Simple Bounded LTL Model Checking

Timo Latvala, Armin Biere, Keijo Heljanko, and Tommi Junttila

Timo.Latvala@hut.fi

Laboratory for Theoretical Computer Science

Helsinki University of Technology

Introduction

- Bounded model checking (BMC) is an efficient way of implementing *symbolic model checking*.
- Find violations to LTL specifications.
- BMC: given a system model M, a temporal logic specification ψ, and bound k create a Boolean formula which is satisfiable *iff* M has a counterexample, of length k, to ψ.
- Basic form: $|[M]|_k \wedge |[\psi]|_k$

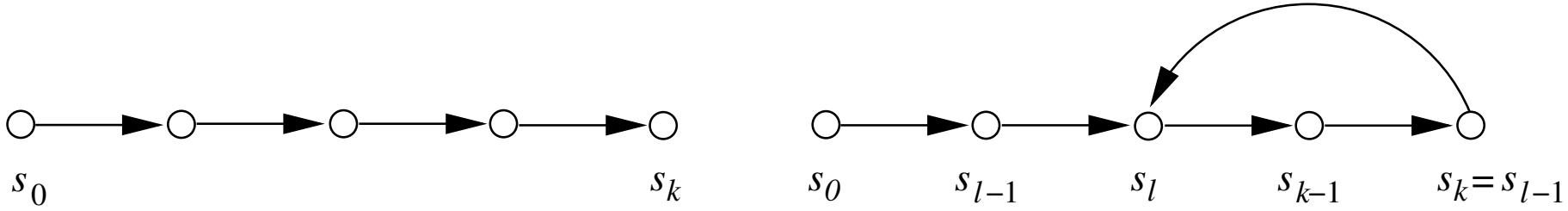
BMC: Pros and Cons

+ Boolean formulas can be more compact BDDs.
+ Leverages efficient SAT-solver technology.
+ Short counterexamples.
- Basic method is incomplete.
- Not always better than BDD-based methods.

Related Work

- Original BMC paper: Biere et. al., TACAS 1999.
- Improving the basic encoding: Cimatti et. al, VMCAI 2002.
- Fixpoint encoding: Sheridan et. al., FMCAD 2002.
- A stable model semantics (logic programs) encoding, Heljanko and Niemelä, LPNMR 2001.

BMC: Basic Encoding Form



(a) no loop

(k,l)–loop

- For each $1 \leq l \leq k$:
  - write constraints such that the formula is satisfiable *iff* the selected (k, l)-loop is a valid counterexample.
-

$$\bigvee_{l=1}^{k} l \, [\![\psi]\!]_k$$

A New Encoding

- Original encoding and its improved versions are *non-linear* in k.
- Basic idea: for lasso-shaped Kripke structures CTL and LTL coincide:

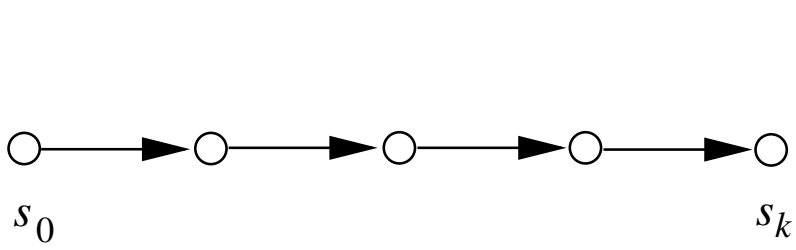
$$\mathbf{E}\left(\psi_1 \mathbf{U} \psi_2\right) \equiv \psi_1 \mathbf{U} \psi_2.$$

- Separate selection of path and model checking.
- Symbolically encode a CTL model checker.
- Derive encoding from fixpoint characterisation:

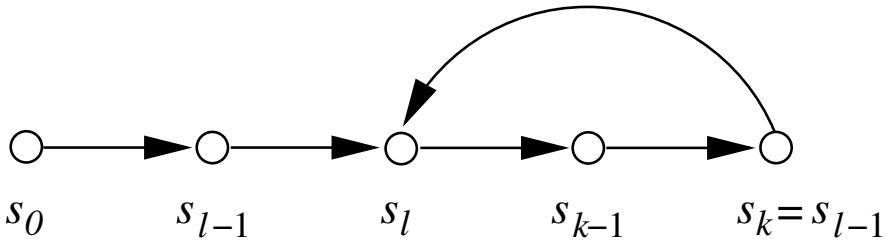
$$\mathbf{E}\left(\psi_1 \mathbf{U} \psi_2\right) \equiv \mu Z.\psi_2 \vee \left(\psi_1 \wedge \mathbf{EX} Z\right)$$

Encoding Paths I

- Encoding should select non-deterministically select a k-length lasso-shaped path.
- Introduce k fresh *loop selector variables* l_i:
  - $l_i \Rightarrow (s_{i-1} = s_k)$.
- Allow *at most one* loop selector to be true



(a) no loop



(k,l)–loop

Encoding Paths II

$$
\begin{aligned}
\left|[M]\right|_k &= I(s_0) \wedge \bigwedge_{i=1}^{k} T(s_{i-1}, s_i) \\
\left|[LoopConstraints]\right|_k &\Leftrightarrow Loop_k \wedge AtMostOne_k \\
Loop_k &\Leftrightarrow \bigwedge_{i=1}^{k} (l_i \Rightarrow (s_{i-1} = s_k)) \\
AtMostOne_k &\Leftrightarrow \bigwedge_{i=1}^{k} (SmallerExists_i \Rightarrow \neg l_i) \\
SmallerExists_1 &\Leftrightarrow \bot \\
SmallerExists_{i+1} &\Leftrightarrow SmallerExists_i \vee l_i, \text{ where } 0 < i \leq k
\end{aligned}
$$

Encoding I

$:=$	$i \leq k$	$i = k+1$
$\lvert[p]\rvert_i$	p_i	$\bigvee_{j=1}^{k} \left(l_j \wedge p_j\right)$
$\lvert[\neg p]\rvert_i$	$\neg p_i$	$\bigvee_{j=1}^{k} \left(l_j \wedge \neg p_j\right)$
$\lvert[\psi_1 \vee \psi_2]\rvert_i$	$\lvert[\psi_1]\rvert_i \vee \lvert[\psi_2]\rvert_i$	$\lvert[\psi_1]\rvert_i \vee \lvert[\psi_2]\rvert_i$
$\lvert[\psi_1 \wedge \psi_2]\rvert_i$	$\lvert[\psi_1]\rvert_i \wedge \lvert[\psi_2]\rvert_i$	$\lvert[\psi_1]\rvert_i \wedge \lvert[\psi_2]\rvert_i$
$\lvert[\mathbf{X}\psi]\rvert_i$	$\lvert[\psi]\rvert_{i+1}$	$\bigvee_{j=1}^{k} \left(l_j \wedge \lvert[\psi]\rvert_{j+1}\right)$

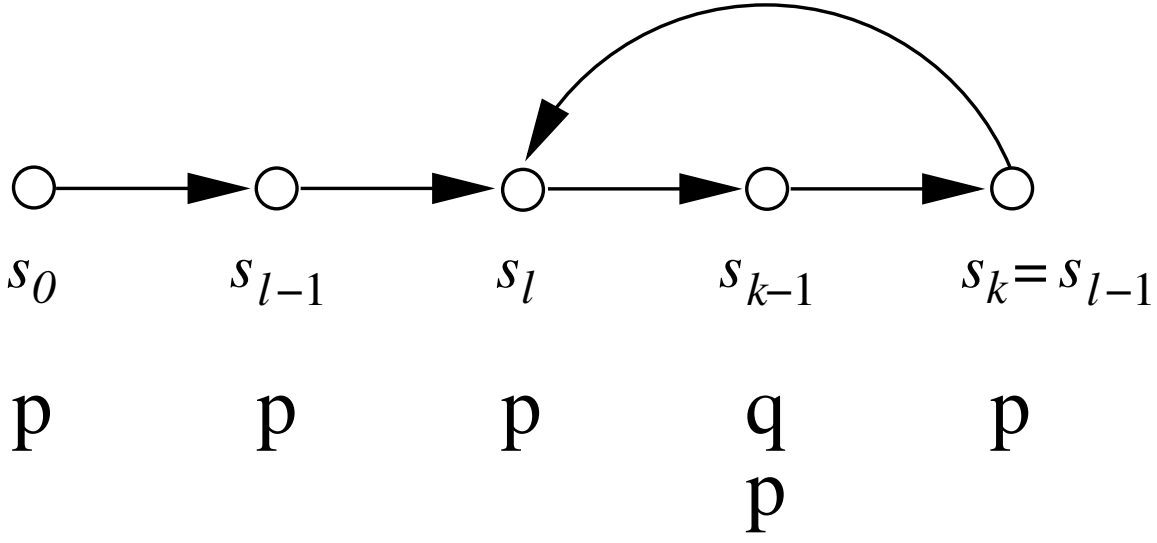
Encoding Until and Release

- The encoding of Until and Release are based on their fixpoint characterisations.
- Use an auxiliary encoding $\langle\langle \cdot \rangle\rangle$ to compute an approximation of the fixpoint
- The approximate values are refined to exact by the $|[\cdot]|$-encoding.
- The auxiliary encoding $\langle\langle \cdot \rangle\rangle_i$ is in fact exact at $i = l$.

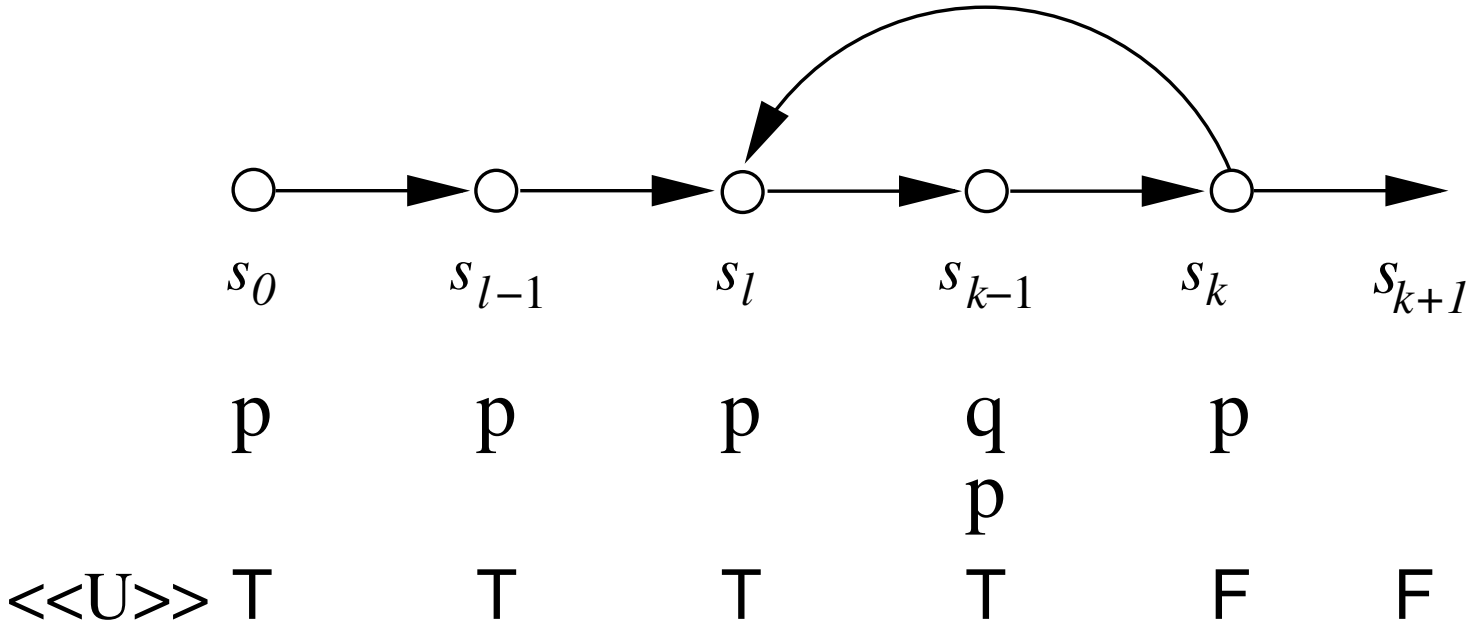
Encoding II

$:=$	$i \leq k$	$i = k+1$
$\lvert[\psi \mathbf{U} \varphi]\rvert_i$	$\lvert[\varphi]\rvert_i \vee \left(\lvert[\psi]\rvert_i \wedge \lvert[\psi \mathbf{U} \varphi]\rvert_{i+1}\right)$	$\bigvee_{j=1}^{k} \left(l_j \wedge \langle\langle \psi \mathbf{U} \varphi \rangle\rangle_j\right)$
$\lvert[\psi \mathbf{R} \varphi]\rvert_i$	$\lvert[\varphi]\rvert_i \wedge \left(\lvert[\psi]\rvert_i \vee \lvert[\psi \mathbf{R} \varphi]\rvert_{i+1}\right)$	$\bigvee_{j=1}^{k} \left(l_j \wedge \langle\langle \psi \mathbf{R} \varphi \rangle\rangle_j\right)$
$\langle\langle \psi \mathbf{U} \varphi \rangle\rangle_i$	$\lvert[\varphi]\rvert_i \vee \left(\lvert[\psi]\rvert_i \wedge \langle\langle \psi \mathbf{U} \varphi \rangle\rangle_{i+1}\right)$	$\bot$
$\langle\langle \psi \mathbf{R} \varphi \rangle\rangle_i$	$\lvert[\varphi]\rvert_i \wedge \left(\lvert[\psi]\rvert_i \vee \langle\langle \psi \mathbf{R} \varphi \rangle\rangle_{i+1}\right)$	$\top$

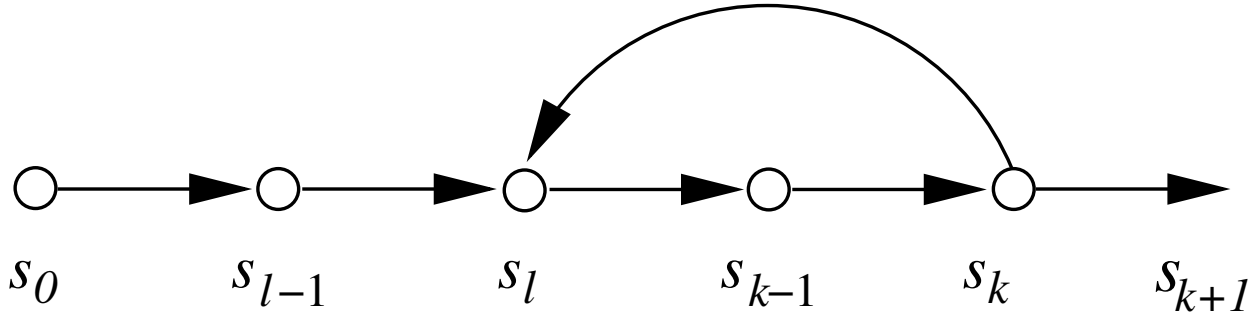
Example: $|[p\mathbf{U}q]|$



Example: $|[p\mathbf{U}q]|$



Example: $|[p\mathbf{U}q]|$



	p	p	p	q p	p	
<<U>>	T	T	T	T	F	F
||U||	T	T	T	T	T	

Main Result

Theorem
$|[M, \psi, k]|$ seen as Boolean circuit is linear in $|T|$, $|\psi|$, and k. More precisely, it is of the size $O(|I| + ((|T| + |\psi|) \cdot k))$.

Properties of the Encoding

- Unique model property.
- Monotonic circuit.
- Simple and easy to understand.

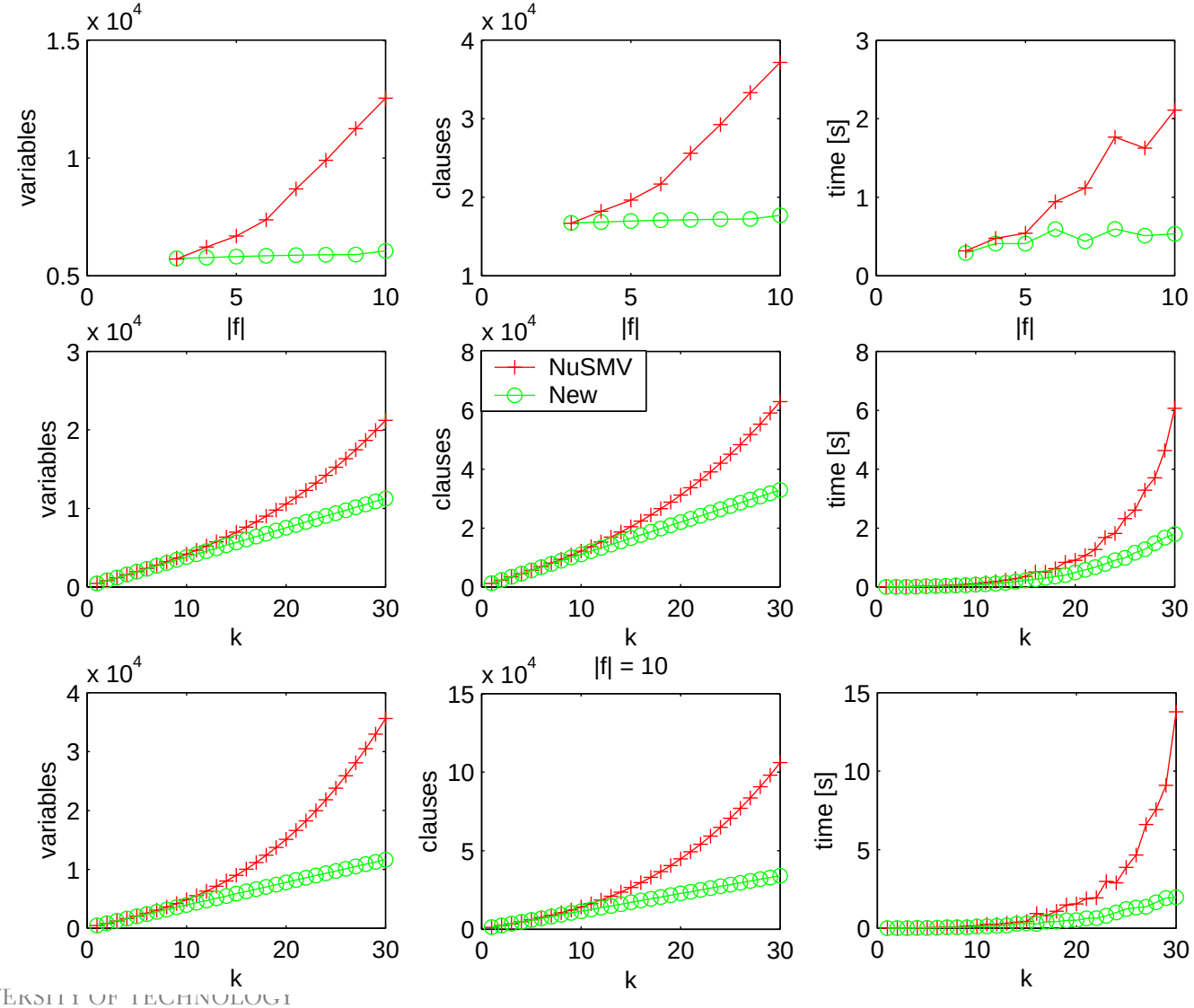
Experiments

- Random formulae on small random Kripke structures.
- Formula sizes between 3-12, and k from $0 - 50$.
- A few real-life examples.
- Compare with encoding of NuSMV and the Fixpoint encoding.
- Measure: number of variables and clauses in the CNF encoding, time to solve instance.

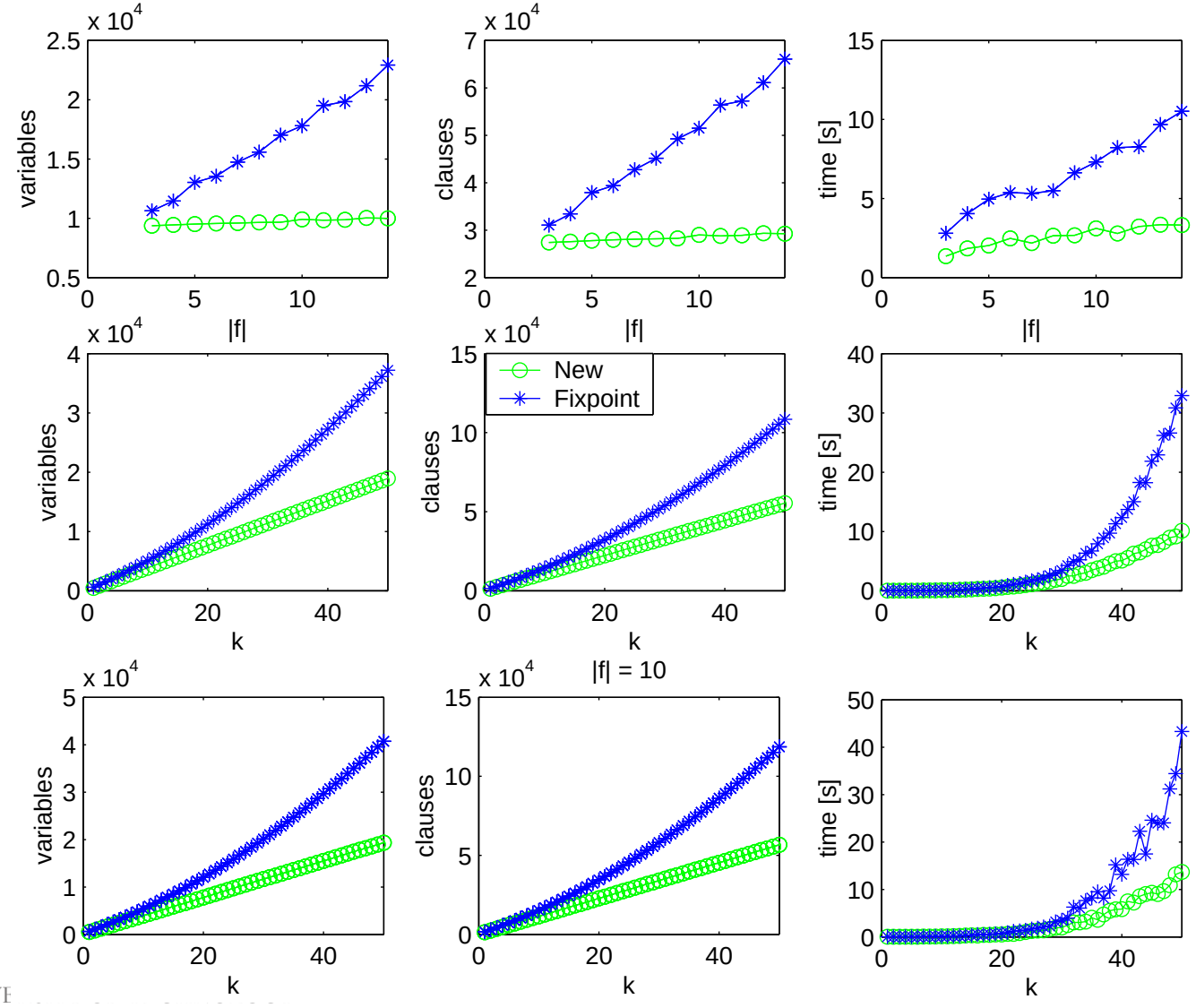
Benchmarks

Model	k	NuSMV			Fixpoint			New		
		vars	*clauses*	*time*	*vars*	*clauses*	*time*	*vars*	*clauses*	*time*
abp	16	19,476	57,373	32.3	18,643	54,637	43.7	18,024	52,969	7.4
brp	10	7,599	21,811	1.3	8,550	24,256	1.2	7,471	21,397	1.5
	15	11,494	33,226	18.7	13,150	37,636	22.0	11,116	32,047	17.9
	20	15,514	45,016	471	18,050	51,916	351	14,761	42,697	484
dme	10	53,400	141,438	2.0	54,407	144,022	0.9	53,293	141,087	2.6
	20	104,885	283,733	180	107,527	290,902	263	104,173	281,537	471
	30	156,870	427,528	1,199	161,847	441,382	1,855	155,053	421,987	1,544
pci	10	56,414	167,753	58.3	56,232	167,042	56.6	55,911	166,214	51.5
	15	85,359	254,133	568	84,372	250,947	370	83,756	249,279	382
	20	115,204	343,213	5,921	112,612	335,152	2,216	111,601	332,344	2,102
srg16	20	N/A	N/A	N/A	10,540	28,786	2.3	5,196	14,921	2.7
	40	N/A	N/A	N/A	25,600	71,686	16.6	10,336	29,841	22.3
	60	N/A	N/A	N/A	45,460	128,986	105	15,476	44,761	83.0

Benchmarks II



Benchmarks III



Conclusions and Future Work

- A new encoding linear in $|T|, |\psi|, k$.
- Performs experimentally well.

Future work:

- Generalise to full past LTL (See VMCAI'2005)
- Use monotonicity
- Incremental BMC