EXAMPLES

```
ROBOT_R
VARS A001, A002;
PROCS
Assign [ | | assingTo: 1, A001; assingTo: 2, A002]
Proc1 [ | a, b | pick: a, Balloons; put: b, Chips ]
goSouth [ | | if: canJumpInDir: 2, south then: [moveToThe: 1, left] else: [nop:] ]
[
Proc1: 2, 2
]
```

```
ROBOT_R
VARS x, y;
PROCS
Move1 [ | a, b | move: a; turn: around ]
Condi2 [ | | while: canMoveToThe: 1, front do: [ turn: right ] ]
[
Move1: 1, 2
]
```

```
ROBOT_R
VARS one, two;
PROCS
Proc1 [ | one, two | goto: one, two ]
Proc2 [ | a, b | face: east; jumpToThe: b, left]
[
Proc1: 1, 5;
Proc2: 1, 4
]
```

```
ROBOT_R
PROCS
Hola [ | | goto: 2, 4; if:  facing: north then: [ nop: ]
else: [ if: facing: west then: [ nop: ] else: [ nop:; assignTo: 2, n] ] ]
[
nop:
]
```

Translation of keywords in the lexer

Keyword	Translation
ROBOT_R	R
VARS	V
IdL	v
IdU	v
PROCS	P
assingTo	a
Num	#
goto	g
move	m
turn	t
face	f
put	p
pick	i
moveToThe	M
moveInDir	o
jumpToThe	jx
jumpInDir	J
nop	n
front	A
back	A
left	A
right	A
around	A
north	Z
south	Z
east	Z
west	Z
Balloons	X
Chips	X
if	I
then	T
else	E
facing	F
canPut	c
canPick	k
canMoveInDir	d
canJumpInDir	G
canMoveToThe	h
canJumpToThe	H
not	O
while	W
do	D
repeat	r

Grammar

Non-terminals

S: Start Symbol

B: Block of instructions

V: Declaration of Variables

P: Procedure declaration

PD: Procedure definition

Ins: Sequence of instruction

Cm: Command

CtS: Control Structure

PC: Procedure call

LV: List of variables

PD: Procedure definition

Prm: Parameters

Cdl: Conditional

RpTms: RepeatTimes

Lp: Loop

Cdn: Condition

NV: number or a variable's name

Special tokens

<ID>: string of alphanumeric characters that begins with a letter; a name.

<NUM>: number.

<D>: left, right, or around.

<O>: north, south, east, or west.

<X>: Balloons or Chips.

Production rules

S → <ROBOT_R> [V] [P] B <EOF>

V → <VARS> LV <;>

LV → <ID> (<,> <ID>)*

P → <PROCS> (PD)+

PD → <ID> <[> <|> [Prm] <|> Ins <]>

Prm → <ID> (<,> <ID>)*

Ins → (Cm | CtS | PC) (<;> Cm | <;> CtS | <;> <ID>)*

Cm → assignTo | goto | move | turn | face | put | pick | moveToThe | moveInDir | jumpToThe | jumpInDir | nop

assignTo → <assignTo> <:> <NUM> <,> <ID>

goto → <goto> <:> NV <,> NV

move → <move> <:> NV

turn → <turn> <:> <D>

face → <face> <:> <O>

put → <put> <:> NV <,> <X>

pick → <pick> <:> NV <,> <X>

moveToThe → <moveToThe> <:> NV <,> <D>

moveInDir → <moveInDir> <:> NV <,> <O>

jumpToThe → <jumpToThe> <:> NV <,> <D>

jumpInDir → <jumpInDir> <:> NV <,> <O>

nop → <nop> <:>

NV → <ID> | <NUM>

CtS → Cdl | Lp | RpTms

Cdl → <IF> <:> Cdn <THEN> <:> B <ELSE> <:> B

Cdn → facing | canPut | canPick |
canMoveInDir | canJumpInDir |
canMoveToThe | canJumpToThe | not

facing → <facing> <:> <O>

canPut → <canPut> <:> NV <,> <X>

canPick → <canPick> <:> NV <,> <X>

canMoveInDir → <canMoveInDir> <:>
NV <,> <D>

canJumpInDir → <canJumpInDir> <:>
NV <,> <D>

canMoveToThe → <canMoveToThe>
<:> NV <,> <O>

canJumpToThe → <canJumpToThe> <:>
NV <,> <O>

not → <not> <:> Cdn

Lp → <WHILE> <:> Cdn <DO> <:> B

RpTms → <REPEAT> <:> NV B

PC → <ID> <:> [NV (<,> NV)*]

B → <[> [Ins] <]>

Note:

1. Every variable must be declared.
2. A block of instructions can be empty.
3. A declaration of variables must have at least one name.
4. A procedure declaration must have one or more procedure definitions.
5. Only the declaration of variables or only the procedure declaration is allowed.
6. “assign_to” command’s <ID> can only reference a variable’s name