LTL Translation Improvements in Spot

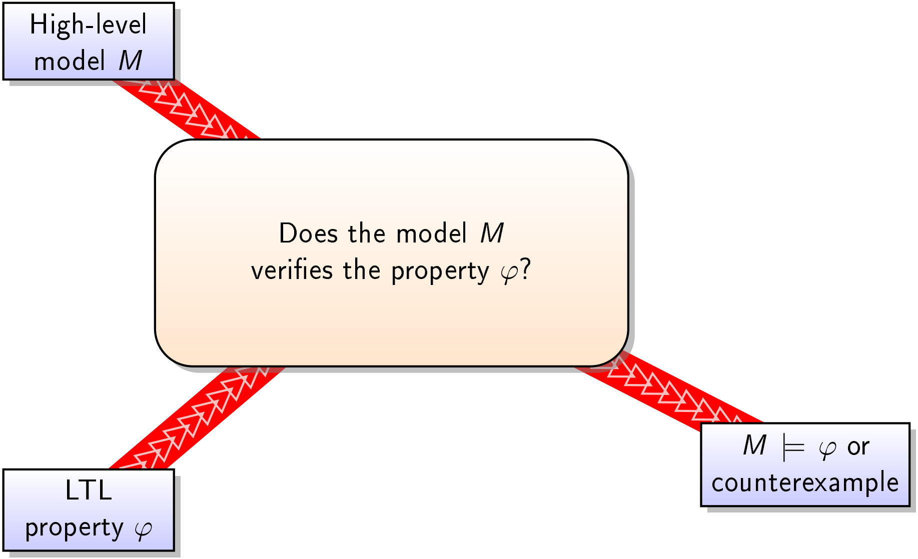
Alexandre Duret-Lutz

<adl@lrde.epita.fr>

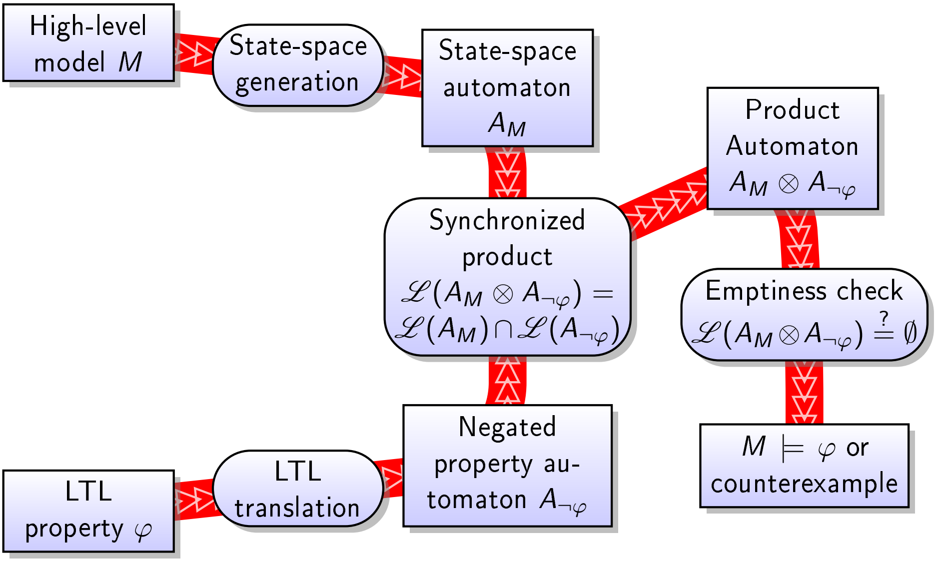
http://www.lrde.epita.fr/~adl/

VECoS'11 — 16 September 2011

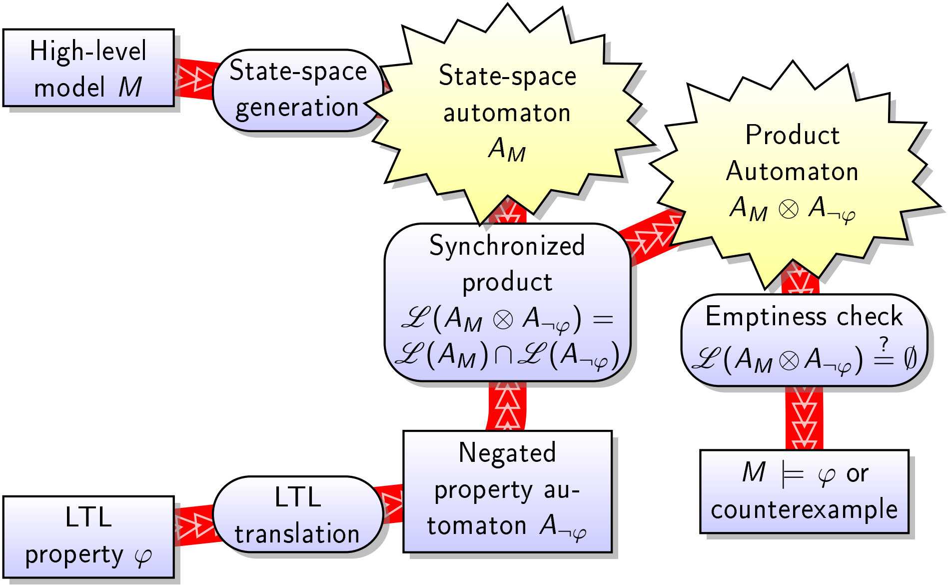
Context


High-level model M
Does the model M verifies the property φ?
LTL property φ
$M \models \varphi$ or counterexample

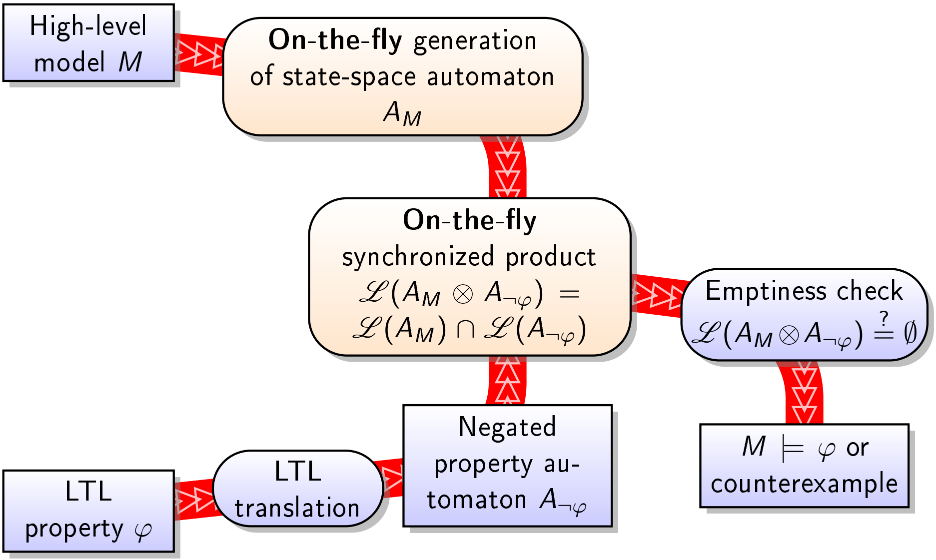
Automata-Theoretic LTL Model Checking



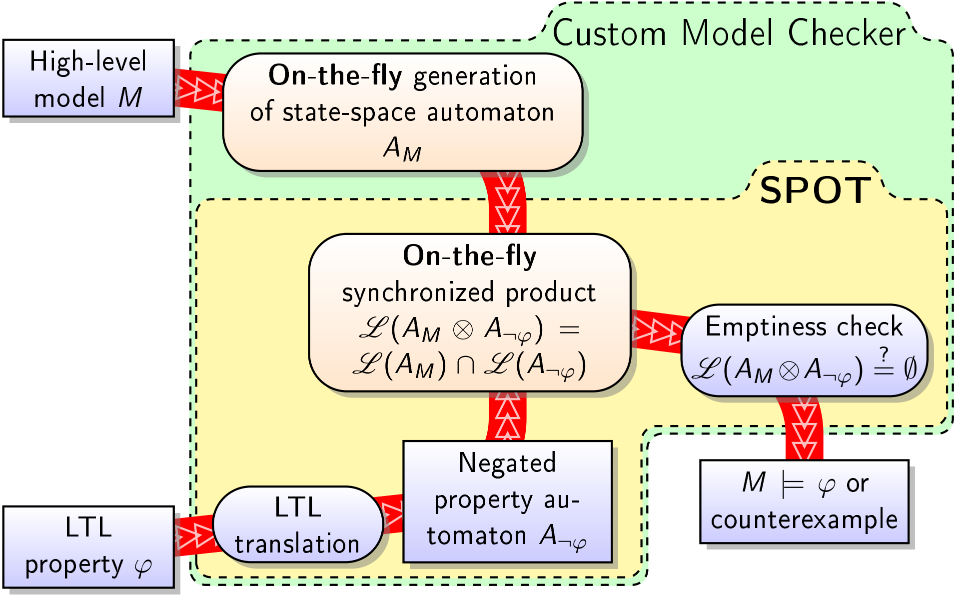
Automata-Theoretic LTL Model Checking



Automata-Theoretic LTL Model Checking



Automata-Theoretic LTL Model Checking



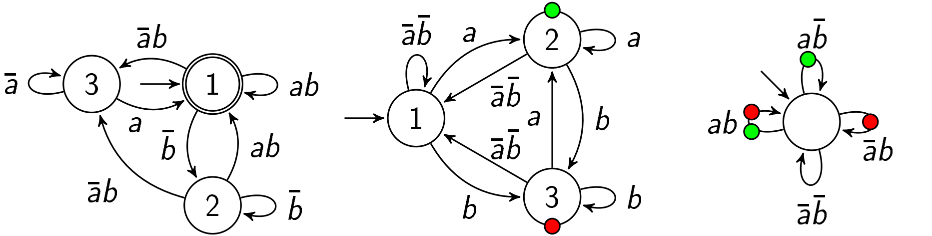
Let's Talk about the LTL Translation

- We translate LTL to generalized Büchi automata with transition-based acceptance.
- Most people prefer non-generalized Büchi automata with state acceptance.
- Our translation is competitive, thanks to:
  - A simple tableau construction that use BDD for simplifications
  - Two techniques to improve determinism
  - Several LTL rewriting rules that ease the translation (not shown)

Transition-based Generalized Acceptance Conditions

Different Kinds of Büchi Automata ($\mathbf{G}\,\mathbf{F}\,a \wedge \mathbf{G}\,\mathbf{F}\,b$)

Büchi Automaton	Generalized Büchi Automaton	Transition-based Generalized Büchi Automaton



- Same expressive power.
- Converting BA to GBA, or GBA to TGBA, is trivial.
- The opposite direction requires a “degeneralization”.
- (T)GBA occur naturally when translating LTL.

Translation of Litterature Formulas

Cumulated sizes of automata for 188 formulas from the litterature → $\Sigma|A_{\neg\varphi}|$

Products with a random state-space of 200 states → $\Sigma|A_M \otimes A_{\neg\varphi}|$

		$\Sigma\lvert A_{\neg\varphi}\rvert$ st.	$\Sigma\lvert A_{\neg\varphi}\rvert$ tr.	$\Sigma\lvert A_M \otimes A_{\neg\varphi}\rvert$ st.	$\Sigma\lvert A_M \otimes A_{\neg\varphi}\rvert$ tr.
BA	Spin 6.1.0 (☠×9)	1 572	7 214	311 032	20 924 268
	LTL2BA 1.1	1 080	3 646	215 717	12 766 425
	Modella 1.5.9 (☢×1)	1 394	4 576	274 881	10 960 064
	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
TGBA	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

☠ = 15min timeout
☢ = bogus translation

Produce more **det**erministic aut. → Spot 0.7.1 det.
WDBA minimization when applicable → Spot 0.7.1 WDBA

Rozier & Vardi's Scalability Experiment (1/2)

- An LTL formula C_n that can be encoded by a $n2^n$-state automaton.

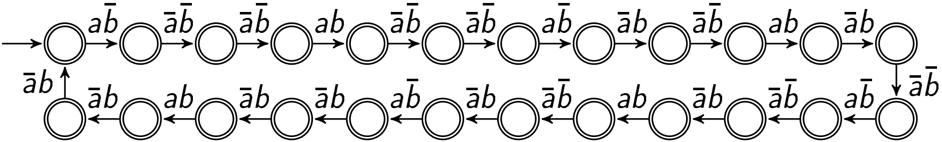
K. Y. Rozier and M. Y. Vardi. LTL satisfiability checking. In Proc. of SPIN'07, vol. 4595 of LNCS, pp. 149–167. Springer

Rozier & Vardi's Scalability Experiment (1/2)

- An LTL formula C_n that can be encoded by a $n2^n$-state automaton.
- E.g. $C_3 = ((a \land (\mathbf{G}(a \rightarrow (\mathbf{X}(\neg a \land \mathbf{X}(\neg a \land \mathbf{X}\, a)))))) \land ((\neg b) \land \mathbf{X}(\neg b \land \mathbf{X}\, \neg b)) \land (\mathbf{G}((a \land \neg b) \rightarrow (\mathbf{X}((\mathbf{X}\,\mathbf{X}\, b) \land (((\neg a) \land (b \rightarrow \mathbf{X}\,\mathbf{X}\,\mathbf{X}\, b) \land ((\neg b) \rightarrow (\mathbf{X}\,\mathbf{X}\,\mathbf{X}\, \neg b)))\,\mathbf{U}\, a))))) \land (\mathbf{G}((a \land b) \rightarrow (\mathbf{X}((\mathbf{X}\,\mathbf{X}\, \neg b) \land ((b \land (\neg a) \land \mathbf{X}\,\mathbf{X}\,\mathbf{X}\, \neg b)\,\mathbf{U}(a \lor ((\neg a) \land (\neg b) \land (\mathbf{X}((\mathbf{X}\,\mathbf{X}\, b) \land (((\neg a) \land (b \rightarrow \mathbf{X}\,\mathbf{X}\,\mathbf{X}\, b) \land ((\neg b) \rightarrow \mathbf{X}\,\mathbf{X}\,\mathbf{X}\, \neg b))\,\mathbf{U}\, a)))))))))))$

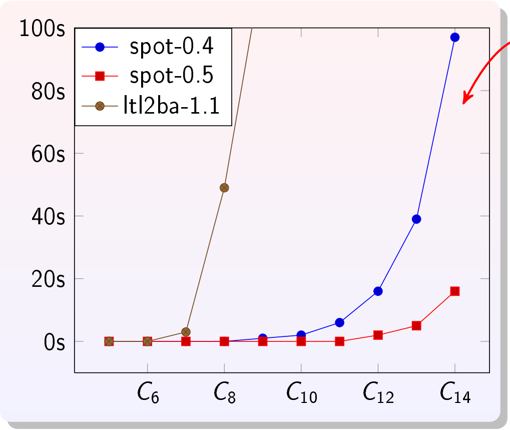
K. Y. Rozier and M. Y. Vardi. LTL satisfiability checking. In Proc. of SPIN'07, vol. 4595 of LNCS, pp. 149–167. Springer

- An LTL formula C_n that can be encoded by a $n2^n$-state automaton.
- E.g. $C_3 = ((a \land (\mathbf{G}(a \rightarrow (\mathbf{X}(\neg a \land \mathbf{X}(\neg a \land \mathbf{X}\, a)))))) \land ((\neg b) \land \mathbf{X}(\neg b \land \mathbf{X}\,\neg b)) \land (\mathbf{G}((a \land \neg b) \rightarrow (\mathbf{X}((\mathbf{X}\,\mathbf{X}\, b) \land (((\neg a) \land (b \rightarrow \mathbf{X}\,\mathbf{X}\,\mathbf{X}\, b) \land ((\neg b) \rightarrow (\mathbf{X}\,\mathbf{X}\,\mathbf{X}\,\neg b)))\,\mathbf{U}\, a))))) \land (\mathbf{G}((a \land b) \rightarrow (\mathbf{X}((\mathbf{X}\,\mathbf{X}\,\neg b) \land ((b \land (\neg a) \land \mathbf{X}\,\mathbf{X}\,\mathbf{X}\,\neg b)\,\mathbf{U}(a \lor ((\neg a) \land (\neg b) \land (\mathbf{X}((\mathbf{X}\,\mathbf{X}\, b) \land (((\neg a) \land (b \rightarrow \mathbf{X}\,\mathbf{X}\,\mathbf{X}\, b) \land ((\neg b) \rightarrow \mathbf{X}\,\mathbf{X}\,\mathbf{X}\,\neg b))\,\mathbf{U}\, a)))))))))))$



K. Y. Rozier and M. Y. Vardi. LTL satisfiability checking. In Proc. of SPIN'07, vol. 4595 of LNCS, pp. 149–167. Springer

Rozier & Vardi's Scalability Experiment (2/2)



Time to translate C_n into BA

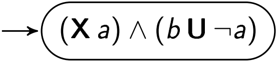
Other explicit translators are off the chart:

- Modella 1.5.9 took nearly 6 minutes to compute C_4 and ran out of memory on C_5.

- Spin 6.1.0 took more than 11 hours to translate C_1 into a 33-state automaton with 447 transitions (instead of 2 states and 2 transitions). Many transitions having unsatisfiable guards such as "`((!b) && (a) && (b))`".

All experiments done on an Intel Core2 Q9550 @2.83GHz with 8GB of RAM.

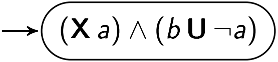
Tableau Construction for LTL



1. Label the initial state by the formula to translate

J.-M. Couvreur. On-the-fly verification of linear temporal logic. In Proc. of FM'99, vol. 1708 of LNCS, pp. 253–271. Springer

Tableau Construction for LTL



1. Label the initial state by the formula to translate
2. Rewrite each state label φ as

$$\bigvee_i \beta_i \land \mathbf{X}\,\psi_i$$

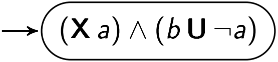
β_i: Boolean formula

ψ_i: LTL formula

Since $f\,\mathbf{U}\,g = g \lor (f \land \mathbf{X}(f\,\mathbf{U}\,g))$ we have:
$(\mathbf{X}\,a) \land (b\,\mathbf{U}\,\neg a) = (\neg a \land \mathbf{X}\,a) \lor (b \land (\mathbf{X}\,a) \land \mathbf{X}(b\,\mathbf{U}\,\neg a))$

J.-M. Couvreur. On-the-fly verification of linear temporal logic. In Proc. of FM'99, vol. 1708 of LNCS, pp. 253–271. Springer

Tableau Construction for LTL



1. Label the initial state by the formula to translate
2. Rewrite each state label φ as

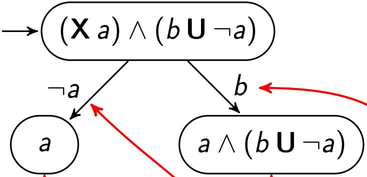
$$\bigvee_i \beta_i \wedge \mathbf{X}\,\psi_i$$

Since $f\,\mathbf{U}\,g = g \vee (f \wedge \mathbf{X}(f\,\mathbf{U}\,g))$ we have:

$(\mathbf{X}\,a) \wedge (b\,\mathbf{U}\,\neg a) = (\neg a \wedge \mathbf{X}\,\mathrm{a}) \vee (b \wedge \mathbf{X}(a \wedge (b\,\mathbf{U}\,\neg a)))$

J.-M. Couvreur. On-the-fly verification of linear temporal logic. In Proc. of FM'99, vol. 1708 of LNCS, pp. 253–271. Springer

Tableau Construction for LTL



1. Label the initial state by the formula to translate
2. Rewrite each state label φ as

$$\bigvee_i \beta_i \wedge \mathbf{X}\, \psi_i$$

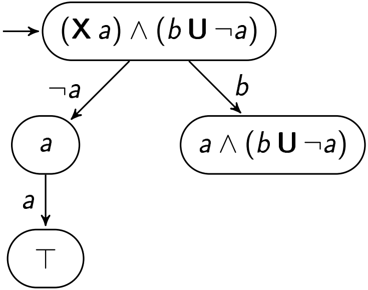
Then connect φ to each ψ_i using β_i as label

Since $f\, \mathbf{U}\, g = g \vee (f \wedge \mathbf{X}(f\, \mathbf{U}\, g))$ we have:

$(\mathbf{X}\, a) \wedge (b\, \mathbf{U} \neg a) = (\neg a \wedge \mathbf{X}\, a) \vee (b \wedge \mathbf{X}(a \wedge (b\, \mathbf{U} \neg a)))$

J.-M. Couvreur. On-the-fly verification of linear temporal logic. In Proc. of FM'99, vol. 1708 of LNCS, pp. 253–271. Springer

Tableau Construction for LTL



1. Label the initial state by the formula to translate
2. Rewrite each state label φ as

$$\bigvee_i \beta_i \wedge \mathbf{X}\,\psi_i$$

Then connect φ to each ψ_i using β_i as label

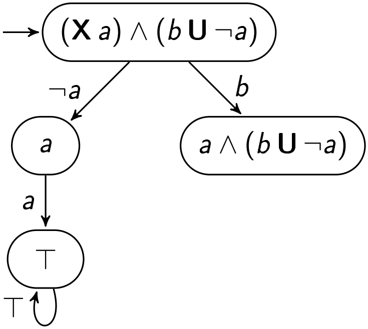
Since $f\,\mathbf{U}\,g = g \vee (f \wedge \mathbf{X}(f\,\mathbf{U}\,g))$ we have:

$(\mathbf{X}\,a) \wedge (b\,\mathbf{U}\,\neg a) = (\neg a \wedge \mathbf{X}\,a) \vee (b \wedge \mathbf{X}(a \wedge (b\,\mathbf{U}\,\neg a)))$

$a = a \wedge \mathbf{X}\,\top$

J.-M. Couvreur. On-the-fly verification of linear temporal logic. In Proc. of FM'99, vol. 1708 of LNCS, pp. 253–271. Springer

Tableau Construction for LTL



1. Label the initial state by the formula to translate
2. Rewrite each state label φ as

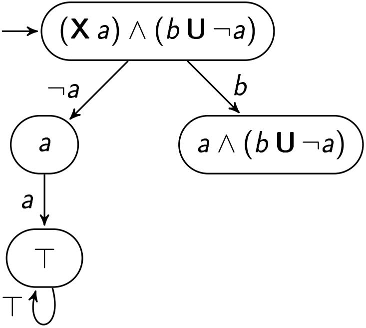
$$\bigvee_i \beta_i \wedge \mathbf{X}\, \psi_i$$

Then connect φ to each ψ_i using β_i as label

Since $f\, \mathbf{U}\, g = g \vee (f \wedge \mathbf{X}(f\, \mathbf{U}\, g))$ we have:
$(\mathbf{X}\, a) \wedge (b\, \mathbf{U} \neg a) = (\neg a \wedge \mathbf{X}\, a) \vee (b \wedge \mathbf{X}(a \wedge (b\, \mathbf{U} \neg a)))$
$a = a \wedge \mathbf{X} \top$; $\top = \top \wedge \mathbf{X} \top$

J.-M. Couvreur. On-the-fly verification of linear temporal logic. In Proc. of FM'99, vol. 1708 of LNCS, pp. 253–271. Springer

Tableau Construction for LTL



1. Label the initial state by the formula to translate
2. Rewrite each state label φ as

   $$\bigvee_i \; \beta_i \land \mathbf{X}\, \psi_i$$

   Then connect φ to each ψ_i using β_i as label

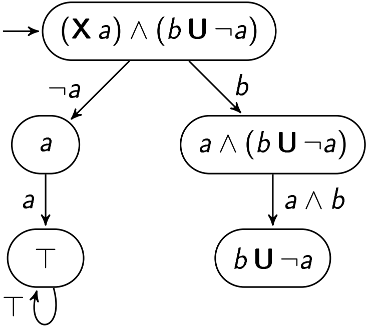
Since $f\, \mathbf{U}\, g = g \lor (f \land \mathbf{X}(f\, \mathbf{U}\, g))$ we have:

$(\mathbf{X}\, a) \land (b\, \mathbf{U} \neg a) = (\neg a \land \mathbf{X}\, a) \lor (b \land \mathbf{X}(a \land (b\, \mathbf{U} \neg a)))$

$a = a \land \mathbf{X} \top$; $\top = \top \land \mathbf{X} \top$; $a \land (b\, \mathbf{U} \neg a) = a \land (\neg a \lor (b \land \mathbf{X}(b\, \mathbf{U} \neg a)))$

J.-M. Couvreur. On-the-fly verification of linear temporal logic. In Proc. of FM'99, vol. 1708 of LNCS, pp. 253–271. Springer

Tableau Construction for LTL



1. Label the initial state by the formula to translate
2. Rewrite each state label φ as

$$\bigvee_i \beta_i \land \mathbf{X}\,\psi_i$$

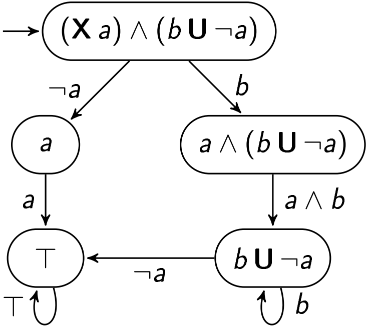
Then connect φ to each ψ_i using β_i as label

Since $f\,\mathbf{U}\,g = g \lor (f \land \mathbf{X}(f\,\mathbf{U}\,g))$ we have:

$(\mathbf{X}\,a) \land (b\,\mathbf{U}\,\neg a) = (\neg a \land \mathbf{X}\,\mathrm{a}) \lor (b \land \mathbf{X}(a \land (b\,\mathbf{U}\,\neg a)))$

$a = a \land \mathbf{X}\,\top$; $\top = \top \land \mathbf{X}\,\top$; $a \land (b\,\mathbf{U}\,\neg a) = a \land b \land \mathbf{X}\,(b\,\mathbf{U}\,\neg a)$

J.-M. Couvreur. On-the-fly verification of linear temporal logic. In Proc. of FM'99, vol. 1708 of LNCS, pp. 253–271. Springer



1. Label the initial state by the formula to translate
2. Rewrite each state label φ as

$$\bigvee_i \quad \beta_i \wedge \mathbf{X}\, \psi_i$$

Then connect φ to each ψ_i using β_i as label

Since $f\, \mathbf{U}\, g = g \vee (f \wedge \mathbf{X}(f\, \mathbf{U}\, g))$ we have:

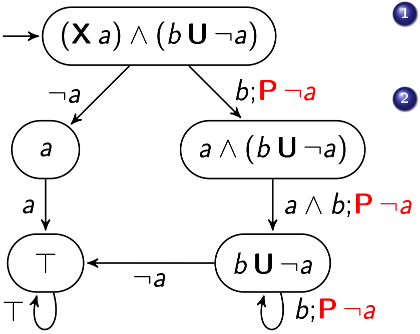
$(\mathbf{X}\, a) \wedge (b\, \mathbf{U} \neg a) = (\neg a \wedge \mathbf{X}\, a) \vee (b \wedge \mathbf{X}(a \wedge (b\, \mathbf{U} \neg a)))$

$a = a \wedge \mathbf{X}\, \top$; $\top = \top \wedge \mathbf{X}\, \top$; $a \wedge (b\, \mathbf{U} \neg a) = a \wedge b \wedge \mathbf{X}\, (b\, \mathbf{U} \neg a)$

$b\, \mathbf{U} \neg a = (a \wedge \mathbf{X}\, \top) \vee (b \wedge \mathbf{X}\, (b\, \mathbf{U} \neg a))$

J.-M. Couvreur. On-the-fly verification of linear temporal logic. In Proc. of FM'99, vol. 1708 of LNCS, pp. 253–271. Springer

Tableau Construction for LTL



1. Label the initial state by the formula to translate
2. Rewrite each state label φ as

$$\bigvee_i \left(\beta_i \land \mathbf{X}\, \psi_i \land \bigwedge_j \mathbf{P}\, \gamma_{ij} \right)$$

Then connect φ to each ψ_i using β_i as label and $\{\mathbf{P}\, \gamma_{ij}\}_j$ as promises

Since $f\, \mathbf{U}\, g = g \lor (f \land \mathbf{X}(f\, \mathbf{U}\, g) \land \mathbf{P}\, g)$ we have:

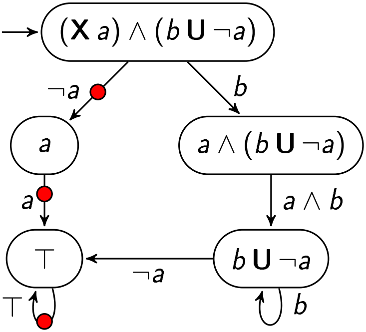
$(\mathbf{X}\, a) \land (b\, \mathbf{U} \neg a) = (\neg a \land \mathbf{X}\, a) \lor (b \land \mathbf{X}(a \land (b\, \mathbf{U} \neg a)) \land \mathbf{P}\, \neg a)$

$a = a \land \mathbf{X}\, \top$; $\top = \top \land \mathbf{X}\, \top$; $a \land (b\, \mathbf{U} \neg a) = a \land b \land \mathbf{X}\, (b\, \mathbf{U} \neg a) \land \mathbf{P}\, \neg a$

$b\, \mathbf{U} \neg a = (a \land \mathbf{X}\, \top) \lor (b \land \mathbf{X}\, (b\, \mathbf{U} \neg a) \land \mathbf{P}\, \neg a)$

J.-M. Couvreur. On-the-fly verification of linear temporal logic. In Proc. of FM'99, vol. 1708 of LNCS, pp. 253–271. Springer

Tableau Construction for LTL



1. Label the initial state by the formula to translate
2. Rewrite each state label φ as
   $$\bigvee_i \left(\beta_i \wedge \mathbf{X}\, \psi_i \wedge \bigwedge_j \mathbf{P}\, \gamma_{ij} \right)$$
   Then connect φ to each ψ_i using β_i as label and $\{\mathbf{P}\,\gamma_{ij}\}_j$ as promises
3. Create Büchi acceptance sets complementing each promise

Since $f \mathbin{\mathbf{U}} g = g \vee (f \wedge \mathbf{X}(f \mathbin{\mathbf{U}} g) \wedge \mathbf{P}\, g)$ we have:

$(\mathbf{X}\, a) \wedge (b \mathbin{\mathbf{U}} \neg a) = (\neg a \wedge \mathbf{X}\, a) \vee (b \wedge \mathbf{X}(a \wedge (b \mathbin{\mathbf{U}} \neg a)) \wedge \mathbf{P}\, \neg a)$

$a = a \wedge \mathbf{X}\, \top$; $\top = \top \wedge \mathbf{X}\, \top$; $a \wedge (b \mathbin{\mathbf{U}} \neg a) = a \wedge b \wedge \mathbf{X}\, (b \mathbin{\mathbf{U}} \neg a) \wedge \mathbf{P}\, \neg a$

$b \mathbin{\mathbf{U}} \neg a = (a \wedge \mathbf{X}\, \top) \vee (b \wedge \mathbf{X}\, (b \mathbin{\mathbf{U}} \neg a) \wedge \mathbf{P}\, \neg a)$

J.-M. Couvreur. On-the-fly verification of linear temporal logic. In Proc. of FM'99, vol. 1708 of LNCS, pp. 253–271. Springer

Implementing the Translation with BDDs (1/2)

$$
\begin{aligned}
r(\top) &= \top \\
r(\bot) &= \bot \\
r(p) &= \mathrm{Var}[p] \\
r(\neg p) &= \neg\mathrm{Var}[p] \\
r(f \vee g) &= r(f) \vee r(g) \\
r(f \wedge g) &= r(f) \wedge r(g) \\
r(\neg(f \vee g)) &= r(\neg f) \wedge r(\neg g) \\
r(\neg(f \wedge g)) &= r(\neg f) \vee r(\neg g) \\
r(\mathbf{X}\, f) &= \mathrm{Next}[f] \\
r(\neg\, \mathbf{X}\, f) &= \mathrm{Next}[\neg f] \\
r(f\, \mathbf{U}\, g) &= r(g) \vee (r(f) \wedge \mathrm{Next}[f\, \mathbf{U}\, g] \wedge \mathrm{P}[g]) \\
r(\neg(f\, \mathbf{U}\, g)) &= r(\neg g) \wedge (r(\neg f) \vee \mathrm{Next}[\neg(f\, \mathbf{U}\, g)])
\end{aligned}
$$

Implementing the Translation using BDDs (2/2)

$$
\begin{aligned}
& r((\mathbf{X}\, a) \land (b \,\mathbf{U}\, \neg a)) \\
= {} & r(\mathbf{X}\, a) \land r(b \,\mathbf{U}\, \neg a) \\
= {} & \mathrm{Next}[a] \land (r(\neg a) \lor (\mathrm{P}[\neg a] \land r(b) \land \mathrm{Next}[b \,\mathbf{U}\, \neg a])) \\
= {} & \mathrm{Next}[a] \land (\neg \mathrm{Var}[a] \lor (\mathrm{P}[\neg a] \land \mathrm{Var}[b] \land \mathrm{Next}[b \,\mathbf{U}\, \neg a]))
\end{aligned}
$$

Implementing the Translation using BDDs (2/2)

$$\begin{aligned}
& r((\mathbf{X}\, a) \land (b\, \mathbf{U} \neg a)) \\
= {} & r(\mathbf{X}\, a) \land r(b\, \mathbf{U} \neg a) \\
= {} & \mathrm{Next}[a] \land (r(\neg a) \lor (\mathrm{P}[\neg a] \land r(b) \land \mathrm{Next}[b\, \mathbf{U} \neg a])) \\
= {} & \mathrm{Next}[a] \land (\neg \mathrm{Var}[a] \lor (\mathrm{P}[\neg a] \land \mathrm{Var}[b] \land \mathrm{Next}[b\, \mathbf{U} \neg a]))
\end{aligned}$$

This BDD is then massaged into an irredundant sum of products:

$$= (\neg \mathrm{Var}[a] \land \mathrm{Next}[a]) \lor (\mathrm{Var}[b] \land \mathrm{Next}[a] \land \mathrm{Next}[b\, \mathbf{U} \neg a] \land \mathrm{P}[\neg a])$$

S. Minato. Fast generation of irredundant sum-of-products forms from binary decision diagrams. In Proc. of SASIMI'92, pp. 64–73

Implementing the Translation using BDDs (2/2)

$$
\begin{aligned}
& r((\mathbf{X}\, a) \land (b\, \mathbf{U}\, \neg a)) \\
&= r(\mathbf{X}\, a) \land r(b\, \mathbf{U}\, \neg a) \\
&= \mathrm{Next}[a] \land (r(\neg a) \lor (\mathrm{P}[\neg a] \land r(b) \land \mathrm{Next}[b\, \mathbf{U}\, \neg a])) \\
&= \mathrm{Next}[a] \land (\neg \mathrm{Var}[a] \lor (\mathrm{P}[\neg a] \land \mathrm{Var}[b] \land \mathrm{Next}[b\, \mathbf{U}\, \neg a]))
\end{aligned}
$$

This BDD is then massaged into an irredundant sum of products:

$$
= (\neg \mathrm{Var}[a] \land \mathrm{Next}[a]) \lor (\mathrm{Var}[b] \land \mathrm{Next}[a] \land \mathrm{Next}[b\, \mathbf{U}\, \neg a] \land \mathrm{P}[\neg a])
$$

S. Minato. Fast generation of irredundant sum-of-products forms from binary decision diagrams. In Proc. of SASIMI'92, pp. 64–73

BDD Gains (1/3): Automatic Simplifications

- Trivial simplifications of dead branches:
- Less trivial simplifications: e.g. $\neg((a \mathbf{U} b) \vee (b \mathbf{U} c))$.

BDD Gains (1/3): Automatic Simplifications

- Trivial simplifications of dead branches:
  A transition labeled by $a \wedge \neg a$ cannot exist
- Less trivial simplifications: e.g. $\neg((a \mathbf{U} b) \vee (b \mathbf{U} c))$.

BDD Gains (1/3): Automatic Simplifications

- Trivial simplifications of dead branches:
  A transition labeled by $a \wedge \neg a$ cannot exist
- Less trivial simplifications: e.g. $\neg((a \mathbf{U} b) \vee (b \mathbf{U} c))$.
  Tableau rules give four immediate successors:

$$
\begin{aligned}
& (\neg a) \wedge (\neg b) \wedge (\neg c) \\
\vee\ & (\neg a) \wedge (\neg b) \wedge (\neg c) \wedge \mathbf{X} \neg(b \mathbf{U} c) \\
\vee\ & (\neg b) \wedge (\neg c) \wedge (\mathbf{X} \neg(a \mathbf{U} b)) \\
\vee\ & (\neg b) \wedge (\neg c) \wedge (\mathbf{X} \neg(a \mathbf{U} b)) \wedge \mathbf{X} \neg(b \mathbf{U} c)
\end{aligned}
$$

  BDD rewritings give two successors:

$$
\begin{aligned}
& \neg \mathrm{Var}[b] \wedge (\neg \mathrm{Var}[a] \vee \mathrm{Next}[\neg(a \mathbf{U} b)]) \\
\wedge\ & \neg \mathrm{Var}[c] \wedge (\neg \mathrm{Var}[b] \vee \mathrm{Next}[\neg(b \mathbf{U} c)])
\end{aligned}
$$

BDD Gains (1/3): Automatic Simplifications

- Trivial simplifications of dead branches:
  A transition labeled by $a \land \neg a$ cannot exist
- Less trivial simplifications: e.g. $\neg((a \mathbf{U} b) \lor (b \mathbf{U} c))$.
  Tableau rules give four immediate successors:

$$\begin{aligned} & (\neg a) \land (\neg b) \land (\neg c) \\ \lor\ & (\neg a) \land (\neg b) \land (\neg c) \land \mathbf{X}\neg(b \mathbf{U} c) \\ \lor\ & (\neg b) \land (\neg c) \land (\mathbf{X}\neg(a \mathbf{U} b)) \\ \lor\ & (\neg b) \land (\neg c) \land (\mathbf{X}\neg(a \mathbf{U} b)) \land \mathbf{X}\neg(b \mathbf{U} c) \end{aligned}$$

  BDD rewritings give two successors:

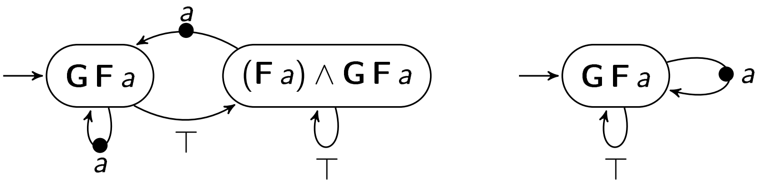
$$\begin{aligned} & \neg\mathrm{Var}[b] \land (\neg\mathrm{Var}[a] \lor \mathrm{Next}[\neg(a \mathbf{U} b)]) \\ \land\ & \neg\mathrm{Var}[c] \land (\neg\mathrm{Var}[b] \lor \mathrm{Next}[\neg(b \mathbf{U} c)]) \\ =\ & \neg\mathrm{Var}[b] \land \neg\mathrm{Var}[c] \land (\neg\mathrm{Var}[a] \lor \mathrm{Next}[\neg(a \mathbf{U} b)]) \end{aligned}$$

BDD Gains (2/3): Automatic Quotienting

BDD rewritings give us an equivalence relation between LTL formulas.

$$r(\mathbf{G}\,\mathbf{F}\,a) = ((\mathrm{Next}[\mathbf{F}\,a] \wedge \mathrm{P}[a]) \vee \mathrm{Var}[a]) \wedge \mathrm{Next}[\mathbf{G}\,\mathbf{F}\,a]$$

$$r(\mathbf{F}\,a \wedge \mathbf{G}\,\mathbf{F}\,a) = ((\mathrm{Next}[\mathbf{F}\,a] \wedge \mathrm{P}[a]) \vee \mathrm{Var}[a]) \wedge \mathrm{Next}[\mathbf{G}\,\mathbf{F}\,a]$$

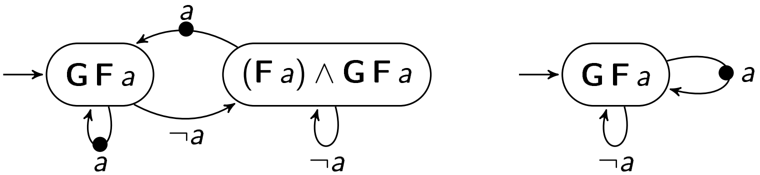


Could we produce something *more deterministic*?

BDD Gain (3/3): Improving Determinism

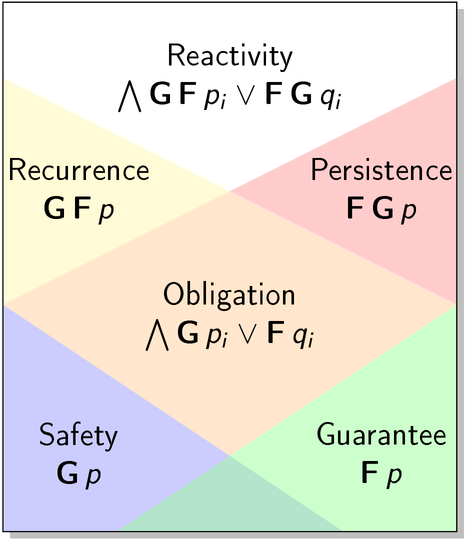
Let's improve the determinism of the previous construction by checking how different variable assignments influence destinations:

$$r(\mathbf{G}\,\mathbf{F}\,a) = ((\mathrm{Next}[\mathbf{F}\,a] \land \mathrm{P}[a]) \lor \mathrm{Var}[a]) \land \mathrm{Next}[\mathbf{G}\,\mathbf{F}\,a]$$
$$r(\mathbf{G}\,\mathbf{F}\,a) \land \mathrm{Var}[a] = \mathrm{Var}[a] \land \mathrm{Next}[\mathbf{G}\,\mathbf{F}\,a]$$
$$r(\mathbf{G}\,\mathbf{F}\,a) \land \neg\mathrm{Var}[a] = \neg\mathrm{Var}[a] \land \mathrm{Next}[\mathbf{F}\,a] \land \mathrm{P}[a] \land \mathrm{Next}[\mathbf{G}\,\mathbf{F}\,a]$$



Experiment on 188 LTL formulas from the literature applied to random graphs: 0.4% less states, **25% less transitions** in product.

Temporal Hierarchy

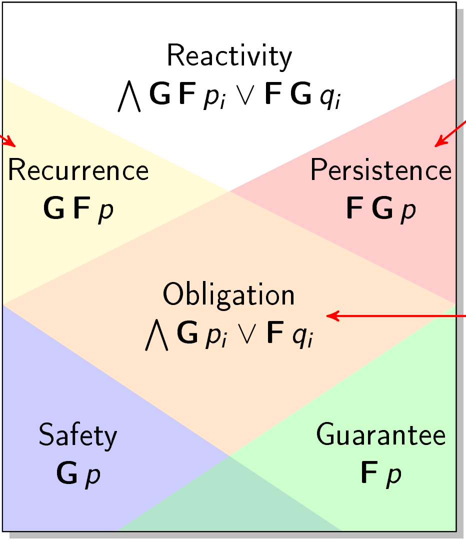


Z. Manna and A. Pnueli. A hierarchy of temporal properties. In Proc. of PODC'90, pp. 377–410. ACM

Temporal Hierarchy

Deterministic Büchi Automata

Weak Büchi Automata



Weak Det. Büchi Aut. (WDBA)

I. Černá and R. Pelánek. Relating hierarchy of temporal properties to model checking. In Proc. of MFCS'03, vol. 2747 of LNCS, pp. 318–327. Springer

Dealing with Obligation Properties

- 118 out of the 188 formulas (62%) are obligations properties
- WDBA can be minimized in a same way as DFAs (Löding; 2001)
- Dax et al. (2007) show how to minimize any automaton that can be expressed as a WDBA. In Spot this algorithm takes a TGBA and outputs a WDBA when the minimization is applicable
- Although converting to a minimal WDBA incurs a determinization, the number of states seldom increases

C. Löding. Efficient minimization of deterministic weak ω-automata. Information Processing Letters, 79(3):105–109, 2001

C. Dax, J. Eisinger, and F. Klaedtke. Mechanizing the powerset construction for restricted classes of ω-automata. In Proc. of ATVA'07, vol. 4762 of LNCS. Springer

Translation of Litterature Formulas

Cumulated sizes of automata for 188 formulas from the litterature → $\Sigma|A_{\neg\varphi}|$

Products with a random state-space of 200 states → $\Sigma|A_M \otimes A_{\neg\varphi}|$

		$\Sigma\lvert A_{\neg\varphi}\rvert$		$\Sigma\lvert A_M \otimes A_{\neg\varphi}\rvert$	
		st.	tr.	st.	tr.
BA	Spin 6.1.0 (☠×9)	1 572	7 214	311 032	20 924 268
	LTL2BA 1.1	1 080	3 646	215 717	12 766 425
	Modella 1.5.9 (☢×1)	1 394	4 576	274 881	10 960 064
	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
TGBA	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

☠ = 15min timeout

☢ = bogus translation

Produce more **det**erministic aut. (→ Spot 0.7.1 det.)

WDBA minimization when applicable (→ Spot 0.7.1 WDBA)

Conclusion

LTL translation

- The core translation is efficient and simple to implement
- Four tricks:
  - Supporting the weak until operator (not shown).
  - Choosing appropriate rewriting rules (not shown).
  - Using BDD is important (for speed **and** determinism)
  - Buiding WDBA when possible is often a good idea
- Try it on-line: `http://spot.lip6.fr/ltl2tgba.html`

What's Cooking

- Support for PSL (Property Specification Language)
  Essentially: LTL + rational operators.
  `{ 1[*]; init; busy[=2]; done } []-> (!init U bell)`
- Simulation-based reductions (for TGBA)