CILA

Language specification

Jakub Mendyk

June 1, 2019

Grammar

integer	::=	*digit* | integer *digit*
keyword	::=	if | then | else | fi | while | do | od | div | mod | or | and | not | let
alfanum	::=	*letter* | alfanum *letter* | alfanum *digit*
ident	::=	alfanum (not in keyword)
program	::=	instruction | program instruction
instruction	::=	assignment
	|	**if** $\text{logic}_{\text{expr}}$ **then** program **fi**
	|	**if** $\text{logic}_{\text{expr}}$ **then** program **else** program **fi**
	|	**while** $\text{logic}_{\text{expr}}$ **do** program **od**
assignment	::=	**let ident** := $\text{arith}_{\text{expr}}$;
	|	**let ident** := [$\text{arith}_{\text{expr}}$ *]{ $\text{arith}_{\text{expr}}$, ... } ;
	|	**ident**[$\text{arith}_{\text{expr}}$]* := $\text{arith}_{\text{expr}}$;
	|	**fun ident (ident**, ... **) nuf** ;
$\text{logic}_{\text{expr}}$	::=	($\text{logic}_{\text{expr}}$ **or**)* $\text{logic}_{\text{summand}}$
$\text{logic}_{\text{summand}}$	::=	($\text{logic}_{\text{summand}}$ **and**)* $\text{logic}_{\text{multiplicand}}$
$\text{logic}_{\text{multiplicand}}$	::=	rel_{expr} | **not** $\text{logic}_{\text{multiplicand}}$
rel_{expr}	::=	$\text{arith}_{\text{expr}}$ rel_{op} $\text{arith}_{\text{expr}}$ | ($\text{logic}_{\text{expr}}$)
rel_{op}	::=	= | < | > | <= | >= | <>
$\text{arith}_{\text{expr}}$	::=	($\text{arith}_{\text{expr}}$ summ_{op})* $\text{arith}_{\text{summand}}$
$\text{arith}_{\text{summand}}$	::=	($\text{arith}_{\text{summand}}$ mult_{op})* $\text{arith}_{\text{multiplicand}}$
$\text{arith}_{\text{multiplicand}}$	::=	$\text{simple}_{\text{expr}}$ | $\text{simple}_{\text{expr}}$ ^ $\text{arith}_{\text{multiplicand}}$
$\text{simple}_{\text{expr}}$	::=	($\text{arith}_{\text{expr}}$) | **integer** | **ident** | **ident**[$\text{arith}_{\text{expr}}$] | **ident**($\text{arith}_{\text{expr}}$, ...)
summ_{op}	::=	+ | −
mult_{op}	::=	***** | **div** | **mod**