

CS 4240 Phase 1

David Zhang

Matheus Smythe Svolenski

February 23, 2014

The lexer/parser is run by compiling all the Java files and running the General.Runner class. The program takes in one argument for the tiger file, which must have a .tiger extension. The program outputs two files programname.tokens and programname.tokens.err, the first is a list of tokens. The second is any error output if it exists.

1 Lexical Rules

Pre-parse the input to remove comments with the following DFA. The resulting NOT-COMMENT tokens are concatenated together.

Everything under the token column represents either generated tokens or custom table actions on the character buffer used to produce the text associated with tokens.

start state	symbol	next state	token
START	$\Sigma - \{/, \}$	START	NOT-COMMENT
START	"	STRING	
START	/	SLASH	
STRING	$\Sigma - \{\\, \}$	STRING	
STRING	\	STRING-SLASH	
STRING	"	START	NOT-COMMENT
STRING-SLASH	Σ	STRING	
SLASH	$\Sigma - \{*\}$	START	NOT-COMMENT
SLASH	*	COMMENT	
COMMENT	$\Sigma - \{*\}$	COMMENT	
COMMENT	*	COMMENT-END	
COMMENT-END	$\Sigma - \{*, /\}$	COMMENT	
COMMENT-END	*	COMMENT-END	
COMMENT-END	/	START	COMMENT

The DFA for uncommented code.

Note, any time backtracking is mentioned, it essentially is the same as treating the current state as the start state and doing the corresponding transitions or token generations. This is included to simplify the table by removing duplication of the start state transitions.

Note, all ids are later matched character by character with keywords to determine if they are keywords.

Note, drop character for the error state means that the last read character is ignored, and the state remains unchanged.

start state	symbol	next state	token
START	+	START	PLUS
START	-	START	MIN

START	*	START	MULT
START	/	START	DIV
START	=	START	EQ
START	(START	LPAREN
START)	START	RPAREN
START	,	START	COMMA
START	&	START	AND
START		START	OR
START	[START	LSQUARE
START]	START	RSQUARE
START	;	START	SEMI
START	<	LANGLE	
START	>	RANGLE	
START	:	COLON	
START	0-9	INT-LIT	
START	"	STRING-LIT	
START	a-zA-Z	ID	
START	whitespace	START	ignore
START	others	ERROR	drop character
LANGLE	$\Sigma - \{=, \}$	START	LESS , backtrack
LANGLE	>	START	NOTEQ
LANGLE	=	START	LESSEQ
RANGLE	$\Sigma - \{=\}$	START	GREATER, backtrack
RANGLE	=	START	GREATEREQ
COLON	=	START	ASSIGN
COLON	$\Sigma - \{=\}$	START	COLON, backtrack
INT-LIT	0-9	INT-LIT	
INT-LIT	$\Sigma - 0 - 9$	START	INT-LIT, backtrack
ID	a-zA-Z0-9_	ID	
ID	$\Sigma - a - zA - Z0 - 9_$	START	ID, backtrack
STRING-LIT	$\Sigma - \backslash$	STRING-LIT	
STRING-LIT	"	START	STRING-LIT
STRING-LIT	\	STRING-LIT-SLASH	
STRING-LIT-SLASH	n	STRING-LIT	
STRING-LIT-SLASH	t	STRING-LIT	
STRING-LIT-SLASH	"	STRING-LIT	
STRING-LIT-SLASH	\	STRING-LIT	
STRING-LIT-SLASH	^	STRING-LIT-CTL	
STRING-LIT-SLASH	0-9	STRING-LIT-CODE-1	
STRING-LIT-SLASH	whitespace	STRING-LIT-SPACE	ignore 2 characters
STRING-LIT-SLASH	others	ERROR	drop character
STRING-LIT-CTL	@A-Z[\]^_.	STRING-LIT	
STRING-LIT-CTL	others	ERROR	drop character
STRING-LIT-CODE-1	0-9	STRING-LIT-CODE-2	
STRING-LIT-CODE-1	others	ERROR	drop character
STRING-LIT-CODE-2	0-9	STRING-LIT	
STRING-LIT-CODE-2	others	ERROR	drop character
STRING-LIT-SPACE	whitespace	STRING-LIT-SPACE	ignore
STRING-LIT-SPACE	\	STRING-LIT	ignore
STRING-LIT-SPACE	others	ERROR	drop character

2 Grammar Rules

Given the raw grammar for the Tiger Language, provided in the Tiger Language Reference Manual, and shown below, we have generated a new grammar, by modifying it in order to ensure that it is not ambiguous, by enforcing operator precedences and left associativity, and to ensure that the grammar supports LL(1) parsing, by removing any left recursion and performing left factoring.

symbol	rule
$\langle \text{tiger-program} \rangle$	let $\langle \text{declaration-segment} \rangle$ in $\langle \text{stat-seq} \rangle$ end
$\langle \text{declaration-segment} \rangle$	$\langle \text{type-declaration-list} \rangle \langle \text{var-declaration-list} \rangle \langle \text{funct-declaration-list} \rangle$
$\langle \text{type-declaration-list} \rangle$	$\langle \text{type-declaration} \rangle \langle \text{type-declaration-list} \rangle$
$\langle \text{type-declaration-list} \rangle$	NULL
$\langle \text{var-declaration-list} \rangle$	$\langle \text{var-declaration} \rangle \langle \text{var-declaration-list} \rangle$
$\langle \text{var-declaration-list} \rangle$	NULL
$\langle \text{funct-declaration-list} \rangle$	$\langle \text{funct-declaration} \rangle \langle \text{funct-declaration-list} \rangle$
$\langle \text{funct-declaration-list} \rangle$	NULL
$\langle \text{type-declaration} \rangle$	type id = $\langle \text{type} \rangle$;
$\langle \text{type} \rangle$	$\langle \text{type-id} \rangle$
$\langle \text{type} \rangle$	array [INTLIT] $\langle \text{type-dim} \rangle$ of $\langle \text{type-id} \rangle$
$\langle \text{type-dim} \rangle$	[INTLIT] $\langle \text{type-dim} \rangle$
$\langle \text{type-dim} \rangle$	NULL
$\langle \text{type-id} \rangle$	int
$\langle \text{type-id} \rangle$	string
$\langle \text{type-id} \rangle$	id
$\langle \text{var-declaration} \rangle$	var $\langle \text{id-list} \rangle$: $\langle \text{type-id} \rangle$ $\langle \text{optional-init} \rangle$;
$\langle \text{id-list} \rangle$	id
$\langle \text{id-list} \rangle$	id , $\langle \text{id-list} \rangle$
$\langle \text{optional-init} \rangle$	NULL
$\langle \text{optional-init} \rangle$:= $\langle \text{const} \rangle$
$\langle \text{funct-declaration} \rangle$	function id ($\langle \text{param-list} \rangle$) $\langle \text{ret-type} \rangle$ begin $\langle \text{stat-seq} \rangle$ end ;
$\langle \text{param-list} \rangle$	NULL
$\langle \text{param-list} \rangle$	$\langle \text{param} \rangle \langle \text{param-list-tail} \rangle$
$\langle \text{param-list-tail} \rangle$	NULL
$\langle \text{param-list-tail} \rangle$, $\langle \text{param} \rangle \langle \text{param-list-tail} \rangle$
$\langle \text{ret-type} \rangle$	NULL
$\langle \text{ret-type} \rangle$: $\langle \text{type-id} \rangle$
$\langle \text{param} \rangle$	id : $\langle \text{type-id} \rangle$
$\langle \text{stat-seq} \rangle$	$\langle \text{stat} \rangle \langle \text{stat-seq} \rangle$
$\langle \text{stat-seq} \rangle$	$\langle \text{stat} \rangle$
$\langle \text{stat} \rangle$	$\langle \text{lvalue} \rangle$:= $\langle \text{expr} \rangle$;
$\langle \text{stat} \rangle$	if $\langle \text{expr} \rangle$ then $\langle \text{stat-seq} \rangle$ endif ;
$\langle \text{stat} \rangle$	if $\langle \text{expr} \rangle$ then $\langle \text{stat-seq} \rangle$ else $\langle \text{stat-seq} \rangle$ endif ;
$\langle \text{stat} \rangle$	while $\langle \text{expr} \rangle$ do $\langle \text{stat-seq} \rangle$ enddo ;
$\langle \text{stat} \rangle$	for id := $\langle \text{expr} \rangle$ to $\langle \text{expr} \rangle$ do $\langle \text{stat-seq} \rangle$ enddo ;
$\langle \text{stat} \rangle$	$\langle \text{opt-prefix} \rangle$ id ($\langle \text{expr-list} \rangle$) ;
$\langle \text{stat} \rangle$	break ;
$\langle \text{stat} \rangle$	return $\langle \text{expr} \rangle$;
$\langle \text{expr} \rangle$	$\langle \text{expr} \rangle \langle \text{binary-operator} \rangle \langle \text{expr} \rangle$
$\langle \text{expr} \rangle$	$\langle \text{const} \rangle$
$\langle \text{expr} \rangle$	$\langle \text{lvalue} \rangle$
$\langle \text{expr} \rangle$	- $\langle \text{expr} \rangle$
$\langle \text{expr} \rangle$	($\langle \text{expr} \rangle$)
$\langle \text{binary-operator} \rangle$	*

$\langle \text{binary-operator} \rangle$	/
$\langle \text{binary-operator} \rangle$	+
$\langle \text{binary-operator} \rangle$	-
$\langle \text{binary-operator} \rangle$	=
$\langle \text{binary-operator} \rangle$	<
$\langle \text{binary-operator} \rangle$	>
$\langle \text{binary-operator} \rangle$	<=
$\langle \text{binary-operator} \rangle$	>=
$\langle \text{binary-operator} \rangle$	<>
$\langle \text{binary-operator} \rangle$	&
$\langle \text{binary-operator} \rangle$	
$\langle \text{binary-operator} \rangle$:=
$\langle \text{opt-prefix} \rangle$	$\langle \text{lvalue} \rangle :=$
$\langle \text{opt-prefix} \rangle$	NULL
$\langle \text{const} \rangle$	INTLIT
$\langle \text{const} \rangle$	STRLIT
$\langle \text{const} \rangle$	nil
$\langle \text{expr-list} \rangle$	$\langle \text{expr} \rangle \langle \text{expr-list-tail} \rangle$
$\langle \text{expr-list} \rangle$	NULL
$\langle \text{expr-list-tail} \rangle$, $\langle \text{expr} \rangle \langle \text{expr-list-tail} \rangle$
$\langle \text{expr-list-tail} \rangle$	NULL
$\langle \text{lvalue} \rangle$	id $\langle \text{lvalue-tail} \rangle$
$\langle \text{lvalue-tail} \rangle$	[$\langle \text{expr} \rangle$] $\langle \text{lvalue-tail} \rangle$
$\langle \text{lvalue-tail} \rangle$	NULL

After performing the grammar's modifications, we came to the following grammar.

symbol	rule
$\langle \text{tiger-program} \rangle$	let $\langle \text{declaration-segment} \rangle$ in $\langle \text{stat-seq} \rangle$ end
$\langle \text{declaration-segment} \rangle$	$\langle \text{type-declaration-list} \rangle \langle \text{var-declaration-list} \rangle \langle \text{funct-declaration-list} \rangle$
$\langle \text{type-declaration-list} \rangle$	$\langle \text{type-declaration} \rangle \langle \text{type-declaration-list} \rangle$
$\langle \text{type-declaration-list} \rangle$	NULL
$\langle \text{var-declaration-list} \rangle$	$\langle \text{var-declaration} \rangle \langle \text{var-declaration-list} \rangle$
$\langle \text{var-declaration-list} \rangle$	NULL
$\langle \text{funct-declaration-list} \rangle$	$\langle \text{funct-declaration} \rangle \langle \text{funct-declaration-list} \rangle$
$\langle \text{funct-declaration-list} \rangle$	NULL
$\langle \text{type-declaration} \rangle$	type id = $\langle \text{type} \rangle$;
$\langle \text{var-declaration} \rangle$	var $\langle \text{id-list} \rangle$: $\langle \text{type-id} \rangle \langle \text{optional-init} \rangle$;
$\langle \text{funct-declaration} \rangle$	function id ($\langle \text{param-list} \rangle$) $\langle \text{ret-type} \rangle$ begin $\langle \text{stat-seq} \rangle$ end ;
$\langle \text{type} \rangle$	$\langle \text{type-id} \rangle$
$\langle \text{type} \rangle$	array [INTLIT] $\langle \text{type-dim} \rangle$ of $\langle \text{type-id} \rangle$
$\langle \text{type-dim} \rangle$	[INTLIT] $\langle \text{type-dim} \rangle$
$\langle \text{type-dim} \rangle$	NULL
$\langle \text{type-id} \rangle$	id
$\langle \text{id-list} \rangle$	id $\langle \text{id-list-tail} \rangle$
$\langle \text{id-list-tail} \rangle$, id $\langle \text{id-list-tail} \rangle$
$\langle \text{id-list-tail} \rangle$	NULL
$\langle \text{optional-init} \rangle$:= $\langle \text{const} \rangle$
$\langle \text{optional-init} \rangle$	NULL
$\langle \text{param-list} \rangle$	$\langle \text{param} \rangle \langle \text{param-list-tail} \rangle$
$\langle \text{param-list} \rangle$	NULL
$\langle \text{param-list-tail} \rangle$, $\langle \text{param} \rangle \langle \text{param-list-tail} \rangle$
$\langle \text{param-list-tail} \rangle$	NULL

<ret-type>	: <type-id>
<ret-type>	NULL
<param>	id : <type-id>
<stat-seq>	<stat> <stat-seq-tail>
<stat-seq-tail>	<stat> <stat-seq-tail>
<stat-seq-tail>	NULL
<stat>	if <expr> then <stat-seq> <stat-if-tail>
<stat>	while <expr> do <stat-seq> enddo ;
<stat>	for id := <expr> to <expr> do <stat-seq> enddo ;
<stat>	break ;
<stat>	return <expr> ;
<stat>	id <stat-func-or-assign>
<stat-func-or-assign>	(<expr-list>) ;
<stat-func-or-assign>	<lvalue-tail> := <stat-assign> ;
<stat-if-tail>	else <stat-seq> endif ;
<stat-if-tail>	endif ;
<stat-assign>	- <unaryminus> <stat-assign-tail>
<stat-assign>	(<expr>) <stat-assign-tail>
<stat-assign>	<const> <stat-assign-tail>
<stat-assign>	id <stat-assign-id>
<stat-assign-id>	(<expr-list>)
<stat-assign-id>	<lvalue-tail> <stat-assign-tail>
<stat-assign-tail>	<expr-tail>
<stat-assign-tail>	<orexpr-tail>
<stat-assign-tail>	<andexpr-tail>
<stat-assign-tail>	<compare-tail>
<stat-assign-tail>	<term-tail>
<expr>	<orexpr> <expr-tail>
<expr-tail>	<orop> <orexpr> <expr-tail>
<expr-tail>	NULL
<orexpr>	<andexpr> <orexpr-tail>
<orexpr-tail>	<andop> <andexpr> <orexpr-tail>
<orexpr-tail>	NULL
<andexpr>	<compare> <andexpr-tail>
<andexpr-tail>	<compop> <compare> <andexpr-tail>
<andexpr-tail>	NULL
<compare>	<term> <compare-tail>
<compare-tail>	<addop> <term> <compare-tail>
<compare-tail>	NULL
<term>	<factor> <term-tail>
<term-tail>	<mulop> <factor> <term-tail>
<term-tail>	NULL
<factor>	<unaryminus>
<factor>	- <unaryminus>
<unaryminus>	(<expr>)
<unaryminus>	<const>
<unaryminus>	<lvalue>
<const>	INTLIT
<const>	STRLIT
<const>	nil
<orop>	
<andop>	&
<compop>	=

$\langle \text{compop} \rangle$	$\langle \rangle$
$\langle \text{compop} \rangle$	$>$
$\langle \text{compop} \rangle$	$<$
$\langle \text{compop} \rangle$	$>=$
$\langle \text{compop} \rangle$	$<=$
$\langle \text{addop} \rangle$	$+$
$\langle \text{addop} \rangle$	$-$
$\langle \text{mulop} \rangle$	$*$
$\langle \text{mulop} \rangle$	$/$
$\langle \text{expr-list} \rangle$	$\langle \text{expr} \rangle \langle \text{expr-list-tail} \rangle$
$\langle \text{expr-list} \rangle$	NULL
$\langle \text{expr-list-tail} \rangle$	$, \langle \text{expr} \rangle \langle \text{expr-list-tail} \rangle$
$\langle \text{expr-list-tail} \rangle$	NULL
$\langle \text{lvalue} \rangle$	$\text{id} \langle \text{lvalue-tail} \rangle$
$\langle \text{lvalue-tail} \rangle$	$[\langle \text{expr} \rangle] \langle \text{lvalue-tail} \rangle$
$\langle \text{lvalue-tail} \rangle$	NULL

After ensuring that the new grammar meets all the requirements, the first and follow sets were generated for every non-terminal symbol of the grammar.

non-terminal	first set
$\langle \text{lvalue-tail} \rangle$	$[, \epsilon$
$\langle \text{lvalue} \rangle$	id
$\langle \text{expr-list-tail} \rangle$	$,, \epsilon$
$\langle \text{expr-list} \rangle$	$(, \text{nil}, \text{STRLIT}, \text{INTLIT}, \text{id}, -, \epsilon$
$\langle \text{mulop} \rangle$	$*, /$
$\langle \text{addop} \rangle$	$+, -$
$\langle \text{compop} \rangle$	$=, <, >, <=, >=, \langle \rangle$
$\langle \text{andop} \rangle$	$\&$
$\langle \text{orop} \rangle$	$ $
$\langle \text{const} \rangle$	$\text{nil}, \text{STRLIT}, \text{INTLIT}$
$\langle \text{unaryminus} \rangle$	$(, \text{nil}, \text{STRLIT}, \text{INTLIT}, \text{id}$
$\langle \text{factor} \rangle$	$(, \text{nil}, \text{STRLIT}, \text{INTLIT}, \text{id}, -$
$\langle \text{term-tail} \rangle$	$*, /, \epsilon$
$\langle \text{term} \rangle$	$(, \text{nil}, \text{STRLIT}, \text{INTLIT}, \text{id}, -$
$\langle \text{compare-tail} \rangle$	$+, -, \epsilon$
$\langle \text{compare} \rangle$	$(, \text{nil}, \text{STRLIT}, \text{INTLIT}, \text{id}, -$
$\langle \text{andexpr-tail} \rangle$	$=, <, >, <=, >=, \langle \rangle, \epsilon$
$\langle \text{andexpr} \rangle$	$(, \text{nil}, \text{STRLIT}, \text{INTLIT}, \text{id}, -$
$\langle \text{orexpr} \rangle$	$(, \text{nil}, \text{STRLIT}, \text{INTLIT}, \text{id}, -$
$\langle \text{orexpr-tail} \rangle$	$\&, \epsilon$
$\langle \text{expr} \rangle$	$(, \text{nil}, \text{STRLIT}, \text{INTLIT}, \text{id}, -$
$\langle \text{expr-tail} \rangle$	$, \epsilon$
$\langle \text{stat-assign} \rangle$	$\text{id}, -, (, \text{nil}, \text{STRLIT}, \text{INTLIT}$
$\langle \text{stat-assign-id} \rangle$	$[, (, *, /, +, -, =, <, >, <=, >=, \langle \rangle, \&, , \epsilon$
$\langle \text{stat-assign-tail} \rangle$	$*, /, +, -, =, <, >, <=, >=, \langle \rangle, \&, , \epsilon$
$\langle \text{stat-if-tail} \rangle$	$\text{else}, \text{endif}$
$\langle \text{stat-func-or-assign} \rangle$	$(, :=, [$
$\langle \text{stat} \rangle$	$\text{if}, \text{while}, \text{for}, \text{break}, \text{return}, \text{id}$
$\langle \text{stat-seq} \rangle$	$\text{if}, \text{while}, \text{for}, \text{break}, \text{return}, \text{id}$
$\langle \text{stat-seq-tail} \rangle$	$\text{if}, \text{while}, \text{for}, \text{break}, \text{return}, \text{id}, \epsilon$
$\langle \text{param} \rangle$	id
$\langle \text{ret-type} \rangle$	$:, \epsilon$

$\langle \text{param-list-tail} \rangle$,, ϵ
$\langle \text{param-list} \rangle$	id, ϵ
$\langle \text{optional-init} \rangle$:=, ϵ
$\langle \text{id-list-tail} \rangle$,, ϵ
$\langle \text{id-list} \rangle$	id
$\langle \text{type-id} \rangle$	id
$\langle \text{type-dim} \rangle$	[, ϵ
$\langle \text{type} \rangle$	array, id
$\langle \text{funct-declaration} \rangle$	function
$\langle \text{var-declaration} \rangle$	var
$\langle \text{type-declaration} \rangle$	type
$\langle \text{funct-declaration-list} \rangle$	function, ϵ
$\langle \text{var-declaration-list} \rangle$	var, ϵ
$\langle \text{type-declaration-list} \rangle$	type, ϵ
$\langle \text{declaration-segment} \rangle$	function, var, type, ϵ
$\langle \text{tiger-program} \rangle$	let

non-terminal	follow set
$\langle \text{lvalue-tail} \rangle$:=, *, /, +, -, =, <, >, <=, >=, <>, &,
$\langle \text{lvalue-tail} \rangle$,), ,, ,], then, do, to, ;
$\langle \text{expr-list-tail} \rangle$)
$\langle \text{expr-list} \rangle$)
$\langle \text{expr-tail} \rangle$), ,, ,], then, do, to, ;
$\langle \text{orexpr-tail} \rangle$,), ,, ,], then, do, to, ;
$\langle \text{andexpr-tail} \rangle$	&, ,), ,, ,], then, do, to, ;
$\langle \text{compare-tail} \rangle$	&, ,), ,, ,], then, do, to, ;, =, <, >, <=, >=, <>
$\langle \text{term-tail} \rangle$	&, ,), ,, ,], then, do, to, ;, =, <, >, <=, >=, <>, +, -
$\langle \text{term-tail} \rangle$	
$\langle \text{stat-assign-tail} \rangle$;
$\langle \text{stat-assign-id} \rangle$;
$\langle \text{stat-seq-tail} \rangle$	endif, end, enddo, else
$\langle \text{ret-type} \rangle$	begin
$\langle \text{param-list-tail} \rangle$)
$\langle \text{param-list} \rangle$)
$\langle \text{optional-init} \rangle$;
$\langle \text{id-list-tail} \rangle$:
$\langle \text{type-dim} \rangle$	of
$\langle \text{funct-declaration-list} \rangle$	in
$\langle \text{var-declaration-list} \rangle$	function, in
$\langle \text{type-declaration-list} \rangle$	var, function, in
$\langle \text{declaration-segment} \rangle$	in

At last, the LL(1) parser table for Tiger was generated, as shown below.

Note that if there is no corresponding rule, then that means a parser error is generated. Also note that errors are handled by dropping tokens until a valid token is found.

symbol	next token	rule
$\langle \text{addop} \rangle$	+	+
$\langle \text{addop} \rangle$	-	-
$\langle \text{andexpr} \rangle$	($\langle \text{compare} \rangle \langle \text{andexpr-tail} \rangle$
$\langle \text{andexpr} \rangle$	nil	$\langle \text{compare} \rangle \langle \text{andexpr-tail} \rangle$
$\langle \text{andexpr} \rangle$	STRLIT	$\langle \text{compare} \rangle \langle \text{andexpr-tail} \rangle$

$\langle \text{andexpr} \rangle$	INTLIT	$\langle \text{compare} \rangle \langle \text{andexpr-tail} \rangle$
$\langle \text{andexpr} \rangle$	id	$\langle \text{compare} \rangle \langle \text{andexpr-tail} \rangle$
$\langle \text{andexpr} \rangle$	-	$\langle \text{compare} \rangle \langle \text{andexpr-tail} \rangle$
$\langle \text{andexpr-tail} \rangle$	=	$\langle \text{compop} \rangle \langle \text{compare} \rangle \langle \text{andexpr-tail} \rangle$
$\langle \text{andexpr-tail} \rangle$	<	$\langle \text{compop} \rangle \langle \text{compare} \rangle \langle \text{andexpr-tail} \rangle$
$\langle \text{andexpr-tail} \rangle$	>	$\langle \text{compop} \rangle \langle \text{compare} \rangle \langle \text{andexpr-tail} \rangle$
$\langle \text{andexpr-tail} \rangle$	>=	$\langle \text{compop} \rangle \langle \text{compare} \rangle \langle \text{andexpr-tail} \rangle$
$\langle \text{andexpr-tail} \rangle$	<=	$\langle \text{compop} \rangle \langle \text{compare} \rangle \langle \text{andexpr-tail} \rangle$
$\langle \text{andexpr-tail} \rangle$	<>	$\langle \text{compop} \rangle \langle \text{compare} \rangle \langle \text{andexpr-tail} \rangle$
$\langle \text{andexpr-tail} \rangle$	&	ϵ
$\langle \text{andexpr-tail} \rangle$		ϵ
$\langle \text{andexpr-tail} \rangle$)	ϵ
$\langle \text{andexpr-tail} \rangle$,	ϵ
$\langle \text{andexpr-tail} \rangle$]	ϵ
$\langle \text{andexpr-tail} \rangle$	then	ϵ
$\langle \text{andexpr-tail} \rangle$	do	ϵ
$\langle \text{andexpr-tail} \rangle$	to	ϵ
$\langle \text{andexpr-tail} \rangle$;	ϵ
$\langle \text{andop} \rangle$	&	&
$\langle \text{compare} \rangle$	($\langle \text{term} \rangle \langle \text{compare-tail} \rangle$
$\langle \text{compare} \rangle$	nil	$\langle \text{term} \rangle \langle \text{compare-tail} \rangle$
$\langle \text{compare} \rangle$	STRLIT	$\langle \text{term} \rangle \langle \text{compare-tail} \rangle$
$\langle \text{compare} \rangle$	INTLIT	$\langle \text{term} \rangle \langle \text{compare-tail} \rangle$
$\langle \text{compare} \rangle$	id	$\langle \text{term} \rangle \langle \text{compare-tail} \rangle$
$\langle \text{compare} \rangle$	-	$\langle \text{term} \rangle \langle \text{compare-tail} \rangle$
$\langle \text{compare-tail} \rangle$	+	$\langle \text{addop} \rangle \langle \text{term} \rangle \langle \text{compare-tail} \rangle$
$\langle \text{compare-tail} \rangle$	-	$\langle \text{addop} \rangle \langle \text{term} \rangle \langle \text{compare-tail} \rangle$
$\langle \text{compare-tail} \rangle$	&	ϵ
$\langle \text{compare-tail} \rangle$		ϵ
$\langle \text{compare-tail} \rangle$)	ϵ
$\langle \text{compare-tail} \rangle$,	ϵ
$\langle \text{compare-tail} \rangle$]	ϵ
$\langle \text{compare-tail} \rangle$	then	ϵ
$\langle \text{compare-tail} \rangle$	do	ϵ
$\langle \text{compare-tail} \rangle$	to	ϵ
$\langle \text{compare-tail} \rangle$;	ϵ
$\langle \text{compare-tail} \rangle$	=	ϵ
$\langle \text{compare-tail} \rangle$	<	ϵ
$\langle \text{compare-tail} \rangle$	>	ϵ
$\langle \text{compare-tail} \rangle$	<=	ϵ
$\langle \text{compare-tail} \rangle$	>=	ϵ
$\langle \text{compare-tail} \rangle$	<>	ϵ
$\langle \text{compop} \rangle$	=	=
$\langle \text{compop} \rangle$	<	<
$\langle \text{compop} \rangle$	>	>
$\langle \text{compop} \rangle$	<=	<=
$\langle \text{compop} \rangle$	>=	>=
$\langle \text{compop} \rangle$	<>	<>
$\langle \text{const} \rangle$	nil	nil
$\langle \text{const} \rangle$	STRLIT	STRLIT
$\langle \text{const} \rangle$	INTLIT	INTLIT
$\langle \text{declaration-segment} \rangle$	function	$\langle \text{type-declaration-list} \rangle \langle \text{var-declaration-list} \rangle \langle \text{funct-declaration-list} \rangle$
$\langle \text{declaration-segment} \rangle$	var	$\langle \text{type-declaration-list} \rangle \langle \text{var-declaration-list} \rangle \langle \text{funct-declaration-list} \rangle$

$\langle \text{declaration-segment} \rangle$	type	$\langle \text{type-declaration-list} \rangle \langle \text{var-declaration-list} \rangle \langle \text{funct-declaration-list} \rangle$
$\langle \text{declaration-segment} \rangle$	in	ϵ
$\langle \text{expr-list} \rangle$)	ϵ
$\langle \text{expr-list} \rangle$	($\langle \text{expr} \rangle \langle \text{expr-list-tail} \rangle$
$\langle \text{expr-list} \rangle$	nil	$\langle \text{expr} \rangle \langle \text{expr-list-tail} \rangle$
$\langle \text{expr-list} \rangle$	STRLIT	$\langle \text{expr} \rangle \langle \text{expr-list-tail} \rangle$
$\langle \text{expr-list} \rangle$	INTLIT	$\langle \text{expr} \rangle \langle \text{expr-list-tail} \rangle$
$\langle \text{expr-list} \rangle$	id	$\langle \text{expr} \rangle \langle \text{expr-list-tail} \rangle$
$\langle \text{expr-list} \rangle$	-	$\langle \text{expr} \rangle \langle \text{expr-list-tail} \rangle$
$\langle \text{expr-list-tail} \rangle$)	ϵ
$\langle \text{expr-list-tail} \rangle$,	$\langle \text{expr} \rangle \langle \text{expr-list-tail} \rangle$
$\langle \text{expr} \rangle$	($\langle \text{orexpr} \rangle \langle \text{expr-tail} \rangle$
$\langle \text{expr} \rangle$	nil	$\langle \text{orexpr} \rangle \langle \text{expr-tail} \rangle$
$\langle \text{expr} \rangle$	STRLIT	$\langle \text{orexpr} \rangle \langle \text{expr-tail} \rangle$
$\langle \text{expr} \rangle$	INTLIT	$\langle \text{orexpr} \rangle \langle \text{expr-tail} \rangle$
$\langle \text{expr} \rangle$	id	$\langle \text{orexpr} \rangle \langle \text{expr-tail} \rangle$
$\langle \text{expr} \rangle$	-	$\langle \text{orexpr} \rangle \langle \text{expr-tail} \rangle$
$\langle \text{expr-tail} \rangle$)	ϵ
$\langle \text{expr-tail} \rangle$,	ϵ
$\langle \text{expr-tail} \rangle$]	ϵ
$\langle \text{expr-tail} \rangle$	then	ϵ
$\langle \text{expr-tail} \rangle$	do	ϵ
$\langle \text{expr-tail} \rangle$	to	ϵ
$\langle \text{expr-tail} \rangle$;	ϵ
$\langle \text{factor} \rangle$	($\langle \text{unaryminus} \rangle$
$\langle \text{factor} \rangle$	nil	$\langle \text{unaryminus} \rangle$
$\langle \text{factor} \rangle$	STRLIT	$\langle \text{unaryminus} \rangle$
$\langle \text{factor} \rangle$	INTLIT	$\langle \text{unaryminus} \rangle$
$\langle \text{factor} \rangle$	id	$\langle \text{unaryminus} \rangle$
$\langle \text{factor} \rangle$	-	$\langle \text{unaryminus} \rangle$
$\langle \text{funct-declaration} \rangle$	function	function id ($\langle \text{param-list} \rangle$) $\langle \text{ret-type} \rangle$ begin $\langle \text{stat-seq} \rangle$ end ;
$\langle \text{funct-declaration-list} \rangle$	function	$\langle \text{funct-declaration} \rangle \langle \text{funct-declaration-list} \rangle$
$\langle \text{funct-declaration-list} \rangle$	in	ϵ
$\langle \text{id-list} \rangle$	id	id $\langle \text{id-list-tail} \rangle$
$\langle \text{id-list-tail} \rangle$:	ϵ
$\langle \text{id-list-tail} \rangle$,	$\langle \text{id} \rangle \langle \text{id-list-tail} \rangle$
$\langle \text{lvalue} \rangle$	id	id $\langle \text{lvalue-tail} \rangle$
$\langle \text{lvalue-tail} \rangle$	[[$\langle \text{expr} \rangle$] $\langle \text{lvalue-tail} \rangle$
$\langle \text{lvalue-tail} \rangle$:=	ϵ
$\langle \text{lvalue-tail} \rangle$	*	ϵ
$\langle \text{lvalue-tail} \rangle$	/	ϵ
$\langle \text{lvalue-tail} \rangle$	+	ϵ
$\langle \text{lvalue-tail} \rangle$	-	ϵ
$\langle \text{lvalue-tail} \rangle$	=	ϵ
$\langle \text{lvalue-tail} \rangle$	<	ϵ
$\langle \text{lvalue-tail} \rangle$	>	ϵ
$\langle \text{lvalue-tail} \rangle$	<=	ϵ
$\langle \text{lvalue-tail} \rangle$	>=	ϵ
$\langle \text{lvalue-tail} \rangle$	<>	ϵ
$\langle \text{lvalue-tail} \rangle$	&	ϵ
$\langle \text{lvalue-tail} \rangle$		ϵ
$\langle \text{lvalue-tail} \rangle$)	ϵ
$\langle \text{lvalue-tail} \rangle$,	ϵ

$\langle \text{lvalue-tail} \rangle$	\mid	ϵ
$\langle \text{lvalue-tail} \rangle$	then	ϵ
$\langle \text{lvalue-tail} \rangle$	do	ϵ
$\langle \text{lvalue-tail} \rangle$	to	ϵ
$\langle \text{lvalue-tail} \rangle$;	ϵ
$\langle \text{mulop} \rangle$	*	*
$\langle \text{mulop} \rangle$	/	/
$\langle \text{optional-init} \rangle$:=	:= $\langle \text{const} \rangle$
$\langle \text{optional-init} \rangle$;	ϵ
$\langle \text{orexpr} \rangle$	($\langle \text{andexpr} \rangle \langle \text{orexpr-tail} \rangle$
$\langle \text{orexpr} \rangle$	nil	$\langle \text{andexpr} \rangle \langle \text{orexpr-tail} \rangle$
$\langle \text{orexpr} \rangle$	STRLIT	$\langle \text{andexpr} \rangle \langle \text{orexpr-tail} \rangle$
$\langle \text{orexpr} \rangle$	INTLIT	$\langle \text{andexpr} \rangle \langle \text{orexpr-tail} \rangle$
$\langle \text{orexpr} \rangle$	id	$\langle \text{andexpr} \rangle \langle \text{orexpr-tail} \rangle$
$\langle \text{orexpr} \rangle$	-	$\langle \text{andexpr} \rangle \langle \text{orexpr-tail} \rangle$
$\langle \text{orexpr-tail} \rangle$	&	$\langle \text{andop} \rangle \langle \text{andexpr} \rangle \langle \text{orexpr-tail} \rangle$
$\langle \text{orexpr-tail} \rangle$		ϵ
$\langle \text{orexpr-tail} \rangle$)	ϵ
$\langle \text{orexpr-tail} \rangle$,	ϵ
$\langle \text{orexpr-tail} \rangle$]	ϵ
$\langle \text{orexpr-tail} \rangle$	then	ϵ
$\langle \text{orexpr-tail} \rangle$	do	ϵ
$\langle \text{orexpr-tail} \rangle$	to	ϵ
$\langle \text{orexpr-tail} \rangle$;	ϵ
$\langle \text{orop} \rangle$		
$\langle \text{param} \rangle$	id	id : $\langle \text{type-id} \rangle$
$\langle \text{param-list} \rangle$)	ϵ
$\langle \text{param-list} \rangle$	id	$\langle \text{param} \rangle \langle \text{param-list-tail} \rangle$
$\langle \text{param-list-tail} \rangle$)	ϵ
$\langle \text{param-list-tail} \rangle$,	, $\langle \text{param} \rangle \langle \text{param-list-tail} \rangle$
$\langle \text{ret-type} \rangle$	begin	ϵ
$\langle \text{ret-type} \rangle$:	: $\langle \text{type-id} \rangle$
$\langle \text{stat-assign} \rangle$	id	id $\langle \text{stat-assign-id} \rangle$
$\langle \text{stat-assign} \rangle$	-	- $\langle \text{unaryminus} \rangle \langle \text{stat-assign-tail} \rangle$
$\langle \text{stat-assign} \rangle$	(($\langle \text{expr} \rangle$) $\langle \text{stat-assign-tail} \rangle$
$\langle \text{stat-assign} \rangle$	nil	$\langle \text{const} \rangle \langle \text{stat-assign-tail} \rangle$
$\langle \text{stat-assign} \rangle$	STRLIT	$\langle \text{const} \rangle \langle \text{stat-assign-tail} \rangle$
$\langle \text{stat-assign} \rangle$	INTLIT	$\langle \text{const} \rangle \langle \text{stat-assign-tail} \rangle$
$\langle \text{stat-assign-id} \rangle$;	ϵ
$\langle \text{stat-assign-id} \rangle$		$\langle \text{lvalue-tail} \rangle \langle \text{stat-assign-tail} \rangle$
$\langle \text{stat-assign-id} \rangle$	&	$\langle \text{lvalue-tail} \rangle \langle \text{stat-assign-tail} \rangle$
$\langle \text{stat-assign-id} \rangle$	<>	$\langle \text{lvalue-tail} \rangle \langle \text{stat-assign-tail} \rangle$
$\langle \text{stat-assign-id} \rangle$	>=	$\langle \text{lvalue-tail} \rangle \langle \text{stat-assign-tail} \rangle$
$\langle \text{stat-assign-id} \rangle$	<=	$\langle \text{lvalue-tail} \rangle \langle \text{stat-assign-tail} \rangle$
$\langle \text{stat-assign-id} \rangle$	>	$\langle \text{lvalue-tail} \rangle \langle \text{stat-assign-tail} \rangle$
$\langle \text{stat-assign-id} \rangle$	<	$\langle \text{lvalue-tail} \rangle \langle \text{stat-assign-tail} \rangle$
$\langle \text{stat-assign-id} \rangle$	=	$\langle \text{lvalue-tail} \rangle \langle \text{stat-assign-tail} \rangle$
$\langle \text{stat-assign-id} \rangle$	-	$\langle \text{lvalue-tail} \rangle \langle \text{stat-assign-tail} \rangle$
$\langle \text{stat-assign-id} \rangle$	+	$\langle \text{lvalue-tail} \rangle \langle \text{stat-assign-tail} \rangle$
$\langle \text{stat-assign-id} \rangle$	/	$\langle \text{lvalue-tail} \rangle \langle \text{stat-assign-tail} \rangle$
$\langle \text{stat-assign-id} \rangle$	*	$\langle \text{lvalue-tail} \rangle \langle \text{stat-assign-tail} \rangle$
$\langle \text{stat-assign-id} \rangle$	[$\langle \text{lvalue-tail} \rangle \langle \text{stat-assign-tail} \rangle$
$\langle \text{stat-assign-id} \rangle$	(($\langle \text{expr-list} \rangle$)

$\langle \text{stat-assign-tail} \rangle$;	ϵ
$\langle \text{stat-assign-tail} \rangle$		$\langle \text{expr-tail} \rangle$
$\langle \text{stat-assign-tail} \rangle$	&	$\langle \text{orexpr-tail} \rangle$
$\langle \text{stat-assign-tail} \rangle$	$\langle \rangle$	$\langle \text{andexpr-tail} \rangle$
$\langle \text{stat-assign-tail} \rangle$	\geq	$\langle \text{andexpr-tail} \rangle$
$\langle \text{stat-assign-tail} \rangle$	\leq	$\langle \text{andexpr-tail} \rangle$
$\langle \text{stat-assign-tail} \rangle$	$>$	$\langle \text{andexpr-tail} \rangle$
$\langle \text{stat-assign-tail} \rangle$	$<$	$\langle \text{andexpr-tail} \rangle$
$\langle \text{stat-assign-tail} \rangle$	$=$	$\langle \text{andexpr-tail} \rangle$
$\langle \text{stat-assign-tail} \rangle$	-	$\langle \text{compare-tail} \rangle$
$\langle \text{stat-assign-tail} \rangle$	+	$\langle \text{compare-tail} \rangle$
$\langle \text{stat-assign-tail} \rangle$	/	$\langle \text{term-tail} \rangle$
$\langle \text{stat-assign-tail} \rangle$	*	$\langle \text{term-tail} \rangle$
$\langle \text{stat-func-or-assign} \rangle$	(($\langle \text{expr-list} \rangle$) ;
$\langle \text{stat-func-or-assign} \rangle$:=	$\langle \text{lvalue-tail} \rangle := \langle \text{stat-assign} \rangle$;
$\langle \text{stat-func-or-assign} \rangle$	[$\langle \text{lvalue-tail} \rangle := \langle \text{stat-assign} \rangle$;
$\langle \text{stat-if-tail} \rangle$	else	else $\langle \text{stat-seq} \rangle$ endif ;
$\langle \text{stat-if-tail} \rangle$	endif	endif ;
$\langle \text{stat} \rangle$	if	if $\langle \text{expr} \rangle$ then $\langle \text{stat-seq} \rangle$ $\langle \text{stat-if-tail} \rangle$
$\langle \text{stat} \rangle$	while	while $\langle \text{expr} \rangle$ do $\langle \text{stat-seq} \rangle$ enddo ;
$\langle \text{stat} \rangle$	for	for id := $\langle \text{expr} \rangle$ to $\langle \text{expr} \rangle$ do $\langle \text{stat-seq} \rangle$ enddo ;
$\langle \text{stat} \rangle$	break	break ;
$\langle \text{stat} \rangle$	return	return $\langle \text{expr} \rangle$;
$\langle \text{stat} \rangle$	id	id $\langle \text{stat-func-or-assign} \rangle$
$\langle \text{stat-seq} \rangle$	if	$\langle \text{stat} \rangle$ $\langle \text{stat-seq-tail} \rangle$
$\langle \text{stat-seq} \rangle$	while	$\langle \text{stat} \rangle$ $\langle \text{stat-seq-tail} \rangle$
$\langle \text{stat-seq} \rangle$	for	$\langle \text{stat} \rangle$ $\langle \text{stat-seq-tail} \rangle$
$\langle \text{stat-seq} \rangle$	break	$\langle \text{stat} \rangle$ $\langle \text{stat-seq-tail} \rangle$
$\langle \text{stat-seq} \rangle$	return	$\langle \text{stat} \rangle$ $\langle \text{stat-seq-tail} \rangle$
$\langle \text{stat-seq} \rangle$	id	$\langle \text{stat} \rangle$ $\langle \text{stat-seq-tail} \rangle$
$\langle \text{stat-seq-tail} \rangle$	endif	ϵ
$\langle \text{stat-seq-tail} \rangle$	end	ϵ
$\langle \text{stat-seq-tail} \rangle$	enddo	ϵ
$\langle \text{stat-seq-tail} \rangle$	else	ϵ
$\langle \text{stat-seq-tail} \rangle$	if	$\langle \text{stat} \rangle$ $\langle \text{stat-seq-tail} \rangle$
$\langle \text{stat-seq-tail} \rangle$	while	$\langle \text{stat} \rangle$ $\langle \text{stat-seq-tail} \rangle$
$\langle \text{stat-seq-tail} \rangle$	for	$\langle \text{stat} \rangle$ $\langle \text{stat-seq-tail} \rangle$
$\langle \text{stat-seq-tail} \rangle$	break	$\langle \text{stat} \rangle$ $\langle \text{stat-seq-tail} \rangle$
$\langle \text{stat-seq-tail} \rangle$	return	$\langle \text{stat} \rangle$ $\langle \text{stat-seq-tail} \rangle$
$\langle \text{stat-seq-tail} \rangle$	id	$\langle \text{stat} \rangle$ $\langle \text{stat-seq-tail} \rangle$
$\langle \text{term} \rangle$	-	$\langle \text{factor} \rangle$ $\langle \text{term-tail} \rangle$
$\langle \text{term} \rangle$	id	$\langle \text{factor} \rangle$ $\langle \text{term-tail} \rangle$
$\langle \text{term} \rangle$	INTLIT	$\langle \text{factor} \rangle$ $\langle \text{term-tail} \rangle$
$\langle \text{term} \rangle$	STRLIT	$\langle \text{factor} \rangle$ $\langle \text{term-tail} \rangle$
$\langle \text{term} \rangle$	nil	$\langle \text{factor} \rangle$ $\langle \text{term-tail} \rangle$
$\langle \text{term} \rangle$	($\langle \text{factor} \rangle$ $\langle \text{term-tail} \rangle$
$\langle \text{term-tail} \rangle$	*	$\langle \text{mulop} \rangle$ $\langle \text{factor} \rangle$ $\langle \text{term-tail} \rangle$
$\langle \text{term-tail} \rangle$	/	$\langle \text{mulop} \rangle$ $\langle \text{factor} \rangle$ $\langle \text{term-tail} \rangle$
$\langle \text{term-tail} \rangle$)	ϵ
$\langle \text{term-tail} \rangle$		ϵ
$\langle \text{term-tail} \rangle$	&	ϵ
$\langle \text{term-tail} \rangle$	-	ϵ
$\langle \text{term-tail} \rangle$	+	ϵ

$\langle \text{term-tail} \rangle$	$\langle \rangle$	ϵ
$\langle \text{term-tail} \rangle$	$>=$	ϵ
$\langle \text{term-tail} \rangle$	$<=$	ϵ
$\langle \text{term-tail} \rangle$	$>$	ϵ
$\langle \text{term-tail} \rangle$	$<$	ϵ
$\langle \text{term-tail} \rangle$	$=$	ϵ
$\langle \text{term-tail} \rangle$	$;$	ϵ
$\langle \text{term-tail} \rangle$	to	ϵ
$\langle \text{term-tail} \rangle$	do	ϵ
$\langle \text{term-tail} \rangle$	then	ϵ
$\langle \text{term-tail} \rangle$	$]$	ϵ
$\langle \text{term-tail} \rangle$	$,$	ϵ
$\langle \text{tiger-program} \rangle$	let	let $\langle \text{declaration-segment} \rangle$ in $\langle \text{stat-seq} \rangle$ end
$\langle \text{type} \rangle$	array	array [INTLIT] $\langle \text{type-dim} \rangle$ of $\langle \text{type-id} \rangle$
$\langle \text{type} \rangle$	id	$\langle \text{type-id} \rangle$
$\langle \text{type-declaration-list} \rangle$	type	$\langle \text{type-declaration} \rangle$ $\langle \text{type-declaration-list} \rangle$
$\langle \text{type-declaration-list} \rangle$	var	ϵ
$\langle \text{type-declaration-list} \rangle$	function	ϵ
$\langle \text{type-declaration-list} \rangle$	in	ϵ
$\langle \text{type-declaration} \rangle$	type	type id = $\langle \text{type} \rangle$;
$\langle \text{type-dim} \rangle$	[[INTLIT] $\langle \text{type-dim} \rangle$
$\langle \text{type-dim} \rangle$	of	ϵ
$\langle \text{type-id} \rangle$	id	id
$\langle \text{unaryminus} \rangle$	(($\langle \text{expr} \rangle$)
$\langle \text{unaryminus} \rangle$	nil	$\langle \text{const} \rangle$
$\langle \text{unaryminus} \rangle$	STRLIT	$\langle \text{const} \rangle$
$\langle \text{unaryminus} \rangle$	INTLIT	$\langle \text{const} \rangle$
$\langle \text{unaryminus} \rangle$	id	$\langle \text{lvalue} \rangle$
$\langle \text{var-declaration-list} \rangle$	function	ϵ
$\langle \text{var-declaration-list} \rangle$	in	ϵ
$\langle \text{var-declaration-list} \rangle$	var	$\langle \text{var-declaration} \rangle$ $\langle \text{var-declaration-list} \rangle$
$\langle \text{var-declaration} \rangle$	var	var $\langle \text{id-list} \rangle$: $\langle \text{type-id} \rangle$ $\langle \text{optional-init} \rangle$;