HW 3 Solution – Evaluation with Context Semantics

CS 422 – Spring 2016
Revision 1.0

Assigned February 11, 2016
Due February 18, 2016, 8:00pm
Extension 24 hours (10% penalty)

1 Change Log

1.1 Made the names of the various types of context uniform; corrected the mistakes in the names of the contexts in Substitutions in Boolean Contexts.

1.0 Initial Release.

2 Turn-In Procedure

Answer the following problems, save you work as a PDF named `hw3-submission.pdf` (either scanned if hand-written, or converted from a program), and submit your PDF by first adding it to your svn repository (`svn add hw3-submission.pdf`) and commit it (`svn commit -m "submitting hw3"`). Your file contain your name, netid and solution. It should be named `hw3-submission.pdf` and committed in your `assignments/hw3` directory.

3 Objectives

The purpose of this HW is to familiarize using Reduction Semantics with Evaluation Contexts (Context Semantics) for calculating the evaluation of a SIMPL1 program.

4 Background

In this assignment you will be giving all the steps of evaluation of a short program in SIMPL1 using Reduction Semantics with Evaluation Contexts. As background, we will review the syntax and Context Semantics for SIMPL1 and include the example worked in class. You are expectd to give your answer with the same level of detail.

4.1 Reduction Semantics with Evaluation Contexts (Context Semantics) for SIMPL1

4.1.1 Syntax

Arithmetic Expressions:

$$
\begin{array}{rcl}
I & \in & \textit{Identifiers} \\
N & \in & \textit{Numerals} \\
E & ::= & N \mid I \mid E + E \mid E * E \mid E - E \mid EECxt[E] \\
EECxt & ::= & \Box_{\mathbf{E}} \mid EECxt + E \mid N + EECxt \mid EECxt * E \mid N * EECxt \mid EECxt - E \mid N - EECxt
\end{array}
$$

Boolean Expressions

$$
\begin{array}{rcl}
B & ::= & \mathsf{true} \mid \mathsf{false} \mid B \,\&\, B \mid B \;\mathsf{or}\; B \mid \mathsf{not}\; B \mid E < E \mid E = E \mid BECxt[E] \mid BBCxt[B] \\
BECxt & ::= & \mid BECxt \,\&\, B \mid BECxt \;\mathsf{or}\; B \mid \mathsf{not}\; BECxt \mid EECxt < E \mid N < EECxt \mid EECxt = E \mid N = EECxt \\
BBCxt & ::= & \Box_{\mathbf{B}} \mid BBCxt \,\&\, B \mid BBCxt \;\mathsf{or}\; B \mid \mathsf{not}\; BBCxt
\end{array}
$$

Commands

$$
\begin{array}{rcl}
C & ::= & \mathsf{skip} \mid C; C \mid I ::= E \mid \mathsf{if}\; B \;\mathsf{then}\; C \;\mathsf{else}\; C \;\mathsf{fi} \\
 & \mid & \{C\} \mid \mathsf{while}\; B \;\mathsf{do}\; C \;\mathsf{od} \mid CECxt[E] \mid CBCxt[B] \mid CCCxt[C] \\
CECxt & ::= & CECxt; C \mid I := EECxt \mid \mathsf{if}\; BECxt \;\mathsf{then}\; C \;\mathsf{else}\; C \;\mathsf{fi} \\
CBCxt & ::= & CBCxt; C \mid \mathsf{if}\; BBCxt \;\mathsf{then}\; C \;\mathsf{else}\; C \;\mathsf{fi} \\
CCCxt & ::= & \Box_{\mathbf{C}} \mid CCCxt; C
\end{array}
$$

4.1.2 Equational Rules:

Arithmetic Contexts:

$$
\begin{array}{l}
\Box_{\mathbf{E}}[E] = E \\
(EECxt \oplus E)[E'] = EECxt[E'] \oplus E \\
(N \oplus EECxt)[E] = N \oplus (EECxt[E])
\end{array}
$$

Substitution in Boolean Contexts:

$$
\begin{array}{ll}
(EECxt \sim E)[E'] = EECxt[E'] \sim E & \Box_{\mathbf{B}}[B] = B \\
(N \sim EECxt)[E] = N \sim (EECxt[E]) & (BBCxt \& B)[B'] = BBCxt[B'] \& B \\
(BECxt \& B)[E] = BECxt[E] \& B & (BBCxt \;\mathsf{or}\; B)[B'] = BBCxt[B'] \;\mathsf{or}\; B \\
(BECxt \;\mathsf{or}\; B)[E] = BECxt[E] \;\mathsf{or}\; B & (\mathsf{not}\; BBCxt)[B'] = \mathsf{not}\; (BBCxt[B]) \\
(\mathsf{not}\; BECxt)[E] = \mathsf{not}\; (BECxt[E]) &
\end{array}
$$

Substitution in Command Contexts:

$$
\begin{array}{l}
(CECxt; C)[E] = CECxt[E]; C \\
(I := EECxt)[E] = I := (EECxt[E])
\end{array}
$$

$$
(\mathsf{if}\; BECxt \;\mathsf{then}\; C \;\mathsf{else}\; C \;\mathsf{fi})[E] = \mathsf{if}\; (BECxt[E]) \;\mathsf{then}\; C \;\mathsf{else}\; C \;\mathsf{fi}
$$

$$
\begin{array}{l}
(CBCxt; C)[B] = CBCxt[B]; C \\
(\mathsf{if}\; BBCxt \;\mathsf{then}\; C \;\mathsf{else}\; C \;\mathsf{fi})[B] = \mathsf{if}\; BBCxt[B] \;\mathsf{then}\; C \;\mathsf{else}\; C \;\mathsf{fi}
\end{array}
$$

$$
\begin{array}{l}
\Box_{\mathbf{C}}[C] = C \\
(CCCxt; C)[C'] = CCCxt[C']; C
\end{array}
$$

4.1.3 Transitions

Arithmetic Expressions

$$(I, m) \longrightarrow (m(I), m)$$
$$(N \oplus N', m) \longrightarrow (N'', m) \quad \text{if } N \oplus N' = N''$$

Booleans

$$(N \sim N', m) \longrightarrow (b, m) \qquad \text{if } N \sim N' = b$$

$$(\mathsf{true} \ \& \ B, m) \longrightarrow (B, m) \qquad (\mathsf{false} \ \& \ B, m) \longrightarrow (\mathsf{false}, m)$$
$$(\mathsf{true \ or} \ B, m) \longrightarrow (\mathsf{true}, m) \qquad (\mathsf{false \ or} \ B, m) \longrightarrow (B, m)$$
$$(\mathsf{not \ true}, m) \longrightarrow (\mathsf{false}, m) \qquad (\mathsf{not \ false}, m) \longrightarrow (\mathsf{true}, m)$$

Commands

$$(I ::= V, m) \longrightarrow (\mathsf{skip}, m[I \leftarrow V])$$

$$(\mathsf{skip} : C, m) \longrightarrow (C, m)$$

$$(\{C\}, m) \longrightarrow (C, m)$$

$$(\mathsf{if \ true \ then} \ C \ \mathsf{else} \ C' \ \mathsf{fi}, m) \longrightarrow (C, m)$$
$$(\mathsf{if \ false \ then} \ C \ \mathsf{else} \ C' \ \mathsf{fi}, m) \longrightarrow (C', m)$$

$$(\mathsf{while} \ B \ \mathsf{do} \ C, m) \longrightarrow (\mathsf{if} \ B \ \mathsf{then} \ C; \mathsf{while} \ B \ \mathsf{do} \ C \ \mathsf{else \ skip \ fi}, m)$$

4.2 Example From Class

$(\mathsf{y := i; while\ i > 3\ do\ \{i := i - 1; y := y * i\}\ od}, \langle \mathsf{i} \mapsto 2 \rangle)$

$= ((\square_{\mathbf{C}}\mathsf{; while\ i > 3\ do\ \{i := i - 1; y := y * i\}\ od})[\mathsf{y := i}], \langle \mathsf{i} \mapsto 2 \rangle)$

$= ((\square_{\mathbf{C}}\mathsf{; while\ i > 3\ do\ \{i := i - 1; y := y * i\}\ od})[\mathsf{(y :=}\square_{\mathbf{E}}\mathsf{)[i]}], \langle \mathsf{i} \mapsto 2 \rangle)$

$\longrightarrow ((\square_{\mathbf{C}}\mathsf{; while\ i > 3\ do\ \{i := i - 1; y := y * i\}\ od})[\mathsf{(y :=}\square_{\mathbf{E}}\mathsf{)[2]}], \langle \mathsf{i} \mapsto 2 \rangle)$

since $((\mathsf{y :=}\square_{\mathbf{E}}\mathsf{)[i]}, \langle \mathsf{i} \mapsto 2 \rangle) \longrightarrow ((\mathsf{y :=}\square_{\mathbf{E}}\mathsf{)[2]}, \langle \mathsf{i} \mapsto 2 \rangle)$

since $(\mathsf{i}, \langle \mathsf{i} \mapsto 2 \rangle) \longrightarrow (2, \langle \mathsf{i} \mapsto 2 \rangle)$

$= ((\square_{\mathbf{C}}\mathsf{; while\ i > 3\ do\ \{i := i - 1; y := y * i\}\ od})[\mathsf{y := 2}], \langle \mathsf{i} \mapsto 2 \rangle)$

$\longrightarrow ((\square_{\mathbf{C}}\mathsf{; while\ i > 3\ do\ \{i := i - 1; y := y * i\}\ od})[\mathsf{skip}], \langle \mathsf{y} \mapsto 2, \mathsf{i} \mapsto 2 \rangle)$

since $(\mathsf{y := 2}, \langle \mathsf{i} \mapsto 2 \rangle) \longrightarrow (\mathsf{skip}, \langle \mathsf{y} \mapsto 2, \mathsf{i} \mapsto 2 \rangle)$

$= ((\mathsf{skip; while\ i > 3\ do\ \{i := i - 1; y := y * i\}\ od}), \langle \mathsf{y} \mapsto 2, \mathsf{i} \mapsto 2 \rangle)$

$\longrightarrow (\mathsf{while\ i > 3\ do\ \{i := i - 1; y := y * i\}\ od}, \langle \mathsf{y} \mapsto 2, \mathsf{i} \mapsto 2 \rangle)$

$\longrightarrow (\mathsf{if\ i > 3\ then\ \{i := i - 1; y := y *i\}; while\ i > 3\ do\ \{i := i - 1; y := y * i\}\ od\ else\ skip\ fi}, \langle \mathsf{y} \mapsto 2, \mathsf{i} \mapsto 2 \rangle)$

$= ((\mathsf{if}\ \square_{\mathbf{B}}\ \mathsf{then\ \{i := i - 1; y := y *i\}; while\ i > 3\ do\ \{i := i - 1; y := y * i\}\ od\ else\ skip\ fi})[\mathsf{i > 3}], \langle \mathsf{y} \mapsto 2, \mathsf{i} \mapsto 2 \rangle)$

$= ((\mathsf{if}\ \square_{\mathbf{B}}\ \mathsf{then\ \{i := i - 1; y := y *i\}; while\ i > 3\ do\ \{i := i - 1; y := y * i\}\ od\ else\ skip\ fi})[(\square_{\mathbf{E}} > 3)[\mathsf{i}]], \langle \mathsf{y} \mapsto 2, \mathsf{i} \mapsto 2 \rangle)$

$\longrightarrow ((\mathsf{if}\ \square_{\mathbf{B}}\ \mathsf{then\ \{i := i - 1; y := y *i\}; while\ i > 3\ do\ \{i := i - 1; y := y * i\}\ od\ else\ skip\ fi})[(\square_{\mathbf{E}} > 3)[2]], \langle \mathsf{y} \mapsto 2, \mathsf{i} \mapsto 2 \rangle)$

since $((\square_{\mathbf{E}} > 3)[\mathsf{i}], \langle \mathsf{y} \mapsto 2, \mathsf{i} \mapsto 2 \rangle) \longrightarrow ((\square_{\mathbf{E}} > 3)[2], \langle \mathsf{y} \mapsto 2, \mathsf{i} \mapsto 2 \rangle)$

since $(\mathsf{i}, \langle \mathsf{y} \mapsto 2, \mathsf{i} \mapsto 2 \rangle) \longrightarrow (2, \langle \mathsf{y} \mapsto 2, \mathsf{i} \mapsto 2 \rangle)$

$= ((\mathsf{if}\ \square_{\mathbf{B}}\ \mathsf{then\ \{i := i - 1; y := y *i\}; while\ i > 3\ do\ \{i := i - 1; y := y * i\}\ od\ else\ skip\ fi})[\mathsf{2 > 3}], \langle \mathsf{y} \mapsto 2, \mathsf{i} \mapsto 2 \rangle)$

$\longrightarrow ((\mathsf{if}\ \square_{\mathbf{B}}\ \mathsf{then\ \{i := i - 1; y := y *i\}; while\ i > 3\ do\ \{i := i - 1; y := y * i\}\ od\ else\ skip\ fi})[\mathsf{false}], \langle \mathsf{y} \mapsto 2, \mathsf{i} \mapsto 2 \rangle)$

since $(\mathsf{2 > 3}, \langle \mathsf{y} \mapsto 2, \mathsf{i} \mapsto 2 \rangle) \longrightarrow (\mathsf{false}, \langle \mathsf{y} \mapsto 2, \mathsf{i} \mapsto 2 \rangle)$

$= (\mathsf{if\ false\ then\ \{i := i - 1; y := y *i\}; while\ i > 3\ do\ \{i := i - 1; y := y * i\}\ od\ else\ skip\ fi}, \langle \mathsf{y} \mapsto 2, \mathsf{i} \mapsto 2 \rangle)$

$\longrightarrow (\mathsf{skip}, \langle \mathsf{y} \mapsto 2, \mathsf{i} \mapsto 2 \rangle)$

5 Problems

1. Give all the steps of evaluation, including equational rewrites and justifications for transitions where needed, using the above Reduction Semantics with Evaluation Contexts in SIMPL1 for

$$(\mathsf{while\ 0\ <\ y\ do\ y\ :=\ 0\ -\ y\ od}, \langle \mathsf{y} \mapsto 2 \rangle)$$

Solution:

$$
\begin{aligned}
&(\text{while } 0 < \text{y do y} := 0 - \text{y od}, \langle \text{y} \mapsto 2 \rangle)\\
&\longrightarrow (\text{if } 0 < \text{y then y} := 0 - \text{y; while } 0 < \text{y do y} := 0 - \text{y od else skip fi}, \langle \text{y} \mapsto 2 \rangle)\\
&= ((\text{if } \square_{\mathbf{B}} \text{ then y} := 0 - \text{y; while } 0 < \text{y do y} := 0 - \text{y od else skip fi})[0 < \text{y}], \langle \text{y} \mapsto 2 \rangle)\\
&= ((\text{if } \square_{\mathbf{B}} \text{ then y} := 0 - \text{y; while } 0 < \text{y do y} := 0 - \text{y od else skip fi})[(0 < \square_{\mathbf{E}})[\text{y}]], \langle \text{y} \mapsto 2 \rangle)\\
&\longrightarrow ((\text{if } \square_{\mathbf{B}} \text{ then y} := 0 - \text{y; while } 0 < \text{y do y} := 0 - \text{y od else skip fi})[(0 < \square_{\mathbf{E}})[2]], \langle \text{y} \mapsto 2 \rangle)\\
&\qquad \text{since} \quad ((0 < \square_{\mathbf{E}})[\text{y}], \langle \text{y} \mapsto 2 \rangle) \longrightarrow ((0 < \square_{\mathbf{E}})[2], \langle \text{y} \mapsto 2 \rangle)\\
&\qquad \text{since} \quad (\text{y}, \langle \text{y} \mapsto 2 \rangle) \longrightarrow (2, \langle \text{y} \mapsto 2 \rangle)\\
&= ((\text{if } \square_{\mathbf{B}} \text{ then y} := 0 - \text{y; while } 0 < \text{y do y} := 0 - \text{y od else skip fi})[0 < 2], \langle \text{y} \mapsto 2 \rangle)\\
&\longrightarrow ((\text{if } \square_{\mathbf{B}} \text{ then y} := 0 - \text{y; while } 0 < \text{y do y} := 0 - \text{y od else skip fi})[\text{true}], \langle \text{y} \mapsto 2 \rangle)\\
&\qquad \text{since} \quad (0 < 2, \langle \text{y} \mapsto 2 \rangle) \longrightarrow (\text{true}, \langle \text{y} \mapsto 2 \rangle)\\
&= (\text{if true then y} := 0 - \text{y; while } 0 < \text{y do y} := 0 - \text{y od else skip fi}, \langle \text{y} \mapsto 2 \rangle)\\
&\longrightarrow (\text{y} := 0 - \text{y; while } 0 < \text{y do y} := 0 - \text{y od}, \langle \text{y} \mapsto 2 \rangle)\\
&= ((\square_{\mathbf{C}}; \text{ while } 0 < \text{y do y} := 0 - \text{y od})[\text{y} := 0 - \text{y}], \langle \text{y} \mapsto 2 \rangle)\\
&= ((\square_{\mathbf{C}}; \text{ while } 0 < \text{y do y} := 0 - \text{y od})[(\text{y} := \square_{\mathbf{E}})[0 - \text{y}]], \langle \text{y} \mapsto 2 \rangle)\\
&= ((\square_{\mathbf{C}}; \text{ while } 0 < \text{y do y} := 0 - \text{y od})[(\text{y} := \square_{\mathbf{E}})[(0 - \square_{\mathbf{E}})[\text{y}]]], \langle \text{y} \mapsto 2 \rangle)\\
&\longrightarrow ((\square_{\mathbf{C}}; \text{ while } 0 < \text{y do y} := 0 - \text{y od})[(\text{y} := \square_{\mathbf{E}})[(0 - \square_{\mathbf{E}})[2]]], \langle \text{y} \mapsto 2 \rangle)\\
&\qquad \text{since} \quad ((\text{y} := \square_{\mathbf{E}})[(0 - \square_{\mathbf{E}})[\text{y}]], \langle \text{y} \mapsto 2 \rangle) \longrightarrow ((\text{y} := \square_{\mathbf{E}})[(0 - \square_{\mathbf{E}})[2]], \langle \text{y} \mapsto 2 \rangle)\\
&\qquad \text{since} \quad ((0 - \square_{\mathbf{E}})[\text{y}], \langle \text{y} \mapsto 2 \rangle) \longrightarrow ((0 - \square_{\mathbf{E}})[2], \langle \text{y} \mapsto 2 \rangle)\\
&\qquad \text{since} \quad (\text{y}, \langle \text{y} \mapsto 2 \rangle) \longrightarrow (2, \langle \text{y} \mapsto 2 \rangle)\\
&= ((\square_{\mathbf{C}}; \text{ while } 0 < \text{y do y} := 0 - \text{y od})[(\text{y} := \square_{\mathbf{E}})[0 - 2]], \langle \text{y} \mapsto 2 \rangle)\\
&\longrightarrow ((\square_{\mathbf{C}}; \text{ while } 0 < \text{y do y} := 0 - \text{y od})[(\text{y} := \square_{\mathbf{E}})[-2]], \langle \text{y} \mapsto 2 \rangle)\\
&\qquad \text{since} \quad ((\text{y} := \square_{\mathbf{E}})[0 - 2], \langle \text{y} \mapsto 2 \rangle) \longrightarrow ((\text{y} := \square_{\mathbf{E}})[-2], \langle \text{y} \mapsto 2 \rangle)\\
&\qquad \text{since} \quad (0 - 2, \langle \text{y} \mapsto 2 \rangle) \longrightarrow (-2, \langle \text{y} \mapsto 2 \rangle)\\
&= ((\square_{\mathbf{C}}; \text{ while } 0 < \text{y do y} := 0 - \text{y od})[\text{y} := -2], \langle \text{y} \mapsto 2 \rangle)\\
&\longrightarrow ((\square_{\mathbf{C}}; \text{ while } 0 < \text{y do y} := 0 - \text{y od})[\text{skip}], \langle \text{y} \mapsto -2 \rangle)\\
&\qquad \text{since} \quad (\text{y} := -2, \langle \text{y} \mapsto 2 \rangle) \longrightarrow (\text{skip}, \langle \text{y} \mapsto -2 \rangle)\\
&= (\text{skip; while } 0 < \text{y do y} := 0 - \text{y od}, \langle \text{y} \mapsto -2 \rangle)\\
&\longrightarrow (\text{while } 0 < \text{y do y} := 0 - \text{y od}, \langle \text{y} \mapsto -2 \rangle)\\
&\longrightarrow (\text{if } 0 < \text{y then y} := 0 - \text{y; while } 0 < \text{y do y} := 0 - \text{y od else skip fi}, \langle \text{y} \mapsto -2 \rangle)\\
&= ((\text{if } \square_{\mathbf{B}} \text{ then y} := 0 - \text{y; while } 0 < \text{y do y} := 0 - \text{y od else skip fi})[0 < \text{y}], \langle \text{y} \mapsto -2 \rangle)\\
&= ((\text{if } \square_{\mathbf{B}} \text{ then y} := 0 - \text{y; while } 0 < \text{y do y} := 0 - \text{y od else skip fi})[(0 < \square_{\mathbf{E}})[\text{y}]], \langle \text{y} \mapsto -2 \rangle)\\
&\longrightarrow ((\text{if } \square_{\mathbf{B}} \text{ then y} := 0 - \text{y; while } 0 < \text{y do y} := 0 - \text{y od else skip fi})[(0 < \square_{\mathbf{E}})[-2]], \langle \text{y} \mapsto -2 \rangle)\\
&\qquad \text{since} \quad ((0 < \square_{\mathbf{E}})[\text{y}], \langle \text{y} \mapsto -2 \rangle) \longrightarrow ((0 < \square_{\mathbf{E}})[-2], \langle \text{y} \mapsto -2 \rangle)\\
&\qquad \text{since} \quad (\text{y}, \langle \text{y} \mapsto -2 \rangle) \longrightarrow (-2, \langle \text{y} \mapsto -2 \rangle)\\
&= ((\text{if } \square_{\mathbf{B}} \text{ then y} := 0 - \text{y; while } 0 < \text{y do y} := 0 - \text{y od else skip fi})[0 < -2], \langle \text{y} \mapsto -2 \rangle)\\
&\longrightarrow ((\text{if } \square_{\mathbf{B}} \text{ then y} := 0 - \text{y; while } 0 < \text{y do y} := 0 - \text{y od else skip fi})[\text{false}], \langle \text{y} \mapsto -2 \rangle)\\
&\qquad \text{since} \quad (0 < -2, \langle \text{y} \mapsto -2 \rangle) \longrightarrow (\text{false}, \langle \text{y} \mapsto -2 \rangle)\\
&= (\text{if false then y} := 0 - \text{y; while } 0 < \text{y do y} := 0 - \text{y od else skip fi}, \langle \text{y} \mapsto -2 \rangle)\\
&\longrightarrow (\text{skip}, \langle \text{y} \mapsto -2 \rangle)
\end{aligned}
$$