These benchmarks show the progress in LTL translation of various versions of Spot. Other tools are shown for comparison. In all experiments, smaller numbers are better.

Last updated on 2012-05-09 by Alexandre Duret-Lutz <adl@lrde.epita.fr>.

Translation of 188 formulas from the litterature

For references, see `bench/ltl2tgba/README` in the Spot distribution.

Cumulated sizes of automata for 188 formulas from the litterature

Products with a random state-space of 200 states

		$\Sigma\lvert A_{\neg\varphi}\rvert$		$\Sigma\lvert A_M \otimes A_{\neg\varphi}\rvert$	
		st.	tr.	st.	tr.
BA	Spin 6.1.0 (☠×6)	1 676	8 184	320 799	21 899 319
	ltl2ba 1.1	1 080	3 646	215 717	12 766 425
	Modella 1.5.9	1 390	4 562	274 081	10 916 278
	ltl3ba 1.0	936	2 839	186 989	10 827 554
	ltl3ba 1.0 -M	921	2 761	183 016	6 957 980
	ltl3ba 1.0 -S	858	2 356	171 391	9 578 700
	ltl3ba 1.0 -M -S	896	2 565	178 026	6 652 269
	Spot-0.6	834	2 623	166 579	9 090 819
	Spot-0.6 det.	834	2 623	165 677	6 258 743
	Spot-0.7.1	834	2 419	166 579	8 749 162
	Spot-0.7.1 det.	834	2 419	165 677	6 258 605
	Spot-0.7.1 WDBA	773	2 166	153 535	5 657 125
	Spot-0.8.3	834	2 419	166 579	8 749 162
	Spot-0.8.3 det.	834	2 419	165 677	6 258 605
	Spot-0.8.3 WDBA	770	2 159	152 935	5 633 081
	Spot-0.9	831	2 422	165 971	8 723 693
	Spot-0.9 det.	831	2 422	165 033	6 254 283
	Spot-0.9 WDBA	772	2 173	153 344	5 668 811
	Spot-0.9 Sim	789	2 205	157 003	5 802 744
	Spot-0.9 WDBA+Sim	746	2 032	148 514	5 436 445
TGBA	Spot-0.6	757	2 089	151 185	7 573 811
	Spot-0.6 det.	757	2 089	150 445	5 696 034
	Spot-0.7.1	757	2 089	151 185	7 573 811
	Spot-0.7.1 det.	757	2 089	150 445	5 696 034
	Spot-0.7.1 WDBA	705	1 886	140 100	5 156 767
	Spot-0.8.3	757	2 085	151 185	7 570 733
	Spot-0.8.3 det.	757	2 085	150 445	5 696 034
	Spot-0.8.3 WDBA	704	1 879	139 900	5 148 732
	Spot-0.9	745	2 049	148 785	7 480 832
	Spot-0.9 det.	745	2 049	148 045	5 630 589
	Spot-0.9 WDBA	702	1 875	139 500	5 144 095
	Spot-0.9 Sim	706	1 890	140 621	5 209 858
	Spot-0.9 WDBA+Sim	676	1 761	134 676	4 915 159

☠ = 15min timeout

Modella is run with all available optimizations: `-o1 -g -e -r12`.

Keys for Spot:

det.	More deterministic output (`-x`)
WDBA	Weak Deterministic Büchi Automaton minimization (`-x -Rm`)
Sim	Reduction based on direct simulation on TGBA (`-x -RDS`)
WDBA+Sim	All the above (`-x -Rm -RDS`)

All formula reductions are always applied (`-r7`), and SCC simplifications are always performed (`-R3`).

5 classes of formulas

This benchmark consists in 5 parameterized classes of formulas studied by Cichoń et al. (DEPCOS'09). Each class is translated with parameter n ranging from 1 to 20, so that makes a total of 100 formulas. Each of the tools below produces the theoretically smallest Büchi Automaton, so we only measure the total time it takes to translate these 100 formulas. See `bench/ltlclasses/README` to reproduce.

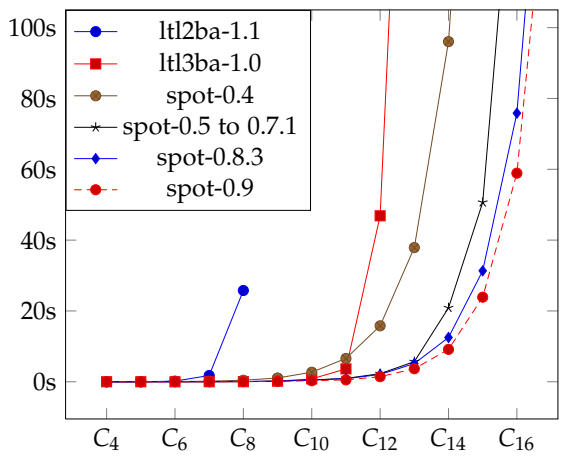
Spot 0.8.3	562 seconds
Spot 0.9	315 seconds
ltl3ba 1.0	78 seconds

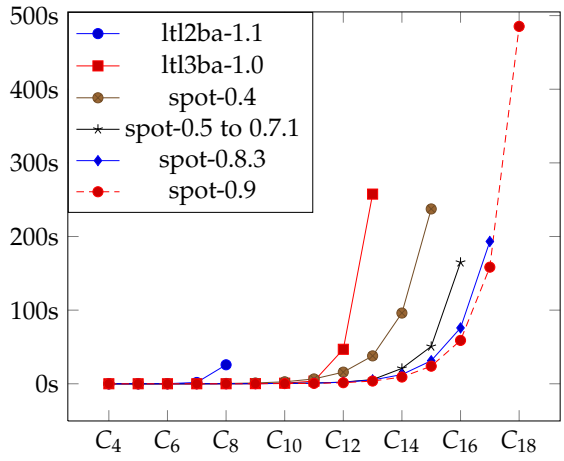
For the above translations Spot is configured with basic LTL rewritings enabled (-r1).
`ltl3ba` is run with its default options, except for the family of formulas of the form $f_n = \mathsf{F}(p_1 \wedge \mathsf{F}(p_2 \wedge \ldots \mathsf{F}(p_n))) \wedge \mathsf{F}(1_1 \wedge \mathsf{F}(q_2 \wedge \ldots \mathsf{F}(q_n)))$ where LTL simplification have to be disabled. (With LTL simplifications turned on `ltl3ba` would take one hour to translate f_{17}.)

Spot 0.9 actually spends all its time translating the class of formulas representing weak fairness constraints: $g_n = \bigwedge_{i=1}^{n} \mathsf{G}\,\mathsf{F}\,p_i$. The other four classes are translated instantaneously, or nearly so: the worst formula outside this class is f_{20} and it takes 2.2s to translate. Spot 0.9 needs 238s to translate g_{20}; comparatively, `ltl3ba`, which has a specific handling of subformulas that have the form of g_n, will translate g_{20} in only 42s!

Rozier's LTL Counters

This parameterized family of LTL formulas, C_n, describes circular automata with $n2^n$ states and as much transitions. These formulas are quite *heavy*; e.g. $C_3 = ((a \wedge (\mathsf{G}(a \rightarrow (\mathsf{X}(\neg a \wedge \mathsf{X}(\neg a \wedge \mathsf{X}\,a)))))) \wedge ((\neg b) \wedge \mathsf{X}(\neg b \wedge \mathsf{X}\,\neg b)) \wedge (\mathsf{G}((a \wedge \neg b) \rightarrow (\mathsf{X}((\mathsf{X}\,\mathsf{X}\,b) \wedge (((\neg a) \wedge (b \rightarrow \mathsf{X}\,\mathsf{X}\,\mathsf{X}\,b) \wedge ((\neg b) \rightarrow (\mathsf{X}\,\mathsf{X}\,\mathsf{X}\,\neg b)))\,\mathsf{U}\,a))))) \wedge (\mathsf{G}((a \wedge b) \rightarrow (\mathsf{X}((\mathsf{X}\,\mathsf{X}\,\neg b) \wedge ((b \wedge (\neg a) \wedge \mathsf{X}\,\mathsf{X}\,\mathsf{X}\,\neg b)\,\mathsf{U}(a \vee ((\neg a) \wedge (\neg b) \wedge (\mathsf{X}((\mathsf{X}\,\mathsf{X}\,b) \wedge (((\neg a) \wedge (b \rightarrow \mathsf{X}\,\mathsf{X}\,\mathsf{X}\,b) \wedge ((\neg b) \rightarrow \mathsf{X}\,\mathsf{X}\,\mathsf{X}\,\neg b))\,\mathsf{U}\,a))))))))))$. See `bench/ltlcounter/README` for references. The following plots (two zoom levels for the same data) show the time it takes to generate a TGBA for increasing n . A timeout was set at 10 minutes, so values above that are not shown.





The values for spot-0.6 and spot-0.7.1, not shown, are the same as those for spot-0.5. Spot is run without any pre- or post-processings: they are not needed to translate these formulas. `ltl2ba` is run with options `-p -l -c -U` disabling pre- and post- processings that would unfairly increase the runtime; similarly `ltl3ba` is run with options `-p -l -c -C -U`. Both tools have been patched to add the option `-U` so they exit immediately after the TGBA has been constructed. This way we measure the actual translation from LTL to TGBA without any extra cost of translating it to a Büchi automaton.

All experiments were ran under GNU/Linux on an Intel Core2 Q9550 running at 2.83GHz with 8GB of RAM.