MAlice Language Specification

Peter Hamilton Sarah Tattersall

December 1, 2011

1 BNF Grammar

S'	→	code_seperator
code_seperator	→	statement_list function_seperator functions | statement_list
statement_list	→	statement seperator | statement seperator statement_list
seperator	→	**'and'** | **'but'** | **'then'** | **','** | **'?'** | **'.'**
statement	→	expression **'spoke'** | expression **'said' 'Alice'** | **'Alice' 'found'** expression | **'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 **‘Alice’** **‘was’** **‘unsure’** **‘which’** |
‘perhaps’ **‘(’** expression_logical **‘)’** **‘so’** statement_list **‘Alice’** **‘was’** **‘unsure’** |
‘perhaps’ **‘(’** expression_logical **‘)’** **‘so’** statement_list logical_clauses |
expression

expression → id **‘(’** function_arguments **‘)’** |
id **‘went’** **‘through’** id |
‘(’ expression **‘)’** |
‘∼’ expression |
id **‘drank’** |
id **‘ate’** |
expression **‘|’** expression |
expression **‘^’** expression |
expression **‘&’** expression |
expression **‘+’** expression |
expression **‘-’** expression |
expression **‘*’** expression |
expression **‘/’** expression |
expression **‘%’** expression
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 expression **‘piece’**

factor	→	number | letter | id | sentence
type	→	**‘number’** | **‘letter’** | **‘sentence’**
functions	→	function function_seperator functions | ref_function function_seperator functions | ref_function | function
function_seperator	→	**‘The’ ‘room’** | **‘The’ ‘Looking-Glass’**
function	→	id **‘(’** arguments **‘)’ ‘contained’ ‘a’** type statement_list | id **‘(’** arguments **‘)’ ‘contained’ ‘a’** type
ref_function	→	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'),
  ('left', 'L_PAREN', 'R_PAREN'),
)
```