1 Syntax of Concurrent Boolean Programs

Syntax			Description
prog	::=	*decl** *proc**	A program is a list of global variable declarations followed by a list of predure definitions
decl	::=	**decl** *id*$^+$ **;**	Declaration of variables
id	::=	[**a-zA-Z_**][**a-zA-Z0-9_**]*	An identifier is a regular C-style identifier
proc	::=	*type id*(*id**) **begin** *enforce sseq* **end**	Procedure definition
type	::=	**void**	Procedures can return an arbitrary number of values
	|	**bool**	
	|	**bool** <*id*$^+$ >	
enforce	::=	**enforce** (*expr*) **;**	Restriction on valuations of variables
	|		
sseq	::=	*lstmt*$^+$	Sequence of statements
lstmt	::=	*stmt*	
	|	*id***:**$^+$ *stmt*	Labeled statement
stmt	::=	**skip ;**	
	|	**goto** *id*$^+$ **;**	Nondeterministic goto
	|	**return** *id** **;**	
	|	*id*$^+$:= *expr*$^+$ **constrain** *expr* **;**	Parallel assignment
	|	**if** (*expr*) **then** *sseq* **else** *sseq* **fi**	Conditional statement
	|	**assert** (*expr*) **;**	Assert statement
	|	**assume** (*expr*) **;**	Assume statement
	|	*id** := *id* (*expr*) **;**	Procedure call
	|	**start_thread** *id* **;**	Thread creation
	|	**end_thread ;**	Thread termination
	|	**atomic_begin ;**	Beginning of atomic section
	|	**atomic_end ;**	Ending of atomic section
expr	::=	*expr binop expr*	
	|	**!** *expr*	Negation
	|	(*expr*)	
	|	*const*	
	|	*id*	Variable
	|	$\star$	Non-deterministic choice
	|	**schoose** [*expr*, *expr*] **;**	Non-deterministic choice
binop	::=	'|' | '**&**' | '∧' | '**=**' | '**!=**' | ' $\Longrightarrow$ '	Logical connectives
const	::=	**0** / **1**	False / True

Table 1: Syntax of concurrent Boolean Programs (adapted from [1] and extended to SatAbs' syntax).

References

[1] Ball, T., Rajamani, S.K.: Bebop: A symbolic model checker for Boolean programs. In: Model Checking and Software Verification (SPIN). Volume 1885 of LNCS., Springer (2000) 113–130