# Definition and Formal Metatheory of ABS


**Abstract**

We define ABS and describe its metatheory.


## Contents

# 1 Introduction

We describe the syntax, semantics, and metatheory of the ABS language.

# 2 Syntax

## 2.1 Functional Syntax

In this section, we define the abstract syntax of ABS.

| | | | |
|---|---|---|---|
| $T$ | ::= | | ground type |
| | \| | $\mathsf{Bool}$ | |
| | \| | $\mathsf{Int}$ | |

| | | | |
|---|---|---|---|
| $F$ | ::= | | function definition |
| | \| | $\mathbf{def}\ T\, fc(T_1\, x_1, \ldots, T_n\, x_n) = e;$ | |

| | | | |
|---|---|---|---|
| $t$ | ::= | | ground term |
| | \| | $b$ | boolean |
| | \| | $z$ | integer |
| | \| | $fut$ | future |

| | | | |
|---|---|---|---|
| $e$ | ::= | | expression |
| | \| | $t$ | term |
| | \| | $x$ | variable |
| | \| | $fc(e_1, \ldots, e_n)$ | function call |
| | \| | $-e$ | |
| | \| | $!e$ | |
| | \| | $e_1 + e_2$ | |
| | \| | $e_1 * e_2$ | |
| | \| | $e_1 == e_2$ | |
| | \| | $e_1 < e_2$ | |
| | \| | $e[x_1 \mapsto y_1, \ldots, x_n \mapsto y_n]$ | M |

## 2.2 Object Syntax

| | | | |
|---|---|---|---|
| $rhs$ | ::= | | right-hand side in assignment |
| | \| | $e$ | expression |
| | \| | $o!m(e_1, \ldots, e_n)$ | method invocation |
| | \| | $f.\mathbf{get}$ | value of future |

| | | | |
|---|---|---|---|
| $stmt$ | ::= | | statement |
| | \| | $stmt_1; stmt_2$ | |
| | \| | $\mathbf{skip}$ | |
| | \| | $x := rhs$ | |
| | \| | $\mathbf{if}\ e\{stmt_1\}\ \mathbf{else}\ \{stmt_2\}$ | |

$|$ **while** $e\{stmt\}$
$|$ **return** $e$

$M$ ::= method definition
$|$ $T\ m(T_1\ x_1,\ \ldots,\ T_i\ x_i)\{T'_1\ y_1;\ \ldots;\ T'_j\ y_j; stmt\}$

$CL$ ::= class definition
$|$ **class** $C\{T_1\ x_1;\ \ldots;\ T_i\ x_i; M_1 \ldots M_j\}$

$P$ ::= program
$|$ $CL_1 \ldots CL_n\{T_1\ x_1;\ \ldots;\ T_i\ x_i; stmt\}$

# 3 Semantics

In this section, we define the semantics of ABS.

## 3.1 Typing syntax

$sig$ ::=
$|$ $T_1,\ \ldots,\ T_n \rightarrow T$

$ctxv$ ::=
$|$ $T$
$|$ $sig$
$|$ **Fut** $< T >$

## 3.2 Runtime syntax

$task$ ::= runtime task
$|$ **tsk** $(stmt, \sigma)$

$cn$ ::= configuration
$|$ **future** $(f, to)$
$|$ **object** $(C, \sigma, tasko, queue)$
$|$ **invoc** $(o, f, m, t_1 \,..\, t_n)$

## 3.3 Typing relation

$\boxed{\Gamma \vdash e : T}$ well-typed expression

$$\frac{}{\Gamma \vdash b : \mathsf{Bool}}\ \text{TYP\_BOOL}$$

$$\frac{}{\Gamma \vdash z : \mathsf{Int}}\ \text{TYP\_INT}$$

$$\frac{\Gamma(x) = T}{\Gamma \vdash x : T} \quad \text{TYP\_VAR}$$

$$\frac{\Gamma \vdash e : \mathsf{Int}}{\Gamma \vdash -e : \mathsf{Int}} \quad \text{TYP\_NEG}$$

$$\frac{\Gamma \vdash e : \mathsf{Bool}}{\Gamma \vdash !e : \mathsf{Bool}} \quad \text{TYP\_NOT}$$

$$\frac{\begin{array}{c}\Gamma \vdash e_1 : \mathsf{Int}\\ \Gamma \vdash e_2 : \mathsf{Int}\end{array}}{\Gamma \vdash e_1 + e_2 : \mathsf{Int}} \quad \text{TYP\_ADD}$$

$$\frac{\begin{array}{c}\Gamma \vdash e_1 : \mathsf{Int}\\ \Gamma \vdash e_2 : \mathsf{Int}\end{array}}{\Gamma \vdash e_1 * e_2 : \mathsf{Int}} \quad \text{TYP\_MUL}$$

$$\frac{\begin{array}{c}\Gamma \vdash e_1 : \mathsf{Int}\\ \Gamma \vdash e_2 : \mathsf{Int}\end{array}}{\Gamma \vdash e_1 == e_2 : \mathsf{Bool}} \quad \text{TYP\_EQ}$$

$$\frac{\begin{array}{c}\Gamma \vdash e_1 : \mathsf{Int}\\ \Gamma \vdash e_2 : \mathsf{Int}\end{array}}{\Gamma \vdash e_1 < e_2 : \mathsf{Bool}} \quad \text{TYP\_LT}$$

$$\frac{\begin{array}{l}\Gamma \vdash e_1 : T_1 \quad ... \quad \Gamma \vdash e_n : T_n\\ \Gamma(fc) = T_1, ..., T_n \to T\end{array}}{\Gamma \vdash fc(e_1, ..., e_n) : T} \quad \text{TYP\_FUNC\_EXPR}$$

$\boxed{\Gamma \vdash F}$ well-typed function declaration

$$\frac{\begin{array}{l}\Gamma(fc) = T_1, ..., T_n \to T\\ \Gamma[x_1 \mapsto T_1, ..., x_n \mapsto T_n] \vdash e : T\\ \textbf{distinct}\,(x_1, ..., x_n)\end{array}}{\Gamma \vdash \textbf{def}\ T\, fc(T_1\, x_1, ..., T_n\, x_n) = e;} \quad \text{TYP\_FUNC\_DECL}$$

## 3.4 Reduction relation

$\boxed{F_1 ... F_n, \sigma \vdash e \rightsquigarrow \sigma' \vdash e'}$ expression evaluation

$$\frac{\sigma(x) = t}{F_1 ... F_n, \sigma \vdash x \rightsquigarrow \sigma \vdash t} \quad \text{RED\_VAR}$$

$$\frac{}{F_1 ... F_n, \sigma \vdash -z \rightsquigarrow \sigma \vdash (-)z} \quad \text{RED\_NEG}$$

$$\frac{}{F_1 ... F_n, \sigma \vdash !b \rightsquigarrow \sigma \vdash (!)b} \quad \text{RED\_NOT}$$

$$\frac{}{F_1 ... F_n, \sigma \vdash z_1 + z_2 \rightsquigarrow \sigma \vdash z_1(+)z_2} \quad \text{RED\_ADD}$$

$$\frac{}{F_1 ... F_n, \sigma \vdash z_1 * z_2 \rightsquigarrow \sigma \vdash z_1(*)z_2} \quad \text{RED\_MUL}$$

$$\frac{}{F_1 ... F_n, \sigma \vdash z_1 == z_2 \rightsquigarrow \sigma \vdash z_1(==)z_2} \quad \text{RED\_EQ}$$

$$\frac{}{F_1 \ldots F_n, \sigma \vdash z_1 < z_2 \leadsto \sigma \vdash z_1 (<) z_2} \quad \text{RED\_LT}$$

$$\frac{F_1 \ldots F_n, \sigma \vdash e \leadsto \sigma' \vdash e'}{F_1 \ldots F_n, \sigma \vdash -e \leadsto \sigma' \vdash -e'} \quad \text{RED\_NEG'}$$

$$\frac{F_1 \ldots F_n, \sigma \vdash e \leadsto \sigma' \vdash e'}{F_1 \ldots F_n, \sigma \vdash !e \leadsto \sigma' \vdash !e'} \quad \text{RED\_NOT'}$$

$$\frac{F_1 \ldots F_n, \sigma \vdash e_1 \leadsto \sigma' \vdash e'}{F_1 \ldots F_n, \sigma \vdash e_1 + e_2 \leadsto \sigma' \vdash e' + e_2} \quad \text{RED\_ADD\_L}$$

$$\frac{F_1 \ldots F_n, \sigma \vdash e_2 \leadsto \sigma' \vdash e'}{F_1 \ldots F_n, \sigma \vdash e_1 + e_2 \leadsto \sigma' \vdash e_1 + e'} \quad \text{RED\_ADD\_R}$$

$$\frac{F_1 \ldots F_n, \sigma \vdash e_1 \leadsto \sigma' \vdash e'}{F_1 \ldots F_n, \sigma \vdash e_1 * e_2 \leadsto \sigma' \vdash e' + e_2} \quad \text{RED\_MUL\_L}$$

$$\frac{F_1 \ldots F_n, \sigma \vdash e_2 \leadsto \sigma' \vdash e'}{F_1 \ldots F_n, \sigma \vdash e_1 * e_2 \leadsto \sigma' \vdash e_1 + e'} \quad \text{RED\_MUL\_R}$$

$$\frac{F_1 \ldots F_n, \sigma \vdash e_1 \leadsto \sigma' \vdash e'}{F_1 \ldots F_n, \sigma \vdash e_1 == e_2 \leadsto \sigma' \vdash e' == e_2} \quad \text{RED\_EQ\_L}$$

$$\frac{F_1 \ldots F_n, \sigma \vdash e_2 \leadsto \sigma' \vdash e'}{F_1 \ldots F_n, \sigma \vdash e_1 == e_2 \leadsto \sigma' \vdash e_1 == e'} \quad \text{RED\_EQ\_R}$$

$$\frac{F_1 \ldots F_n, \sigma \vdash e_1 \leadsto \sigma' \vdash e'}{F_1 \ldots F_n, \sigma \vdash e_1 < e_2 \leadsto \sigma' \vdash e' < e_2} \quad \text{RED\_LT\_L}$$

$$\frac{F_1 \ldots F_n, \sigma \vdash e_2 \leadsto \sigma' \vdash e'}{F_1 \ldots F_n, \sigma \vdash e_1 < e_2 \leadsto \sigma' \vdash e_1 < e'} \quad \text{RED\_LT\_R}$$

$$\frac{F_1 \ldots F_n, \sigma \vdash e \leadsto \sigma' \vdash e'}{F_1 \ldots F_n, \sigma \vdash fc(e_1, \ldots, e_i, e, e'_1, \ldots, e'_j) \leadsto \sigma' \vdash fc(e_1, \ldots, e_i, e', e'_1, \ldots, e'_j)} \quad \text{RED\_FUN\_EXP}$$

$$\frac{\begin{array}{l}\textbf{well\_formed}\,(y_1, \ldots, y_n, e, \sigma)\\ \textbf{disjoint}\,((y_1, \ldots, y_n), (x_1, \ldots, x_n))\end{array}}{F_1 \ldots F_i\ \textbf{def}\ T\, fc(T_1\, x_1, \ldots, T_n\, x_n) = e;\ F'_1 \ldots F'_j, \sigma \vdash fc(t_1, \ldots, t_n) \leadsto \sigma[y_1 \mapsto t_1, \ldots, y_n \mapsto t_n] \vdash e[x_1 \mapsto y_1, \ldots, x_n \mapsto y_n]}$$

# 4 Metatheory

In this section, we define the metatheory of ABS.

**Definition.** A context $\Gamma'$ subsumes a context $\Gamma$, written $\Gamma \subseteq \Gamma'$, if (1) whenever $\Gamma(x) = T$ then $\Gamma'(x) = T$ and (2) whenever $\Gamma(fc) = sig$ then $\Gamma'(fc) = sig$.

**Definition.** A context $\Gamma$ is consistent with a substitutition $\sigma$, written $\Gamma \vdash \sigma$, if whenever $\sigma(x) = t$ and $\Gamma(x) = T$, then $\Gamma \vdash t : T$.

**Theorem 1** (Type preservation)**.** Assume $\Gamma \vdash F_1 \quad \ldots \quad \Gamma \vdash F_n$ and $\Gamma \vdash \sigma$. If $\Gamma \vdash e : T$ and $F_1 \ldots F_n, \sigma \vdash e \leadsto \sigma' \vdash e'$, then there is a $\Gamma'$ such that $\Gamma \subseteq \Gamma'$, $\Gamma' \vdash \sigma'$, and $\Gamma' \vdash e' : T$.