MAlice Language Specification

Peter Hamilton Sarah Tattersall

December 2, 2011

1 BNF Grammar

S’	$\rightarrow$	code_seperator
code_seperator	$\rightarrow$	statement_list function_seperator functions |
		statement_list
statement_list	$\rightarrow$	statement seperator |
		statement seperator statement_list
seperator	$\rightarrow$	**‘and’** | **‘but’** | **‘then’** | **‘,’** | **‘?’** | **‘.’**
statement	$\rightarrow$	expression **‘spoke’** |
		expression **‘said’** **‘Alice’** |
		‘what’ **‘was’** expression |
		expression **‘thought’** **‘Alice’** |
		statement **‘too’** |
		id **‘was’** **‘a’** type |
		id **‘became’** expression |
		array_access **‘became’** expression |
		id **‘had’** expression type |

‘eventually’ ‘(’ expression_logical ‘)’ **‘because’** statement_list **‘enough’** **‘times’** |
‘either’ ‘(’ expression_logical ‘)’ **‘so’** statement_list **‘or’** statement_list logical_ending |
‘perhaps’ ‘(’ expression_logical ‘)’ **‘so’** statement_list logical_ending |
‘perhaps’ ‘(’ expression_logical ‘)’ **‘so’** statement_list logical_clauses logical_ending |
‘Alice found’ expression |
expression

expression → id ‘(’ function_arguments ‘)’ |
id **‘went’** **‘through’** id |
‘(’ expression ‘)’ |
‘∼’ expression |
id **‘drank’** |
id **‘ate’** |
expression ‘|’ expression |
expression ‘ˆ’ expression |
expression ‘&’ expression |
expression ‘+’ expression |
expression ‘-’ expression |
expression ‘*’ expression |
expression ‘/’ expression |
expression ‘%’ expression |
‘-’ expression | *(Uses UMINUS precedence)*
expression_logical |
array_access |
factor

expression_logical → expression ‘==’ expression |
expression ‘<’ expression |
expression ‘>’ expression |
expression ‘>=’ expression |
expression ‘<=’ expression |
expression ‘!=’ expression |
expression ‘&&’ expression |
expression ‘||’ expression

array_access → id**‘ś’** expression **‘piece’**

factor	→	number | letter | id | sentence
type	→	**‘number’** | **‘letter’** | **‘sentence’**
functions	→	function function_seperator functions | function
function_seperator	→	**‘The’ ‘room’** | **‘The’ ‘Looking-Glass’**
function	→	id **‘(’** arguments **‘)’ ‘contained’ ‘a’** type statement_list | id **‘(’** arguments **‘)’ ‘contained’ ‘a’** type id **‘changed’ ‘a’** type statement_list
logical_clauses	→	logical_clause logical_clauses | **‘Alice’ ‘was’ ‘unsure’ ‘which’**
logical_clause	→	**‘or’ ‘maybe’ ‘(’** expression_logical **‘)’ ‘so’** statement_list | **‘or’** statement_list
arguments	→	argument **‘,’** arguments argument |
argument	→	type id | **‘spider’** type id
function_arguments	→	function_argument **‘,’** function_arguments | function_argument
function_argument	→	expression

2 Precedences

```
precedence = (
('left', 'L_OR'),
('left', 'L_AND'),
('left', 'L_EQUAL', 'L_NOT_EQUAL'),
('left', 'L_LESS_THAN', 'L_LESS_THAN_EQUAL', 'L_GREATER_THAN', 'L_GREATER_THAN_EQUAL'),
('left', 'B_OR'),
('left', 'B_XOR'),
('left', 'B_AND'),
('left', 'PLUS', 'MINUS'),
('left', 'MULTIPLY', 'DIVIDE', 'MOD'),
('right', 'INCREMENT', 'DECREMENT', 'B_NOT', 'UMINUS'),
('left', 'L_PAREN', 'R_PAREN'),
)
```