 \times (1 + 1))\)
(b) \(((10 - (4 + 3)) \times (1 + 1)) \rightarrow ((10 - 7) \times (1 + 1))\)
(c) \(((10 - ((2 \times 2) + 3)) \times (1 + 1)) \rightarrow ((10 - (4 + 3)) \times (1 + 1))\)

Problem 3: (14 points).
Give a complete transition sequent to a value for each of the following expressions. You do not need to give a small-step derivation for each step.

(a) \((3 \times 4 + 2) \rightarrow^* 14\)
(b) \(((10 - ((2 \times 2) + 3)) \times (1 + 1)) \rightarrow^* 6\)

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

\[(E_1 ? E_2 : E_3)\]

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

“In the expression

\[expr_1 \? expr_2 : expr_3\]

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.”

Example: Evaluating \((10 \? 6 \times 5 : 17)\) should yield 30; whereas evaluating

\((0 \? 6 \times 5 : 17)\) should yield 17.

(a) Extend the definition of \(\downarrow\) to account for conditional expressions of this form. (Giant hint: Figure out how to fill in the blanks (i.e., the ???’s) in the partial definitions of \(\text{COND}_0\) and \(\text{COND}_1\) in Appendix A below.)

(b) Using your new rule(s), give a formal derivation for the following (where \(s_0\) is the same state as in previous questions):

\(( (22 + 3) \? (8 - 3) : (2/0) ) \downarrow 5\)

Explain why evaluating this does not cause a divide-by-zero error.

Part II: Programming Problems

This part consists of two modest extensions of eval in the eval1.hs file. You are responsible for a reasonable set of tests for both extensions.

Problem 5: (10 points).

(a) (6 points) Extend the definition of eval to handle division per the big step rules. Note that for (Div a1 a2), if a1 evaluates to v1 and a2 evaluates to v2 \( \neq 0 \), then the value of (Div a1 a2) should be \((\text{div} v_1 v_2)\) where \text{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 points) Devise and run a reasonable set of tests for this extension.

Problem 6: (20 points).

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

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

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

Part III: Challenge Problems

Challenge Problem 1: (No points, just glory).

Provide reasonable small-step rules for conditional expressions.

Challenge Problem 2: (No points, just glory).

Automate the construction of small-step derivations and complete transition sequences.

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 just fine. Unreadable answers will not be graded.
- For Part II, use eval1.hs as your starter file.
- You loose 5 points for omitting your name on your papers/programs.
- 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
  Via Blackboard, turn in
  (i) the source files for Part II,
  (ii) the transcripts of test runs, and
  (iii) the cover sheet.
### Appendices

**Key**
- $a$: an arithmetic expression
- $v$: a numeric value

#### §A. Reference: Big Step Rules

<table>
<thead>
<tr>
<th>Operation</th>
<th>Big Step Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PLUS</strong></td>
<td>$a_1 \downarrow v_1$ $a_2 \downarrow v_2$ $(v = v_1 + v_2)$</td>
</tr>
<tr>
<td><strong>MINUS</strong></td>
<td>$a_1 \downarrow v_1$ $a_2 \downarrow v_2$ $(v = v_1 - v_2)$</td>
</tr>
<tr>
<td><strong>MULT</strong></td>
<td>$a_1 \downarrow v_1$ $a_2 \downarrow v_2$ $(v = v_1 \cdot v_2)$</td>
</tr>
<tr>
<td><strong>DIV</strong></td>
<td>$a_1 \downarrow v_1$ $a_2 \downarrow v_2$ $(v = \frac{v_1}{v_2})$</td>
</tr>
<tr>
<td><strong>NUM</strong></td>
<td>$v \downarrow v$</td>
</tr>
</tbody>
</table>

**COND**
- $a_1?a_2:a_3 \downarrow v$ $(???)$
- $a_1?a_2:a_3 \downarrow v$ $(???)$

**A sample big-step derivation**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PLUS</strong></td>
<td>$2 \downarrow 2$ $5 \downarrow 5$ $(2 + 5 = 7)$</td>
</tr>
</tbody>
</table>
| **MULT** | $2 \downarrow 2$ $5 \downarrow 5$ $7 \downarrow 7$ $(2 + 5) \downarrow 7$ $(2 \cdot 5) \downarrow 10$ $(7 \cdot 13) \downarrow 91$ 

### §B. Reference: Small Step Rules

<table>
<thead>
<tr>
<th>Operation</th>
<th>Small Step Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PLUS</strong></td>
<td>$a_1 \rightarrow a_1'$ $(a_1 + a_2) \rightarrow (a_1' + a_2)$</td>
</tr>
<tr>
<td><strong>MINUS</strong></td>
<td>$a_1 \rightarrow a_1'$ $(a_1 - a_2) \rightarrow (a_1' - a_2)$</td>
</tr>
<tr>
<td><strong>MULT</strong></td>
<td>$a_1 \rightarrow a_1'$ $(a_1 \cdot a_2) \rightarrow (a_1' \cdot a_2)$</td>
</tr>
<tr>
<td><strong>DIV</strong></td>
<td>$a_1 \rightarrow a_1'$ $(a_1 / a_2) \rightarrow (a_1' / a_2)$</td>
</tr>
</tbody>
</table>

**A sample small-step derivation**

<table>
<thead>
<tr>
<th>Transition</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MINUS</strong></td>
<td>$(8 - 3) \rightarrow 5$ $(5 = 8 - 3)$</td>
</tr>
<tr>
<td><strong>PLUS</strong></td>
<td>$(6 + (8 - 3)) \rightarrow (6 + 5)$</td>
</tr>
<tr>
<td><strong>MULT</strong></td>
<td>$((6 + (8 - 3)) \rightarrow (6 + 5)) \rightarrow ((6 + 5) \cdot (5 - 2))$</td>
</tr>
</tbody>
</table>

**A sample complete (small-step) transition sequence**

$((6 + (8 - 3)) \cdot (5 - 2)) \rightarrow ((6 + 5) \cdot (5 - 2)) \rightarrow 11 \cdot (5 - 2) \rightarrow 11 \cdot 3 \rightarrow 33$