Oat v. 1 Language Specification

CIS341 – Steve Zdancewic

March 3, 2022

1 Grammar

The following grammar defines the Oat syntax. All binary operations are *left associative* with precedence levels indicated numerically. Higher precedence operators bind tighter than lower precedence ones.

prog	::=		prog
	|	$\mathit{decl}_1 \,..\, \mathit{decl}_i$	
decl	::=		global declarations
	|	*gdecl*	
	|	*fdecl*	
gdecl	::=		global variable declarations
	|	`global` *id* = *gexp*;	
arg	::=		arg
	|	*t id*	
args	::=		args
	|	$\mathit{arg}_1, \,..\,, \mathit{arg}_n$	
fdecl	::=		function declaration
	|	*retty id*(*args*) *block*	
block	::=		blocks
	|	$\{\mathit{stmt}_1 \,..\, \mathit{stmt}_n\}$	
t	::=		types
	|	`int`	
	|	`bool`	
	|	*ref*	

ref	::=		reference types
	|	`string`	
	|	*t*`[]`	

F	::=		function types
	$\mid$	$(t_0, .., t_n) \rightarrow \mathit{retty}$	
retty	::=		return types
	$\mid$	`void`	
	$\mid$	*t*	
bop	::=		(left associative) binary operations
	$\mid$	`*`	multiplication (precedence 100)
	$\mid$	+	addition (precedence 90)
	$\mid$	`-`	subtraction (precedence 90)
	$\mid$	`<<`	shift left (precedence 80)
	$\mid$	`>>`	shift right logical (precedence 80)
	$\mid$	`>>>`	shift right arithmetic (precedence 80)
	$\mid$	`<`	less-than (precedence 70)
	$\mid$	`<=`	less-than or equal (precedence 70)
	$\mid$	`>`	greater-than (precedence 70)
	$\mid$	`>=`	greater-than or equal (precedence 70)
	$\mid$	`==`	equal (precedence 60)
	$\mid$	`!=`	not equal (precedence 60)
	$\mid$	`&`	logical and (precedence 50)
	$\mid$	`|`	logical or (precedence 40)
	$\mid$	`[&]`	bit-wise and (precedence 30)
	$\mid$	`[|]`	bit-wise or (precedence 20)
uop	::=		unary operations
	$\mid$	`-`	
	$\mid$	`!`	
	$\mid$	`~`	
gexp	::=		global initializers
	$\mid$	*integer*	64-bit integer literals
	$\mid$	*string*	C-style strings
	$\mid$	*ref* `null`	
	$\mid$	`true`	
	$\mid$	`false`	
	$\mid$	`new` *t* `[]{`$\mathit{gexp}_1, .., \mathit{gexp}_n$`}`	
lhs	::=		lhs expressions
	$\mid$	*id*	
	$\mid$	exp_1`[`exp_2`]`	

exp	::=		expressions
	|	*id*	
	|	*integer*	64-bit integer literals
	|	*string*	C-style strings
	|	*ref* `null`	
	|	`true`	
	|	`false`	
	|	exp_1 `[`exp_2`]`	
	|	*id*$(exp_1,..,exp_n)$	
	|	`new` *t* `[]{`$exp_1,..,exp_n$`}`	Explicitly initialized array
	|	`new int [`exp_1`]`	Default-initialize int array
	|	`new bool [`exp_1`]`	Default-initialize bool array
	|	exp_1 *bop* exp_2	
	|	*uop exp*	
	|	(*exp*)	
vdecl	::=		local declarations
	|	`var` *id* `=` *exp*	
vdecls	::=		decl list
	|	$vdecl_1,..,vdecl_n$	
stmt	::=		statements
	|	*lhs* `=` *exp* `;`	
	|	*vdecl* `;`	
	|	`return` *exp* `;`	
	|	`return ;`	
	|	*id*$(exp_1,..,exp_n)$ `;`	
	|	*if_stmt*	
	|	`for`(*vdecls* `;` exp_{opt} `;` $stmt_{opt}$) *block*	
	|	`while`(*exp*) *block*	
if_stmt	::=		if statements
	|	`if`(*exp*) *block else_stmt*	
else_stmt	::=		else
	|	ϵ	
	|	`else` *block*	
	|	`else` *if_stmt*	

2 Typing Rules

$\boxed{\vdash bop_1, .., bop_i : F}$

$$\frac{}{\vdash \texttt{+}, \texttt{*}, \texttt{-}, \texttt{<<}, \texttt{>>}, \texttt{>>>}, \texttt{[\&]}, \texttt{[|]} : (\texttt{int}, \texttt{int}) \to \texttt{int}} \quad \textsc{typ\_intOps}$$

$$\frac{}{\vdash \texttt{==}, \texttt{!=}, \texttt{<}, \texttt{<=}, \texttt{>}, \texttt{>=} : (\texttt{int}, \texttt{int}) \to \texttt{bool}} \quad \textsc{typ\_cmpOps}$$

$$\frac{}{\vdash \texttt{\&}, \texttt{|} : (\texttt{bool}, \texttt{bool}) \to \texttt{bool}} \quad \textsc{typ\_boolOps}$$

$\boxed{\vdash uop : F}$

$$\frac{}{\vdash \texttt{!} : (\texttt{bool}) \to \texttt{bool}} \quad \textsc{typ\_lognot}$$

$$\frac{}{\vdash \texttt{\textasciitilde} : (\texttt{int}) \to \texttt{int}} \quad \textsc{typ\_bitneg}$$

$$\frac{}{\vdash \texttt{-} : (\texttt{int}) \to \texttt{int}} \quad \textsc{typ\_neg}$$

$\boxed{G ; L \vdash exp : t}$

$$\frac{x : t \in L}{G ; L \vdash x : t} \quad \textsc{typ\_local}$$

$$\frac{}{G ; L \vdash n : \texttt{int}} \quad \textsc{typ\_int}$$

$$\frac{}{G ; L \vdash s : \texttt{string}} \quad \textsc{typ\_string}$$

$$\frac{}{G ; L \vdash \texttt{string}\,\texttt{null} : \texttt{string}} \quad \textsc{typ\_nullStr}$$

$$\frac{}{G ; L \vdash t\,\texttt{[]}\ \texttt{null} : t\,\texttt{[]}} \quad \textsc{typ\_nullArr}$$

$$\frac{}{G ; L \vdash \texttt{true} : \texttt{bool}} \quad \textsc{typ\_true}$$

$$\frac{}{G ; L \vdash \texttt{false} : \texttt{bool}} \quad \textsc{typ\_false}$$

$$\frac{G ; L \vdash exp_1 : t\,\texttt{[]} \quad G ; L \vdash exp_2 : \texttt{int}}{G ; L \vdash exp_1\,\texttt{[}exp_2\texttt{]} : t} \quad \textsc{typ\_index}$$

$$\frac{f : (t_1, .., t_i) \to t \in G \quad G ; L \vdash exp_1 : t_1 \quad .. \quad G ; L \vdash exp_i : t_i}{G ; L \vdash f(exp_1, .., exp_i) : t} \quad \textsc{typ\_call}$$

$$\frac{G ; L \vdash exp_1 : t \quad .. \quad G ; L \vdash exp_i : t}{G ; L \vdash \texttt{new}\,t\,\texttt{[]}\texttt{\{}exp_1, .., exp_i\texttt{\}} \ : t\,\texttt{[]}} \quad \textsc{typ\_arrlit}$$

$$\frac{G ; L \vdash exp_1 : \texttt{int}}{G ; L \vdash \texttt{new}\,\texttt{int}\ \texttt{[}exp_1\texttt{]} : t\,\texttt{[]}} \quad \textsc{typ\_arrzeroint}$$

$$\frac{G ; L \vdash exp_1 : \texttt{int}}{G ; L \vdash \texttt{new}\,\texttt{bool}\ \texttt{[}exp_1\texttt{]} : t\,\texttt{[]}} \quad \textsc{typ\_arrzerobool}$$

$$\frac{\vdash bop : (t_1, t_2) \to t \quad G ; L \vdash exp_1 : t_1 \quad G ; L \vdash exp_2 : t_2}{G ; L \vdash exp_1\,bop\,exp_2 : t} \quad \textsc{typ\_bop}$$

$$\frac{\vdash uop : (t) \to t \quad G ; L \vdash exp : t}{G ; L \vdash uop\,exp : t} \quad \textsc{typ\_uop}$$

$\boxed{G\,;L_1 \vdash \mathit{vdecl} \Rightarrow L_2}$

$$\frac{G\,;L \vdash \mathit{exp} : t \quad x \notin L}{G\,;L \vdash \texttt{var}\ x\ \texttt{=}\ \mathit{exp} \Rightarrow L, x\!:\!t}\ \textsc{typ\_decl}$$

$\boxed{G\,;L_0 \vdash \mathit{vdecls} \Rightarrow L_i}$

$$\frac{G\,;L_0 \vdash \mathit{vdecl}_1 \Rightarrow L_1 \quad \ldots \quad G\,;L_{i-1} \vdash \mathit{vdecl}_i \Rightarrow L_i}{G\,;L_0 \vdash \mathit{vdecl}_1\texttt{,}\ ..\ \texttt{,}\mathit{vdecl}_i \Rightarrow L_i}\ \textsc{typ\_vdecls}$$

$\boxed{G\,;L_1\,;\mathit{retty} \vdash \mathit{stmt} \Rightarrow L_2}$

$$\frac{G\,;L_1 \vdash \mathit{vdecl} \Rightarrow L_2}{G\,;L_1\,;t \vdash \mathit{vdecl}\texttt{;} \Rightarrow L_2}\ \textsc{typ\_sdecl}$$

$$\frac{G\,;L \vdash \mathit{lhs} : t \quad G\,;L \vdash \mathit{exp}_2 : t}{G\,;L\,;t \vdash \mathit{lhs}\ \texttt{=}\ \mathit{exp}_2\texttt{;} \Rightarrow L}\ \textsc{typ\_assn}$$

$$\frac{f\!:\!(t_1\texttt{,}\ ..\ \texttt{,}t_i) \to \texttt{void} \in G \quad G\,;L \vdash \mathit{exp}_1 : t_1 \quad .. \quad G\,;L \vdash \mathit{exp}_i : t_i}{G\,;L\,;t \vdash f(\mathit{exp}_1\texttt{,}\ ..\ \texttt{,}\mathit{exp}_i)\texttt{;} \Rightarrow L}\ \textsc{typ\_scall}$$

$$\frac{G\,;L \vdash \mathit{exp} : \texttt{bool} \quad G\,;L\,;t \vdash \mathit{block}_1 \quad G\,;L\,;t \vdash \mathit{block}_2}{G\,;L\,;t \vdash \texttt{if}(\mathit{exp})\ \mathit{block}_1\ \texttt{else}\ \mathit{block}_2 \Rightarrow L}\ \textsc{typ\_if}$$

$$\frac{G\,;L \vdash \mathit{exp} : \texttt{bool} \quad G\,;L\,;t \vdash \mathit{block}}{G\,;L\,;t \vdash \texttt{while}(\mathit{exp})\ \mathit{block} \Rightarrow L}\ \textsc{typ\_while}$$

$$\frac{G\,;L_1 \vdash \mathit{vdecls} \Rightarrow L_2 \quad G\,;L_2 \vdash \mathit{exp} : \texttt{bool} \quad G\,;L_2\,;t \vdash \mathit{stmt} \Rightarrow L_3 \quad G\,;L_2\,;t \vdash \mathit{block}}{G\,;L_1\,;t \vdash \texttt{for}(\mathit{vdecls}\texttt{;}\ \mathit{exp}_{\mathit{opt}}\texttt{;}\ \mathit{stmt}_{\mathit{opt}})\ \mathit{block} \Rightarrow L_1}\ \textsc{typ\_for}$$

$$\frac{G\,;L \vdash \mathit{exp} : t}{G\,;L\,;t \vdash \texttt{return}\ \mathit{exp}\texttt{;} \Rightarrow L}\ \textsc{typ\_ret}\mathrm{T}$$

$$\frac{}{G\,;L\,;\texttt{void} \vdash \texttt{return ;} \Rightarrow L}\ \textsc{typ\_ret}\mathrm{Void}$$

$\boxed{G\,;L\,;t \vdash \mathit{block}}$

$$\frac{G\,;L_0\,;t \vdash \mathit{stmt}_1\,..\,\mathit{stmt}_i \Rightarrow L_i}{G\,;L_0\,;t \vdash \texttt{\{}\mathit{stmt}_1\,..\,\mathit{stmt}_i\texttt{\}}}\ \textsc{typ\_block}$$

$\boxed{G\,;L_0\,;t \vdash \mathit{stmt}_1\,..\,\mathit{stmt}_i \Rightarrow L_i}$

$$\frac{G\,;L_0\,;t \vdash \mathit{stmt}_1 \Rightarrow L_1 \quad \ldots \quad G\,;L_{i-1}\,;t \vdash \mathit{stmt}_i \Rightarrow L_i}{G\,;L_0\,;t \vdash \mathit{stmt}_1\,..\,\mathit{stmt}_i \Rightarrow L_i}\ \textsc{typ\_stmts}$$

$\boxed{G_0 \vdash \mathit{decl} \Rightarrow G_1}$

$$\frac{x \notin G \quad \cdot\,;\cdot \vdash \mathit{gexp} : t}{G \vdash \texttt{global}\ x = \mathit{gexp};\ \Rightarrow G, x : t} \quad \text{TYP\_VDECL}$$

$$\frac{f \notin G}{G \vdash t\, f(t_1\, x_1, .., t_i\, x_i)\ \mathit{block} \Rightarrow G, f : (t_1, .., t_i) \to t} \quad \text{TYP\_FDECL}$$

$\boxed{G_0 \vdash \mathit{decl}_1 .. \mathit{decl}_i \Rightarrow G_i}$

$$\frac{G_0 \vdash \mathit{decl}_1 \Rightarrow G_1 \quad \ldots \quad G_{i-1} \vdash \mathit{decl}_i \Rightarrow G_i}{G_0 \vdash \mathit{decl}_1 .. \mathit{decl}_i \Rightarrow G_i} \quad \text{TYP\_GLOBAL\_CTXT}$$

$\boxed{G \vdash \mathit{decl}}$

$$\frac{G\,; x_1 : t_1, .., x_i : t_i\,; t \vdash \mathit{block}}{G \vdash t\, f(t_1\, x_1, .., t_i\, x_i)\ \mathit{block}} \quad \text{TYP\_GFUN}$$

$$\frac{\cdot\,;\cdot \vdash \mathit{gexp} : t}{G \vdash \texttt{global}\ x = \mathit{gexp};} \quad \text{TYP\_GVAR}$$

$\boxed{\vdash \mathit{prog}}$

$$\frac{G_0 \vdash \mathit{decl}_1 .. \mathit{decl}_i \Rightarrow G \quad G \vdash \mathit{decl}_1 \quad .. \quad G \vdash \mathit{decl}_i}{\vdash \mathit{decl}_1 .. \mathit{decl}_i} \quad \text{TYP\_PROG}$$