URSA: Language Description

Predrag Janičić, Faculty of Mathematics, Studentski trg 16, 11000 Belgrade, Serbia

⟨program⟩	::=	⟨procedure def⟩* ⟨statement⟩*
⟨procedure def⟩	::=	"procedure" ⟨procedure name⟩ ("(" ")" |
		"(" (⟨num var id⟩ | ⟨bool var id⟩) ("," (⟨num var id⟩ | ⟨bool var id⟩))* ")")
		"{" ⟨statement⟩* "}"
⟨procedure name⟩	::=	⟨letter⟩(⟨letter⟩ | ⟨digit⟩)*
⟨statement⟩	::=	"{" ⟨statement⟩* "}"
		| ⟨num var⟩ ⟨assign num op⟩ ⟨num expr⟩ ";"
		| ⟨num var⟩ ⟨num op postfix⟩ ";"
		| ⟨bool var⟩ ⟨assign bool op⟩ ⟨bool expr⟩ ";"
		| "while" "(" ⟨bool expr⟩ ")" ⟨statement⟩
		| "for" "(" ⟨statement⟩ ";" ⟨bool expr⟩ ";" ⟨statement⟩ ")" ⟨statement⟩
		| "if" "(" ⟨bool expr⟩ ")" ⟨statement⟩ ["else" ⟨statement⟩]
		| "call" ⟨procedure name⟩ ("(" ")" |
		"(" (⟨num expr⟩ | ⟨bool expr⟩) ("," (⟨num expr⟩ | ⟨bool expr⟩))* ")") ";"
		| "minimize" "(" ⟨num var⟩ "," ⟨num const⟩ "," ⟨num const⟩ ")" ";"
		| "maximize" "(" ⟨num var⟩ "," ⟨num const⟩ "," ⟨num const⟩ ")" ";"
		| "assert" "(" ⟨bool expr⟩ (";" ⟨bool expr⟩)* ")" ";"
		| "assert_all" "(" ⟨bool expr⟩ (";" ⟨bool expr⟩)* ")" ";"
		| "print" (⟨num expr⟩ | ⟨bool expr⟩) ";"
		| "printx" ⟨num expr⟩ ";"
		| "printb" ⟨num expr⟩ ";"
		| "listvars" ";"
		| "clear" ";"
		| "halt" ";"
⟨num expr⟩	::=	⟨num const⟩
		| ⟨num var⟩
		| ⟨un num op⟩ ⟨num expr⟩
		| ⟨num expr⟩ ⟨num op⟩ ⟨num expr⟩
		| ite "(" ⟨bool expr⟩ "," ⟨num expr⟩ "," ⟨num expr⟩")"
		| "sgn" "(" ⟨num expr⟩ ")"
		| "bool2num" "(" ⟨bool expr⟩ ")"
		| "(" ⟨num expr⟩ ")"
⟨num var⟩	::=	⟨num var id⟩
		| ⟨num var id⟩ "[" ⟨num expr⟩ "]"
		| ⟨num var id⟩ "[" ⟨num expr⟩ "]" "[" ⟨num expr⟩ "]"
⟨num const⟩	::=	(⟨digit⟩)+
⟨num var id⟩	::=	"n"(⟨letter⟩ | ⟨digit⟩)*
⟨assign num op⟩	::=	"=" | "+=" | "-=" | "*=" | "&=" | "|=" | "^=" | "<<=" | ">>="
⟨num op⟩	::=	"+" | "-" | "*" | "&" | "|" | "^" | "<<" | ">>"
⟨un num op⟩	::=	"−" | "~"
⟨num op postfix⟩	::=	"++" | "--"
⟨num rel⟩	::=	"<" | ">" | "<=" | ">=" | "==" | "!="
⟨bool expr⟩	::=	⟨bool const⟩
		| ⟨bool var⟩
		| ⟨bool expr⟩ ⟨bool op⟩ ⟨bool expr⟩
		| ⟨un bool op⟩ ⟨bool expr⟩
		| ⟨num expr⟩ ⟨num rel⟩ ⟨num expr⟩
		| ite "(" ⟨bool expr⟩ "," ⟨bool expr⟩ "," ⟨bool expr⟩")"
		| "num2bool" "(" ⟨num expr⟩ ")"
		| "(" ⟨bool expr⟩ ")"
⟨bool const⟩	::=	("true" | "false")
⟨bool var⟩	::=	⟨bool var id⟩
		| ⟨bool var id⟩ "[" ⟨num expr⟩ "]"
		| ⟨bool var id⟩ "[" ⟨num expr⟩ "]" "[" ⟨num expr⟩ "]"
⟨bool var id⟩	::=	"b"(⟨letter⟩ | ⟨digit⟩)*
⟨assign bool op⟩	::=	"=" | "&&=" | "||=" | "^^="
⟨bool op⟩	::=	"&&" | "||" | "^^"
⟨un bool op⟩	::=	"!"

Constants in URSA specification may contain constants written in decimal form (e.g., 26), in hexadecimal form (e.g, 0x1A or 0x1a), or in binary form (e.g., 0b00011010). The command `print` prints numbers in decimal form, unless the number of bits is bigger than the size of integers on the computer e.g., 32/64 – then the numbers are printed in hexadecimal form. The command `printx` prints numbers in hexadecimal form, and the command command `printb` prints numbers in binary form. Solutions of the constraints are printed in decimal form, unless the number of bits is bigger than the size of integers on the computer e.g., 32/64 – then the solutions are printed in hexadecimal form.