LTL to Büchi Automata Translation: Fast and More Deterministic

TACAS 2012

Reporter: Weizhi Feng

6.9

LTL2BA问题

- 本文主要关注的问题是从一个LTL公式φ生成Büchi 自动机A，使得$L(A) = \{u \in \Sigma^\omega \mid u \vDash \varphi\}$.
- 意义：是model checking等领域的重要问题，如要证明程序是否满足某些性质：
  - 将LTL公式取补转为BA，与程序自动机做乘积，检查乘积自动机的语言是否为空.

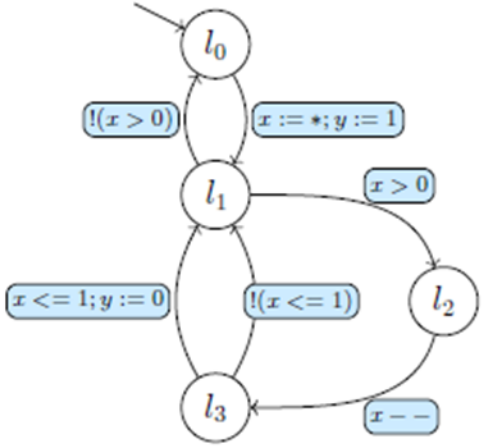


Figure: CFA of P

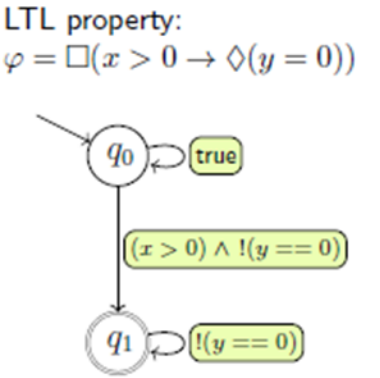
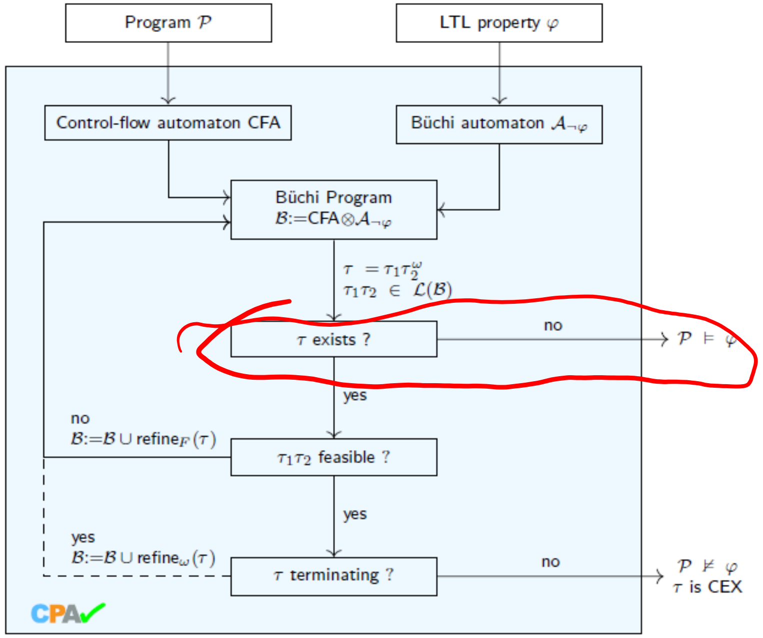


Figure: Büchi Automaton $\mathcal{A}_{\neg\varphi}$

LTL Software Model Checking



主要贡献

CAV 01

- 对经典算法（LTL2BA）进行改进（LTL3BA），更快生成更小、更确定性的自动机。
- 算法框架:
  - 将给定LTL转为一个very weak alternating automaton （VWAA）with a co- Büchi 接受条件;
  - 将这个VWAA转为一个transition-based generalized Büchi automaton (TGBA);
  - 将TGBA转为Büchi automaton (BA).

- 在上面框架的基础上，进行改进与优化，使每一步得到状态数更少的自动机。

Preliminaries

- Linear Temporal Logic (LTL):
- Syntax

$$\varphi \ ::= \ tt \mid a \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \mathsf{X}\varphi \mid \varphi \,\mathsf{U}\, \varphi,$$

- Semantics:

$$\begin{array}{lll}
u \models tt & & \\
u \models a & \text{iff} & a \in u(0) \\
u \models \neg\varphi & \text{iff} & u \not\models \varphi \\
u \models \varphi_1 \vee \varphi_2 & \text{iff} & u \models \varphi_1 \text{ or } u \models \varphi_2 \\
u \models \varphi_1 \wedge \varphi_2 & \text{iff} & u \models \varphi_1 \text{ and } u \models \varphi_2 \\
u \models \mathsf{X}\varphi & \text{iff} & u_1 \models \varphi \\
u \models \varphi_1 \,\mathsf{U}\, \varphi_2 & \text{iff} & \exists i \geq 0\,.\,(\, u_i \models \varphi_2 \text{ and } \forall\, 0 \leq j < i\,.\, u_j \models \varphi_1\,)
\end{array}$$

- 扩展的temporal operators：

  - $\mathsf{F}\varphi$ called *eventually* and equivalent to $tt\,\mathsf{U}\,\varphi$,
  - $\mathsf{G}\varphi$ called *always* and equivalent to $\neg\mathsf{F}\neg\varphi$, and
  - $\varphi\,\mathsf{R}\,\psi$ called *release* and equivalent to $\neg(\neg\varphi\,\mathsf{U}\,\neg\psi)$.

$G\varphi = \neg tt\ R\ \varphi$

U

R

F

G

Preliminaries

Büchi Automata (BA) A BA is a tuple $\mathcal{B} = (Q, \Sigma, \delta, I, F)$, where

- Q is a finite set of *states*,
- Σ is a finite *alphabet*, 2^{AP}
- $\delta : Q \to 2^{\Sigma \times Q}$ is a total *transition function*,
- $I \subseteq Q$ is a set of *initial states*, and
- $F \subseteq Q$ is a set of *accepting states*.

$q_0 \xrightarrow{a} q_1$

co-Büchi

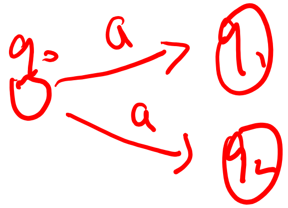
Preliminaries

- Very Weak Alternating co- Büchi Automata (VWAA):

$a : q$

  - Q is a finite set of *states*, and we let $Q' = 2^Q$,
  - Σ is a finite *alphabet*, and we let $\Sigma' = 2^\Sigma$,
  - $\delta : Q \to 2^{\Sigma' \times Q'}$ is a *transition function*,
  - $I \subseteq Q'$ is a set of *initial states*,
  - $F \subseteq Q$ is a set of *accepting states*, and
  - there exists a partial order on Q such that, for each state $q \in Q$, all the states occurring in $\delta(q)$ are lower or equal to q.

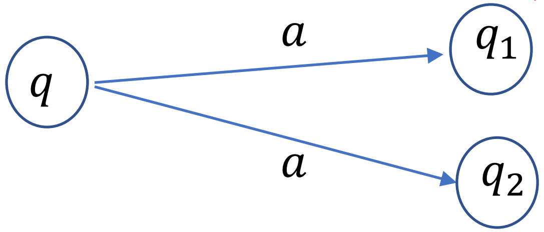
q, — $q_1 \wedge q_2$



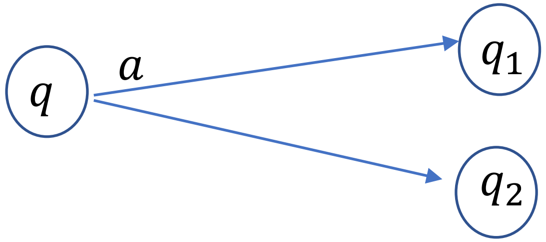
Preliminaries

- Very Weak Alternating co- Büchi Automata (VWAA):

- Alternating automaton的transition分为existential transition和universal transition:
- An existential transition $(q, a, q_1 \vee q_2)$



- An universal transition $(q, a, q_1 \wedge q_2)$



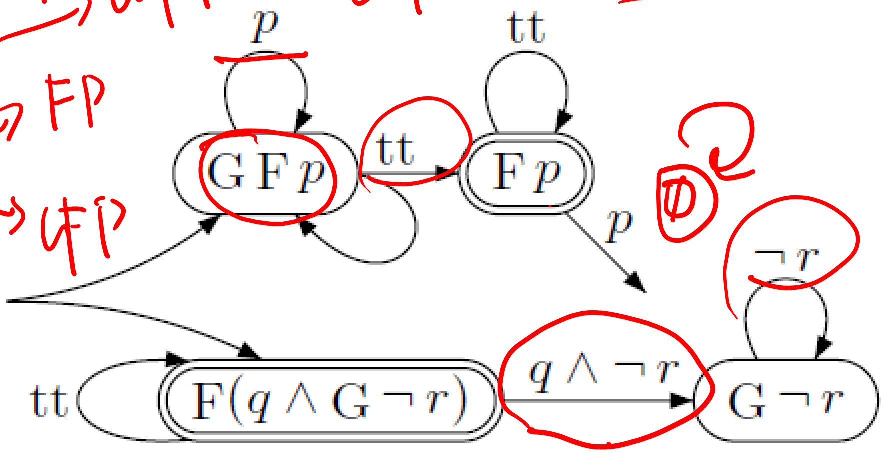
Preliminaries

$q \to \delta(q)$

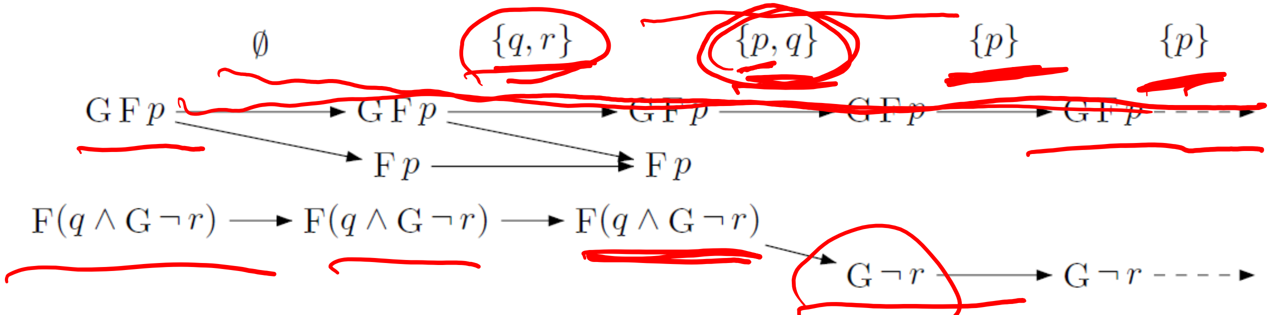
- Very Weak Alternating co- Büchi Automata (VWAA):
- Very weak: there exists a partial order on Q such that, for each state $q \in Q$, all the states occurring in $\delta(q)$ are lower or equal to q.
- Co- Büchi: a run σ is a accepting if each branch in σ contains only finitely many nodes labelled by accepting states.

2^{AP} $\quad \Sigma_p$ $\{p, q ..\}$

- Example: run DAG



Example 3. Here is an example of an accepting run of the automaton $\mathcal{A}_\theta$:



$\delta \quad p, \emptyset$

$\{p, q\}\ Fp \xrightarrow{p} \emptyset$

LTL to VWAA Translation

- Original translation: (
- 定义一些符号:

  – Let $\Sigma' = 2^{\Sigma}$, and let $Q' = 2^{Q}$. Given $J_1, J_2 \in 2^{\Sigma' \times Q'}$, we define

$$J_1 \otimes J_2 = \{(\alpha_1 \cap \alpha_2, O_1 \cup O_2) \mid (\alpha_1, O_1) \in J_1 \text{ and } (\alpha_2, O_2) \in J_2\}.$$

  – Let ψ be an LTL formula in positive normal form. We define $\overline{\psi}$ by:
    - $\overline{\psi} = \{\{\psi\}\}$ if ψ is a temporal formula,
    - $\overline{\psi_1 \wedge \psi_2} = \{O_1 \cup O_2 \mid O_1 \in \overline{\psi_1} \text{ and } O_2 \in \overline{\psi_2}\}$,
    - $\overline{\psi_1 \vee \psi_2} = \overline{\psi_1} \cup \overline{\psi_2}$.

$\overline{aFa \wedge}$

$= \{O_1 \cup O_2 |$

$\overline{aFa}$ $\{\{aFa\}\}$

LTL to VWAA Translation

- Original translation：
- 转换规则：一个VWAA: $A = (Q, \Sigma, \delta, I, F)$,
- Q是φ的temporal subformula的集合，$\Sigma = 2^{AP(\varphi)}$, $I = \bar{\varphi}$, F是φ的U子公式的集合.

$\{(\Sigma_a, \varphi)\}$

$$
\begin{aligned}
\delta(tt) &= \{(\Sigma, \emptyset)\} \\
\delta(p) &= \{(\Sigma_p, \emptyset)\} \text{ where } \Sigma_p = \{a \in \Sigma \mid p \in a\} \\
\delta(\neg p) &= \{(\Sigma_{\neg p}, \emptyset)\} \text{ where } \Sigma_{\neg p} = \Sigma \setminus \Sigma_p \\
\delta(\mathsf{X}\psi) &= \{(\Sigma, O) \mid O \in \overline{\psi}\} \\
\delta(\psi_1 \,\mathsf{U}\, \psi_2) &= \Delta(\psi_2) \cup (\Delta(\psi_1) \otimes \{(\Sigma, \{\psi_1 \,\mathsf{U}\, \psi_2\})\}) \\
\delta(\psi_1 \,\mathsf{R}\, \psi_2) &= \Delta(\psi_2) \otimes (\Delta(\psi_1) \cup \{(\Sigma, \{\psi_1 \,\mathsf{R}\, \psi_2\})\})
\end{aligned}
$$

$F\varphi$

$tt \, U \, \varphi$

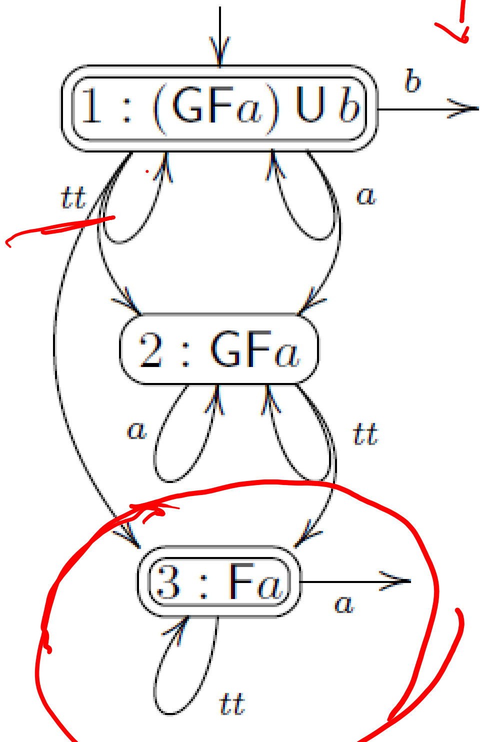
$a \, \varphi$

$\neg tt \, R \, \varphi$

$$
\begin{aligned}
\Delta(\psi) &= \delta(\psi) \text{ if } \psi \text{ is a temporal formula} \\
\Delta(\psi_1 \vee \psi_2) &= \Delta(\psi_1) \cup \Delta(\psi_2) \\
\Delta(\psi_1 \wedge \psi_2) &= \Delta(\psi_1) \otimes \Delta(\psi_2)
\end{aligned}
$$

LTL to VWAA Translation

- Original translation:
- Example: VWAA for $\varphi = Fa$, $\varphi = GFa$, $\varphi = (GFa)Ub$



LTL to VWAA Translation

- Improved translation：
- 定义alternating formulae，利用其性质来改进translation算法.

$$\mu ::= \mathsf{F}\varphi \mid \mu \vee \mu \mid \mu \wedge \mu \mid \mathsf{X}\mu \mid \varphi \,\mathsf{U}\, \mu \mid \mu \,\mathsf{R}\, \mu \mid \mathsf{G}\mu$$
$$\nu ::= \mathsf{G}\varphi \mid \nu \vee \nu \mid \nu \wedge \nu \mid \mathsf{X}\nu \mid \nu \,\mathsf{U}\, \nu \mid \varphi \,\mathsf{R}\, \nu \mid \mathsf{F}\nu$$
$$\xi ::= \mathsf{G}\mu \mid \mathsf{F}\nu \mid \xi \vee \xi \mid \xi \wedge \xi \mid \mathsf{X}\xi \mid \varphi \,\mathsf{U}\, \xi \mid \varphi \,\mathsf{R}\, \xi \mid \mathsf{F}\xi \mid \mathsf{G}\xi$$

- Lemma 2. each alternating formula defines a prefix-invariant language.

Every alternating formula ξ satisfies the following:

$$\forall w \in \Sigma^{\omega}, u \in \Sigma^{*} : uw \models \xi \iff w \models \xi$$

$\mathsf{F}\xi$

LTL to VWAA Translation

- Improved translation：
- 定义alternating formulae，利用其性质来改进translation算法.
- 原始的δ函数反映了如下恒等式：

$$\delta(\psi_1 \,\mathsf{U}\, \psi_2) = \Delta(\psi_2) \cup (\Delta(\psi_1) \otimes \{(\Sigma, \{\psi_1 \,\mathsf{U}\, \psi_2\})\})$$
$$\delta(\psi_1 \,\mathsf{R}\, \psi_2) = \Delta(\psi_2) \otimes (\Delta(\psi_1) \cup \{(\Sigma, \{\psi_1 \,\mathsf{R}\, \psi_2\})\})$$

$$\varphi_1 \,\mathsf{U}\, \varphi_2 \equiv \varphi_2 \vee (\varphi_1 \wedge \mathsf{X}(\varphi_1 \,\mathsf{U}\, \varphi_2))$$
$$\varphi_1 \,\mathsf{R}\, \varphi_2 \equiv \varphi_2 \wedge (\varphi_1 \vee \mathsf{X}(\varphi_1 \,\mathsf{R}\, \varphi_2))$$

- 由alternating公式的性质：$\varphi_1 \equiv X\varphi_1$，有：

suspend

$$\varphi_1 \,\mathsf{U}\, \varphi_2 \equiv \varphi_2 \vee (\mathsf{X}\varphi_1 \wedge \mathsf{X}(\varphi_1 \,\mathsf{U}\, \varphi_2))$$
$$\varphi_1 \,\mathsf{R}\, \varphi_2 \equiv \varphi_2 \wedge (\mathsf{X}\varphi_1 \vee \mathsf{X}(\varphi_1 \,\mathsf{R}\, \varphi_2))$$

LTL to VWAA Translation

- Improved translation：
- 定义alternating formulae，利用其性质来改进translation算法.
- 原始的δ函数反映了如下恒等式：
- 由alternating公式的性质：$\varphi_1 \equiv X\varphi_1$，有：

$$\varphi_1 \,\mathsf{U}\, \varphi_2 \equiv \varphi_2 \vee (\mathsf{X}\varphi_1 \wedge \mathsf{X}(\varphi_1 \,\mathsf{U}\, \varphi_2))$$
$$\varphi_1 \,\mathsf{R}\, \varphi_2 \equiv \varphi_2 \wedge (\mathsf{X}\varphi_1 \vee \mathsf{X}(\varphi_1 \,\mathsf{R}\, \varphi_2))$$

- 应用于转换中，可以公式被有效地suspended and checked one step later.

- 另外，令$X\varphi$类型的公式只生成一个后继.

LTL to VWAA Translation

- Improved translation：

$$
\begin{aligned}
\delta(tt) &= \{(\Sigma, \emptyset)\} \\
\delta(p) &= \{(\Sigma_p, \emptyset)\} \text{ where } \Sigma_p = \{a \in \Sigma \mid p \in a\} \\
\delta(\neg p) &= \{(\Sigma_{\neg p}, \emptyset)\} \text{ where } \Sigma_{\neg p} = \Sigma \backslash \Sigma_p \\
\delta(\mathsf{X}\psi) &= \{(\Sigma, \{\psi\})\} \\
\delta(\psi_1 \vee \psi_2) &= \Delta(\psi_1) \cup \Delta(\psi_2) \\
\delta(\psi_1 \wedge \psi_2) &= \Delta(\psi_1) \otimes \Delta(\psi_2) \\
\delta(\psi_1 \mathsf{U} \psi_2) &= \begin{cases} \Delta(\psi_2) \cup (\{(\Sigma, \{\psi_1\})\} \otimes \{(\Sigma, \{\psi_1 \mathsf{U} \psi_2\})\}) & \text{if } \psi_1 \text{ is alternating,} \\ \Delta(\psi_2) \cup (\Delta(\psi_1) \otimes \{(\Sigma, \{\psi_1 \mathsf{U} \psi_2\})\}) & \text{otherwise.} \end{cases} \\
\delta(\psi_1 \mathsf{R} \psi_2) &= \begin{cases} \Delta(\psi_2) \otimes (\{(\Sigma, \{\psi_1\}), (\Sigma, \{\psi_1 \mathsf{R} \psi_2\})\}) & \text{if } \psi_1 \text{ is alternating,} \\ \Delta(\psi_2) \otimes (\Delta(\psi_1) \cup \{(\Sigma, \{\psi_1 \mathsf{R} \psi_2\})\}) & \text{otherwise.} \end{cases}
\end{aligned}
$$

$\psi \longrightarrow \Sigma \longrightarrow \{\psi_1\}$

$$
\begin{aligned}
\Delta(\psi) &= \begin{cases} \{(\Sigma, \{\psi\})\} & \text{if } \psi \text{ is a temporal alternating formula,} \\ \delta(\psi) & \text{if } \psi \text{ is a temporal formula that is not alternating.} \end{cases} \\
\Delta(\psi_1 \vee \psi_2) &= \Delta(\psi_1) \cup \Delta(\psi_2) \\
\Delta(\psi_1 \wedge \psi_2) &= \Delta(\psi_1) \otimes \Delta(\psi_2)
\end{aligned}
$$

LTL to VWAA Translation

- Optimization of VWAA：
- If $t_1 = (q, \alpha_1, O_1)$, $t_2 = (q, \alpha_2, O_2)$ 满足$\alpha_2 \subseteq \alpha_1$, and $O_1 \subseteq O_2$， then t_2 is removed.

- Determinization:
- If $O_1 \subset O_2$，将t_2中的α_2 用$\alpha_2 \wedge \neg\alpha_1$替换.
- if $O_1 = O_2$， 两个迁移都替换为$(q, \alpha_1 \vee \alpha_2, O_1)$.

a $\quad a \wedge b$ $\quad \vee$

Σa $\quad t_1 \to \cancel{t_2}$

$\{a, \ldots\}$ $\quad \{a, b\}$ $\quad$ 公式

$\{a\}$

t_1 $\quad q \xrightarrow{a_1} O_1$, $q \xrightarrow{a_2} O_2$

$q \to O$

LTL to VWAA Translation

- Improved translation： examples

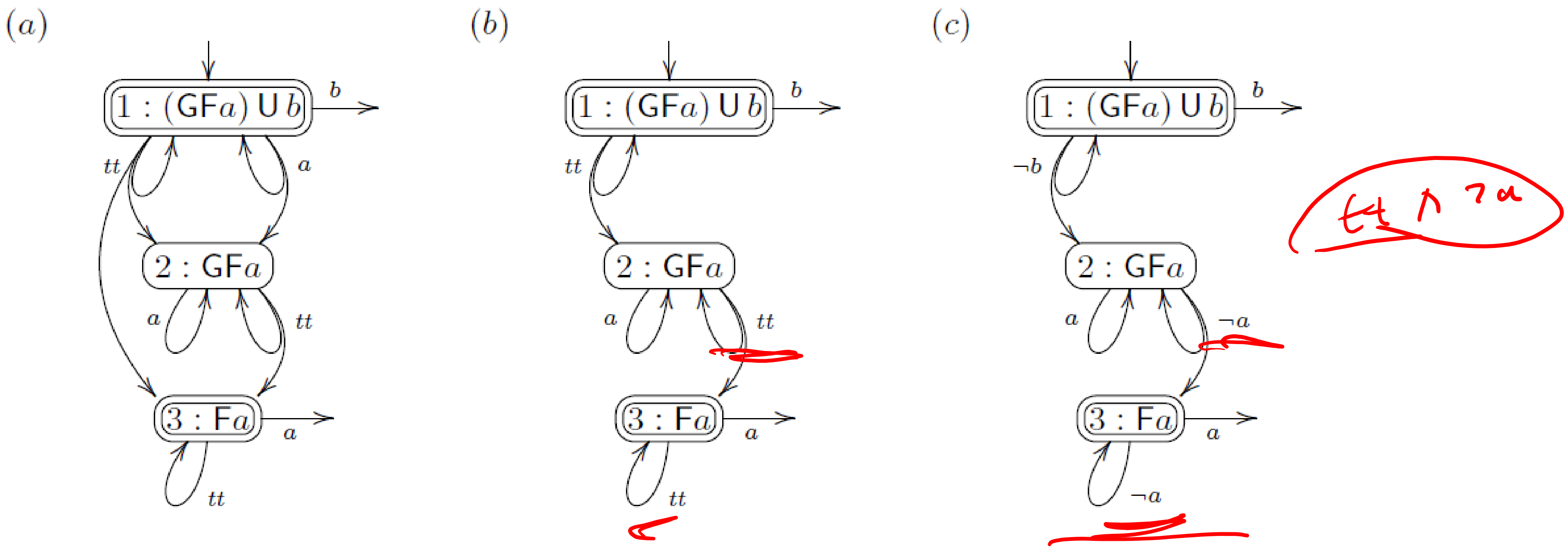


Fig. 1. VWAA for (GFa) U b generated by (a) the translation of [11], (b) our translation with suspension, and (c) our translation with suspension and further determinization.

LTL to VWAA Translation

- Improved translation： examples

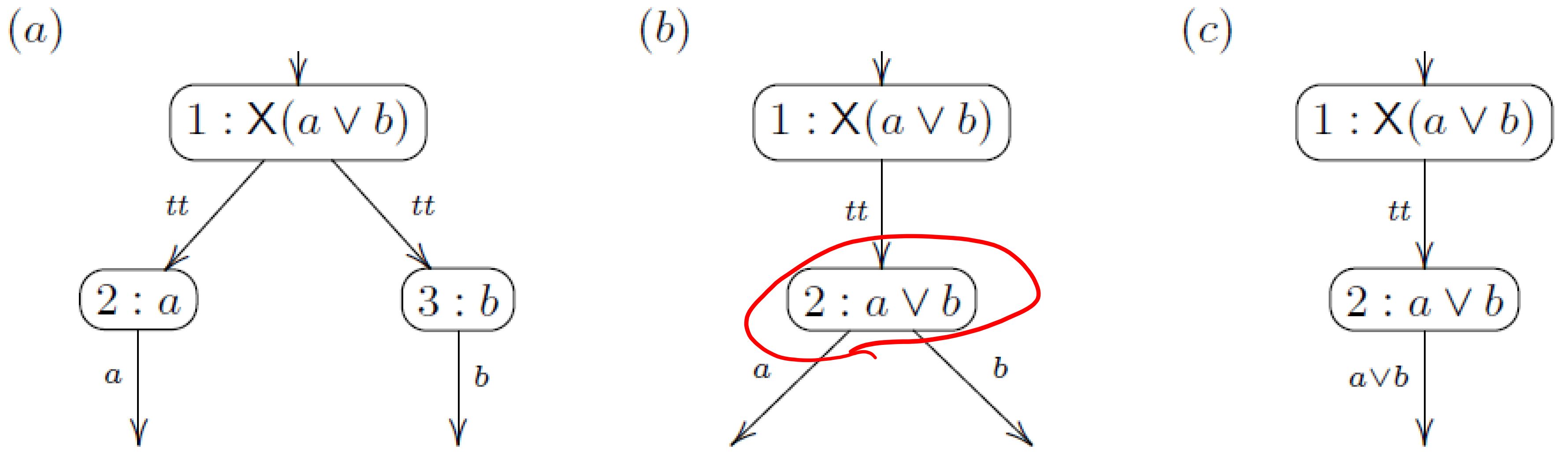


Fig. 2. VWAA for $\mathsf{X}(a \vee b)$ generated by (a) the translation of [11], (b) our translation with suspension, and (c) the translation with suspension and further determinization.

VWAA to TGBA Translation

- Transition Based Generalized Büchi Automata (TGBA)

Transition Based Generalized Büchi Automata (TGBA) A TGBA is a tuple $\mathcal{G} = (Q, \Sigma, \delta, I, \mathcal{F})$, where

- Q is a finite set of *states*,
- Σ is a finite *alphabet*, and we let $\Sigma' = 2^{\Sigma}$
- $\delta : Q \to 2^{\Sigma' \times Q}$ is a total *transition function*,
- $I \subseteq Q$ is a set of *initial states*, and
- $\mathcal{T} = \{T_1, T_2, \ldots, T_m\}$ where $T_j \subseteq Q \times \Sigma' \times Q$ are sets of *accepting transitions*.

- A run is accepting if for each $1 \leq j \leq m$ it uses infinitely many transitions from T_j.

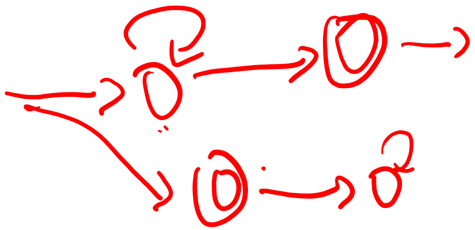
VWAA to TGBA Translation

- Original translation:

$\{1,2,3\}$ 1

Step 2. Let $\mathcal{A} = (Q, \Sigma, \delta, I, F)$ be a VWAA with co-Büchi acceptance conditions. We define the GBA $\mathcal{G}_{\mathcal{A}} = (Q', \Sigma, \delta', I, \mathcal{T})$ where:

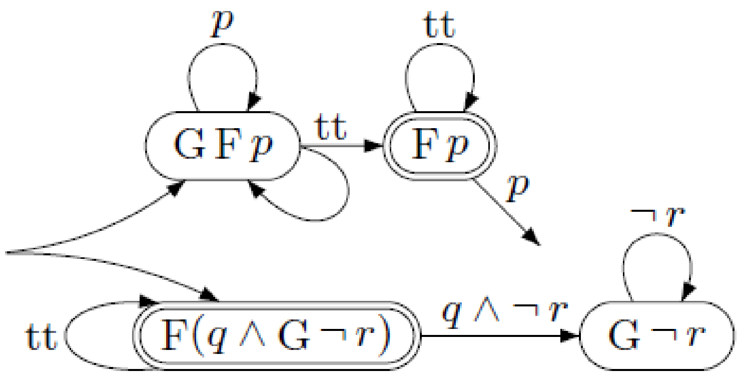
- $Q' = 2^Q$ is identified with conjunctions of states as explained in Definition $\boxed{3}$,
- $\delta''(q_1 \wedge \ldots \wedge q_n) = \bigotimes_{i=1}^{n} \delta(q_i)$, $\delta(q_i)$ $\otimes$
- δ' is the set of $\preccurlyeq$-minimal transitions of δ'' where the relation $\preccurlyeq$ is defined by $t' \preccurlyeq t$ if $t = (e, \alpha, e')$, $t' = (e, \alpha', e'')$, $\alpha \subseteq \alpha'$, $e'' \subseteq e'$, and $\forall T \in \mathcal{T}$, $t \in T \Rightarrow t' \in T$,
- $\mathcal{T} = \{T_f \mid f \in F\}$ where $T_f = \{(e, \alpha, e') \mid f \notin e' \text{ or } \exists(\beta, e'') \in \delta(f),\ \alpha \subseteq \beta \text{ and } f \notin e'' \subseteq e'\}$.

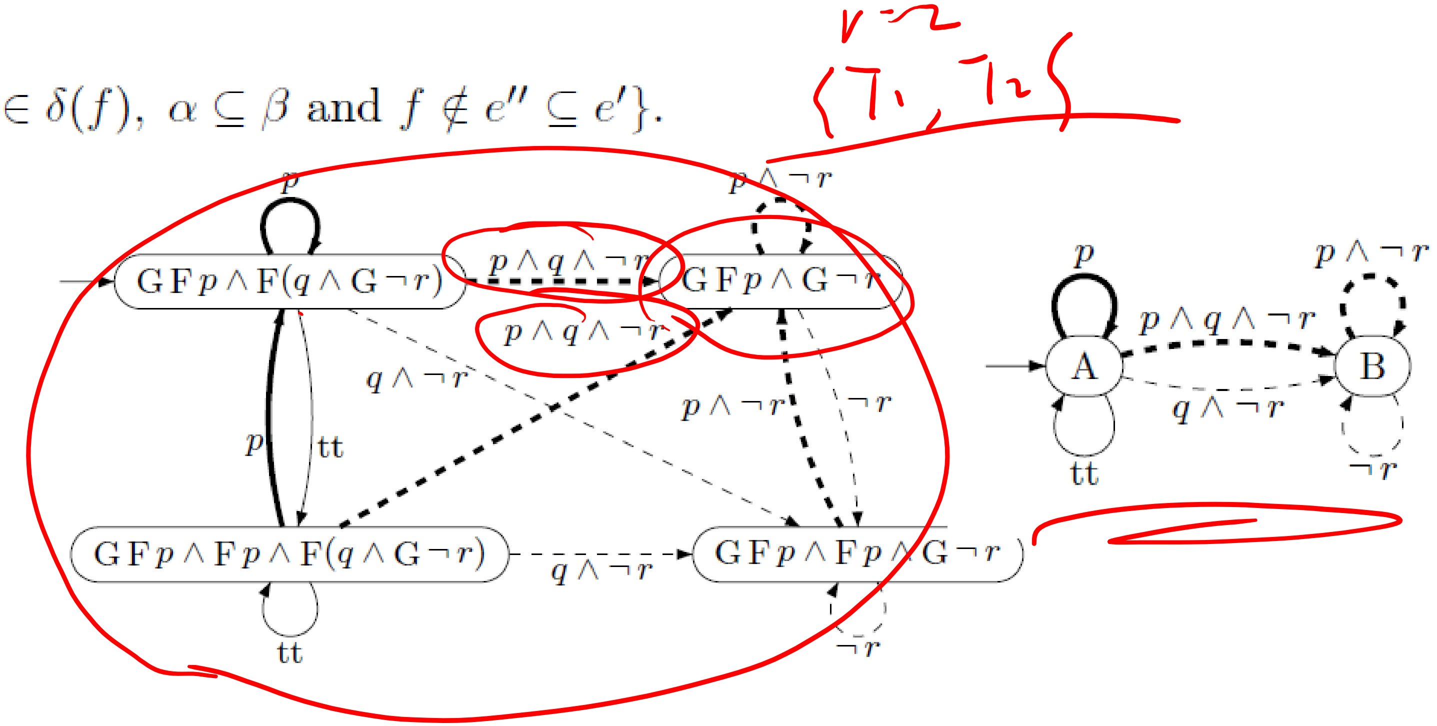


very weak

VWAA to TGBA Translation

- Original translation: example
  - $Q' = 2^Q$ is identified with conjunctions of states as explained in Definition $\boxed{3}$,
  - $\delta''(q_1 \wedge \ldots \wedge q_n) = \bigotimes_{i=1}^{n} \delta(q_i)$,
  - δ' is the set of $\preccurlyeq$-minimal transitions of δ'' where the relation $\preccurlyeq$ is defined by $t' \preccurlyeq t$ if $t = (e, \alpha, e')$, $t' = (e, \alpha', e'')$, $\alpha \subseteq \alpha'$, $e'' \subseteq e'$, and $\forall T \in \mathcal{T}$, $t \in T \Rightarrow t' \in T$,
  - $\mathcal{T} = \{T_f \mid f \in F\}$ where $T_f = \{(e, \alpha, e') \mid f \notin e' \text{ or } \exists(\beta, e'') \in \delta(f),\ \alpha \subseteq \beta \text{ and } f \notin e'' \subseteq e'\}$.





VWAA to TGBA Translation

- Improved translation:

Formally, for each TGBA state $O = \{q_1, q_2, \ldots, q_n\}$ we define $\delta''(O)$ as follows:

$$\delta''(O) = \bigotimes_{i=1}^{n} \delta_O(q_i), \text{ where}$$

$$\delta_O(q_i) = \begin{cases} \{(\Sigma, \{q_i\})\} & \text{if } O \text{ contains a } \boxed{\text{progress non-alternating formula}} \\ & \text{and } q_i \text{ is an alternating formula,} \\ & \text{or } O \text{ contains a progress formula} \\ & \text{and } q_i \text{ is an alternating non-progress formula,} \\ \delta(q_i) & \text{otherwise.} \end{cases}$$

仍然是利用alternating公式的性质，但是要防止不在TGBA的接受路径中永远被suspend，应该只考虑不在TGBA的任何accepting cycle中的状态

- 之前接受状态集合T_f的定义不够直观，这里计算TGBA迁移的T_f为“do not contain any VWAA transition looping in f”.

VWAA to TGBA Translation

- Improved translation:

Let M be the minimal set containing all VWAA states of the form $\psi \,\mathsf{R}\, \rho$ and all subformulae of their right operands ρ. One can easily observe each TGBA state lying on some accepting cycle is a subset of M. The VWAA states outside M, called *progress formulae*, push TGBA computations towards accepting cycles. Suspension is enabled in a TGBA state only if it contains a progress formula. However, if all progress formulae in a TGBA state are alternating, their suspension is not allowed (as suspended progress formulae would not enforce any progress).

VWAA to TGBA Translation

- Improved translation:

Formally, for each TGBA state $O = \{q_1, q_2, \ldots, q_n\}$ we define $\delta''(O)$ as follows:

$$\delta''(O) = \bigotimes_{i=1}^{n} \delta_O(q_i), \text{ where}$$

$\delta(q\ F_a)$

$$\delta_O(q_i) = \begin{cases} \{(\Sigma, \{q_i\})\} & \text{if } O \text{ contains a progress non-alternating formula} \\ & \text{and } q_i \text{ is an alternating formula,} \\ & \text{or } O \text{ contains a progress formula} \\ & \text{and } q_i \text{ is an alternating non-progress formula,} \\ \delta(q_i) & \text{otherwise.} \end{cases}$$

仍然是利用alternating公式的性质，但是要防止不在TGBA的接受路径中永远被suspend，应该只考虑不在TGBA的任何accepting cycle中的状态

- 之前接受状态集合 T_f 的定义不够直观，这里计算TGBA迁移的 T_f 为“do not contain any VWAA transition looping in f”.

VWAA to TGBA Translation

- Improved translation: example

Alternating formula

Non-alternating progress formula

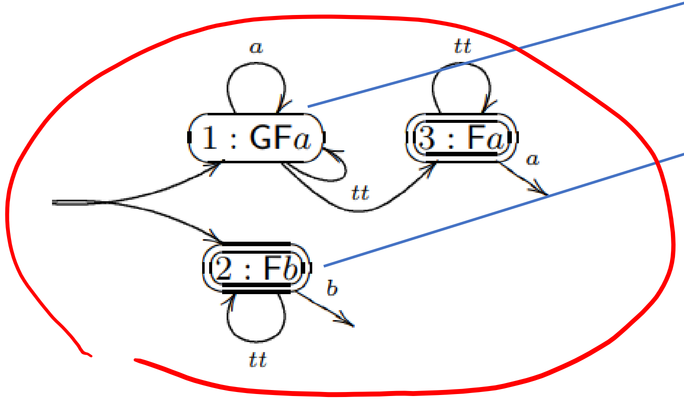


Fig. 3. A VWAA $\mathcal{A}_\psi$ corresponding to $\mathsf{GF}a \wedge \mathsf{F}b$

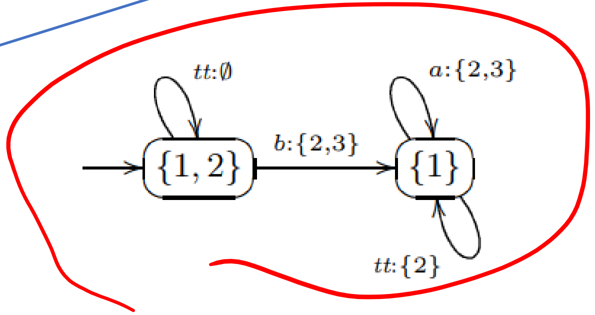
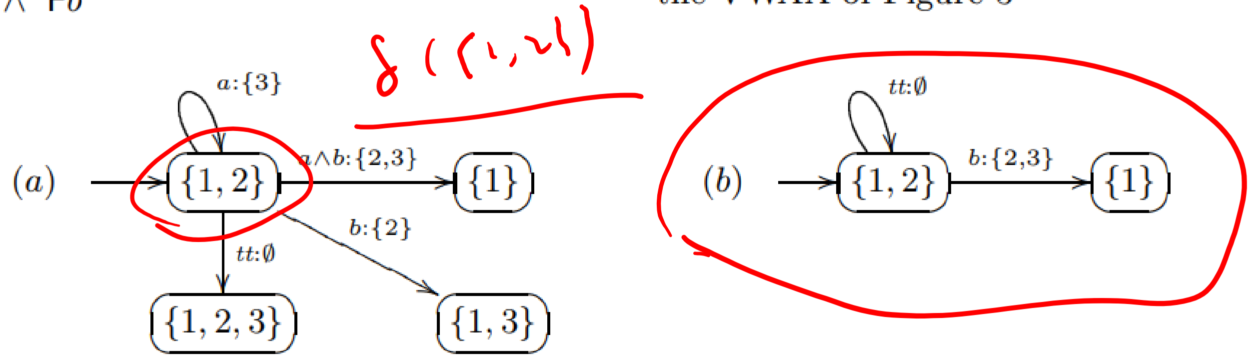


Fig. 4. A TGBA $\mathcal{G}_\psi$ corresponding to the VWAA of Figure 3



GFa is suspended

$\delta(GFa)$ 替换为 $\{(\Sigma, \{GFa\})\}$

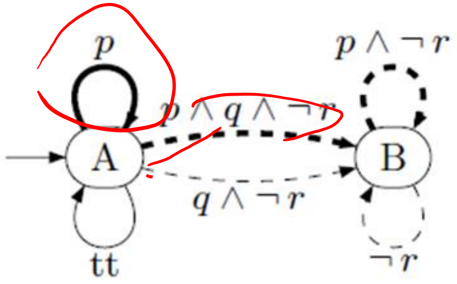
Fig. 5. Transitions leading from state $\{1,2\}$ in the TGBA constructed from the VWAA of Figure 3 by (a) the translation of [11] and by (b) our translation with suspension

TGBA to Büchi Automata

Step 3. Let $\mathcal{G} = (Q, \Sigma, \delta, I, \mathcal{T})$ be a GBA with $\mathcal{T} = \{T_1, \ldots, T_r\}$. We define the BA $\mathcal{B}_\mathcal{G} = (Q \times \{0, \ldots, r\}, \Sigma, \delta', I \times \{0\}, Q \times \{r\})$ where:

– $\delta'((q, j)) = \{(\alpha, (q', j')) \mid (\alpha, q') \in \delta(q) \text{ and } j' = next(j, (q, \alpha, q'))\}$.

$$\text{with next}(j,t) = \begin{cases} \max\{j \le i \le r \mid \forall j < k \le i,\ t \in T_k\} \text{ if } j \neq r \\ \max\{0 \le i \le r \mid \forall 0 < k \le i,\ t \in T_k\} \text{ if } j = r \end{cases}$$



r=2

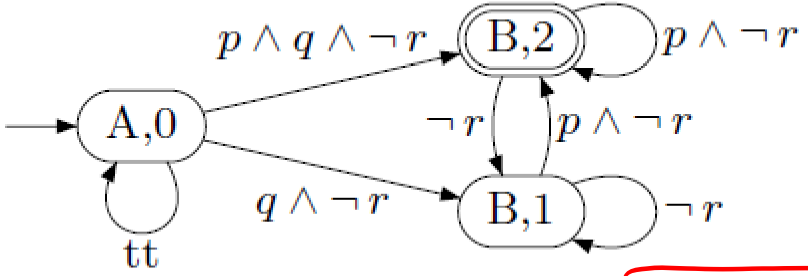


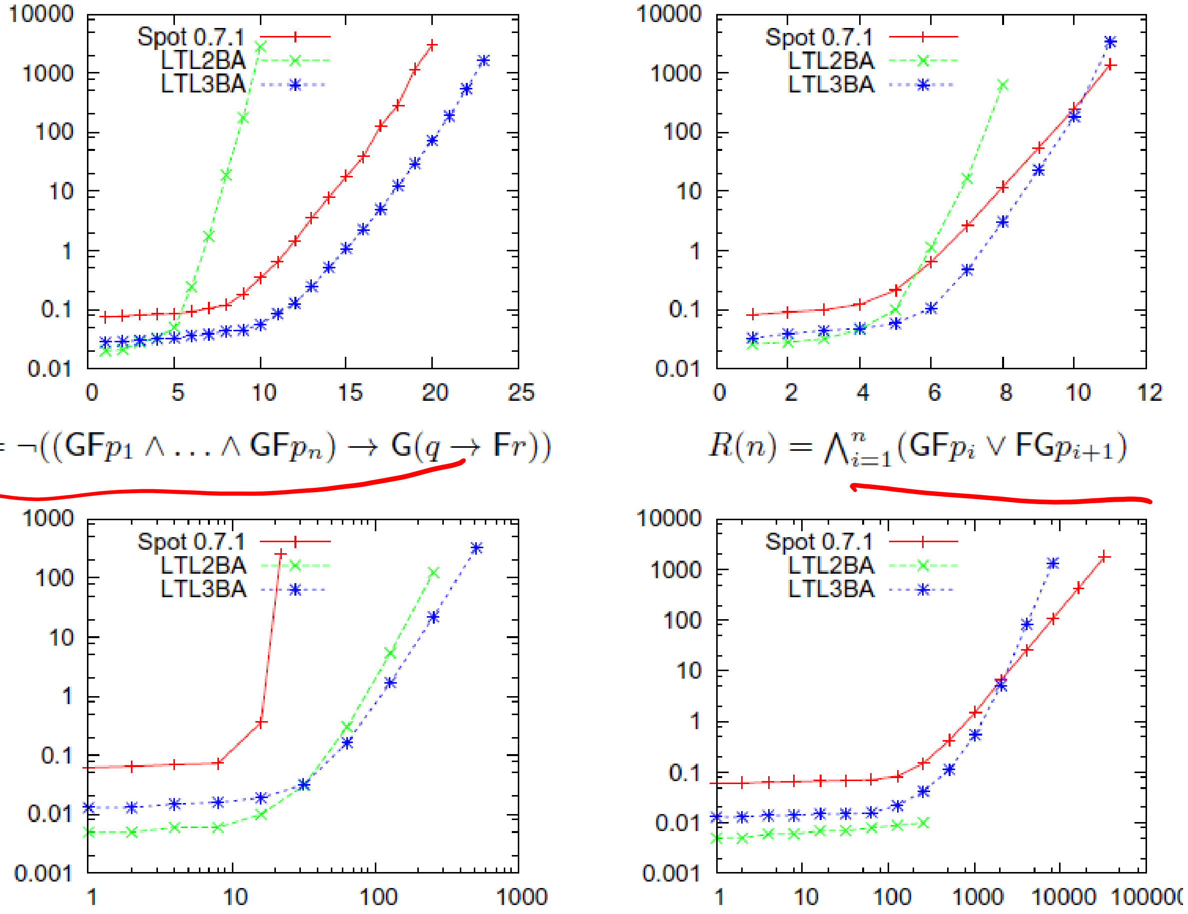
Fig. 3. Automaton $\mathcal{B}_{\mathcal{G}_{\mathcal{A}_\theta}}$ after Simplification.

Experiments

Table 1. Comparison of translators on two sets of random formulae. Time is in seconds, 'det. BA' is the number of deterministic automata produced by the translator. Note that, using WDBA minimization, SPOT failed to translate 6 formulae of Benchmark2 within the one hour limit. In order to see the effect of WDBA minimization to other formulae, we state in braces the original results increased by the values obtained when these 6 formulae were translated withut WDBA minimization.

Translator	Benchmark1				Benchmark2			
	States	Trans.	Time	det. BA	States	Trans.	Time	det. BA
SPOT	1 561	5 729	7.47	55	14 697	95 645	68.46	221
SPOT+WDBA	1 587	5 880	10.81	88	13 097 (14 408)	77 346 (94 248)	5 916.45 (5 919.43)	373 (373)
LTL2BA	2 118	9 000	0.81	25	24 648	232 400	18.57	84
LTL3BA(1)	1 621	5 865	1.26	27	17 107	129 774	22.25	92
LTL3BA(1,2)	1 631	6 094	1.41	54	15 936	115 624	9.04	237
LTL3BA(1,2,3)	1 565	5 615	1.41	54	14 113	91 159	8.53	240
LTL3BA(1,2,3,4)	1 507	5 348	1.38	54	13 244	85 511	8.30	240

Experiments



$\theta_n = \neg((\mathsf{GF}p_1 \wedge \ldots \wedge \mathsf{GF}p_n) \rightarrow \mathsf{G}(q \rightarrow \mathsf{F}r))$

$R(n) = \bigwedge_{i=1}^{n}(\mathsf{GF}p_i \vee \mathsf{FG}p_{i+1})$

$U_2(n) = p_1 \,\mathsf{U}\, (p_2 \,\mathsf{U}\, (\ldots p_{n-1} \,\mathsf{U}\, p_n) \ldots)$

$S(n) = \bigwedge_{i=1}^{n} \mathsf{G}p_i$

Fig. 6. Time consumption for parametric formulae constructed within an hour (the vertical axes are logarithmic and represent time in seconds, while the horizontal axes are linear or logarithmic and represent the parameter n)