Constructions in Rabinizer 2

Abstract. We provide the complete construction of automata and acceptance conditions of Rabinizer and show their correctness.

1 Linear Temporal Logic

This section recalls the notion of linear temporal logic (LTL). We consider a fragment with no occurrence of $U$ inside any $G$:

Definition 1 (LTL Syntax). The formulae of the $LTL_{\neg U}$-fragment of linear temporal logic are given by the following syntax for $\varphi$:

$$\varphi ::= a | \neg a | \varphi \land \varphi | \varphi \lor \varphi | X \varphi | \varphi U \varphi | F \varphi | G \xi$$

$$\xi ::= a | \neg a | \xi \land \xi | \xi \lor \xi | X \xi | F \xi | G \xi$$

over a finite fixed set $Ap$ of atomic propositions.

We use the standard abbreviations $tt := a \lor \neg a$, $ff := a \land \neg a$. We only have negations of atomic propositions, as negations can be pushed inside due to the equivalence of $F \varphi$ and $\neg G \neg \varphi$.

Definition 2 (LTL Semantics). Let $w \in (2^{Ap})^\omega$ be a word. The $i$th letter of $w$ is denoted $w[i]$, i.e. $w = w[0]w[1] \cdots$. Further, we define the $i$th suffix of $w$ as $w_i = w[i]w[i+1] \cdots$. The semantics of a formula on $w$ is then defined inductively as follows:

- $w \models a$ if $a \in w[0]$
- $w \models \neg a$ if $a \notin w[0]$
- $w \models \varphi \land \psi$ if $w \models \varphi$ and $w \models \psi$
- $w \models \varphi \lor \psi$ if $w \models \varphi$ or $w \models \psi$
- $w \models X \varphi$ if $w_1 \models \varphi$
- $w \models F \varphi$ if $\exists k \in \mathbb{N} : w_k \models \varphi$
- $w \models G \varphi$ if $\forall k \in \mathbb{N} : w_k \models \varphi$
- $w \models \varphi U \psi$ if $\exists k \in \mathbb{N} : w_k \models \psi$ and $\forall 0 \leq j < k : w_j \models \varphi$
2 Preliminaries

We define a symbolic one-step unfolding \(\text{Unf}\) of a formula inductively by the following rules:

\[
\begin{align*}
\text{Unf}(a) &= a \\
\text{Unf}(\neg a) &= \neg a \\
\text{Unf}(\varphi \land \psi) &= \text{Unf}(\varphi) \land \text{Unf}(\psi) \\
\text{Unf}(\varphi \lor \psi) &= \text{Unf}(\varphi) \lor \text{Unf}(\psi) \\
\text{Unf}(X\varphi) &= X\varphi \\
\text{Unf}(F\varphi) &= \text{Unf}(\varphi) \lor XF\varphi \\
\text{Unf}(G\varphi) &= \text{Unf}(\varphi) \land XG\varphi \\
\text{Unf}(\varphi U\psi) &= \text{Unf}(\psi) \lor (\text{Unf}(\varphi) \land X(\varphi U\psi))
\end{align*}
\]

Further, we define the “next step” operator. This peels off one next operator wherever possible. We define

\[
\begin{align*}
X^{-1}(\psi_1 \land \psi_2) &= X^{-1}(\psi_1) \land X^{-1}(\psi_2) \\
X^{-1}(\psi_1 \lor \psi_2) &= X^{-1}(\psi_1) \lor X^{-1}(\psi_2) \\
X^{-1}(X\psi) &= \psi \\
X^{-1}(\psi) &= \psi \text{ for all other types of formulae}
\end{align*}
\]

3 Algorithm

3.1 Construction of \(B(\xi)\)

We define a finite automaton \(B(\xi) = (Q_\xi, i_\xi, \delta_\xi, F_\xi)\) over \(2^{\text{Ap}}\) by

- the set of states \(Q_\xi = B^+(\text{sf}(\xi))\), where \(B^+(S)\) is the set of positive Boolean functions over \(S\) and \(\text{tt}\) and \(\text{ff}\),
- the initial state \(i_\xi = \xi\),
- the final states \(F_\xi\) where each atomic proposition has \(F\) or \(G\) as an ancestor in the syntactic tree (i.e. no atomic propositions are guarded by only \(X\)'s and Boolean connectives),
- transition relation \(\delta_\xi\) is defined by transitions

\[
\begin{align*}
X^{-1}(\chi[\nu]) & \text{ for every } \nu \subseteq \text{Ap} \text{ and } \chi \notin F \\
i & \rightarrow i & \text{ for every } \nu \subseteq \text{Ap}
\end{align*}
\]

where \(\chi[\nu]\) is the function \(\chi\) with \(\text{tt}\) and \(\text{ff}\) plugged in for atomic propositions according to \(\nu\) and \(X^{-1}\chi\) strips away the initial \(X\) (whenever there is one) from each formula in the Boolean combination \(\chi\). Note that we do not unfold inner \(F\)- and \(G\)-formulae. See an example for \(\xi = a \lor b \lor X(b \land Ga)\) below.
3.2 Construction of $A(\varphi)$

The state space has two components. Beside the component keeping track of
the input formula, we also keep track of the history for every recurrent formula
of $\mathcal{R}$. Formally, we define $A(\varphi) = (Q, i, \delta)$ to be a
deterministic finite automaton over $\Sigma = 2^{Ap}$ given by

- set of states $Q = C \times \prod_{\xi \in \mathcal{R}} 2^{Q_{\xi}}$
  where $C = B^+(\text{sf}(\varphi) \cup \text{Xsf}(\varphi))$ and $\text{XS} = \{Xs \mid s \in S\}$,
- the initial state $i = (\text{Unf}(\varphi), (\xi \mapsto \{i_\xi\})_{\xi \in \mathcal{R}})$;
- the transition function $\delta$ is defined by transitions

$$\langle \psi, (R_\xi)_{\xi \in \mathcal{R}} \rangle \xrightarrow{\nu} \langle \text{Unf}(X^{-1}(\psi[\nu])), (\delta(\xi, (R_\xi, \nu)))_{\xi \in \mathcal{R}} \rangle$$

3.3 Construction of generalized Rabin pairs condition $C(\varphi)$

We modify the construction of [KE12] and we provide a generalized Rabin condition of the form $\bigvee_i (F_i \land \bigwedge_j I_{ij})$ which can be easily degeneralized to a Rabin condition, where the conjunction is a singleton. In essence, the acceptance condition is responsible for non-deterministically guessing a set $I$ of subformulae that hold infinitely often and then checks that (1) they indeed hold infinitely often and (2) if they hold infinitely often then also $\varphi$ was satisfied in the initial state.

As for (1), whenever $F_\chi \in I$, we need to visit

$$\text{reach}_{F_\chi} := \{\langle \psi, (R_\xi)_{\xi \in \mathcal{R}(\psi)} \rangle \in Q \mid \exists q \in R_\chi \cap F_\chi : X^s = q \}$$

infinitely often, where $X^s = \{Xs \mid s \in S, n \in \mathbb{N}_0\}$. Similarly, whenever $G_\chi \in I$, we need to visit

$$\text{avoid}_{G_\chi} := \{\langle \psi, (R_\xi)_{\xi \in \mathcal{R}(\psi)} \rangle \in Q \mid \exists q \in R_\chi \cap F_\chi : X^s \neq q \}$$

only finitely often. As for (2), we allow only finitely many visits of states

$$\text{avoid} := \{\langle \psi, (R_\xi)_{\xi \in \mathcal{R}(\psi)} \rangle \in Q \mid X^s I \cup \bigcup_{G_\xi \in \mathcal{I}} R_\xi \neq \psi \}$$

where the set $I$ is insufficient to prove that $\varphi$ holds. Altogether

$$C := \bigvee_{I \subseteq G_\varphi \cup F_\varphi} \left(\text{avoid} \cup \bigcup_{G_\chi \in I} \text{avoid}_{G_\chi} \bigwedge_{F_\chi \in I} \text{reach}_{F_\chi}\right)$$
4 Correctness

Given a formula $\varphi$, we have defined a Rabin automaton $A(\varphi)$ and an acceptance condition $C := \bigvee_{I \subseteq G_{\varphi} \cup F_{\varphi}} P_I$. Every word $w : \mathbb{N} \to 2^{Ap}$ induces a run $\rho = A(\varphi)(w) : \mathbb{N} \to Q$ starting in $i$ and following $\delta$. The run is thus accepting and the word is accepted if the set of states visited infinitely often $\text{Inf}(\rho)$ is Muller accepting for $\varphi$. Vice versa, a run $\rho = i(\chi_1, \alpha_1)(\chi_2, \alpha_2) \cdots$ induces a word $Ap(\rho) = \alpha_1 \alpha_2 \cdots$. We now prove that this acceptance condition is sound and complete.

**Theorem 3.** Let $\varphi$ be a formula and $w$ a word. Then $w$ is accepted by the deterministic automaton $A(\varphi)$ with the generalized Rabin pairs condition $C(\varphi)$ if and only if $w \models \varphi$.

The second component of the state space takes care of identifying which recurrent formulae hold infinitely often or eventually always. For a word $w$, let $I(w) = \{ \psi \in G_{\varphi} \mid w \models F \psi \} \cup \{ \psi \in F_{\varphi} \mid w \models G \psi \}$.

**Lemma 4 (Correctness of $B(\xi)$’s).** For every word $w$ and every $\xi \in \text{Rec}$,

1. $w \models GF\xi$ iff $\exists i \in \mathbb{N} : \exists \chi \in B(\xi)(w)[i] \cap F_{\xi} : X^i I(w) \models \chi$,
2. $w \models FG\xi$ iff $\forall i \in \mathbb{N} : \forall \chi \in B(\xi)(w)[i] \cap F_{\xi} : X^i I(w) \models \chi$.

**Proof.** Since for every $n$, $w \models GF\chi$ iff $I \models GFX^n \chi$, and similarly $w \models FG\chi$ iff $I \models FGX^n \chi$, the lemma follows from the correctness of unfolding and $B(\xi)$ having the initial state self-loop as the only cycle. \qed

The first component of the state space takes care of all progress or failure in finite time.

**Lemma 5 (Local (finitary) correctness of $A(\varphi)$).** Let $w$ be a word and $A(\varphi)(w) = i(\chi_0, \alpha_0)(\chi_1, \alpha_1) \cdots$ the corresponding run. Then for all $n \in \mathbb{N}$, we have $w \models \varphi$ if and only if $w_n \models \chi_n$.

**Proof.** The one-step unfold produces a temporally equivalent (w.r.t. LTL satisfaction) formula. The unfold is a Boolean function over atomic propositions and elements of $Xsf(\varphi)$. Therefore, this unfold is satisfied if and only if the next state satisfies $X^{-1}(\psi)$ where $\psi$ is the result of partial application of the Boolean function to the currently read letter of the word. We conclude by induction. \qed

Further, each occurrence of satisfaction of $F$ must happen in finite time. As a consequence, a run with $\chi_i \neq \text{ff}$ is rejecting if and only if satisfaction of some $F \psi$ is always postponed.

**Proposition 6 (Completeness).** If $w \models \varphi$ then $\text{Inf}(A(\varphi)(w))$ is accepting w.r.t. $C(\varphi)$.
Proof. Let us show that the pair $P_I(w)$ is satisfied.

Firstly, we show $avoid$ is visited only finitely often, i.e. the first component $\psi$ is almost always (in states of $Inf(A(\varphi)(w))$ entailed by $X^*I(w)$ and the current states of $B(\xi)$ for each $G_\xi \in I(w)$. Consider some sufficiently large $i$ (for which $I(w)$ holds) and the corresponding $w_i$ and the current state $s_i = (\chi_i, (R_\xi)_{\xi \in rec}$.

By Lemma 5 we have $w_i \models \chi_i$. Notice that $\chi_i$ is a Boolean combination of $XF$-, $XU$- and $XG$-formulae and formulae produced by their unfolding. Whenever $F\psi$ is satisfied whenever entering $s_i$, it is in $I(w)$ and since in $\psi_i$ we have a disjunction of $XF\psi$ and the rest of the unfold, the entailment of this rest is irrelevant as the disjunction is entailed directly by $X^*I(w)$. Similarly, if $\psi_1 U \psi_2$ holds, the unfold (again a disjunction) is entailed since eventually $\psi_2$ holds and we proceed by induction. Finally, if $G\psi$ holds we need to show entailment of its unfolds.

This is a conjunction of $XG\psi$ and the unfolds of $\psi$ and their successors.

The former is entailed by $X^*I(w)$, the latter are the elements of $R_\xi$ (with $F$‘s and $G$‘s unfolded), which are thus entailed by $R_\psi$ (and the unfolded $F$‘s and $G$‘s are entailed recursively by the same argumentation).

Secondly, $avoid_{G\chi}$ is visited only finitely often for each $G\chi \in I(w)$. Indeed, since $w \models FG\chi$ almost all $w_i \models \chi$. Thus almost all tokens generated in $B(\xi)$ end up in a final state that holds in the current position. Since there are only finitely many of those and they are elements of $B^+(G_\varphi \cup F_\varphi)$, they are entailed by $X^*I(w)$ due to Lemma 4.

Thirdly, similarly $reach_{F\chi}$ is visited infinitely often for each $F\chi \in I(w)$.

Indeed, since $w \models GF\chi$ infinitely many $w_i \models \chi$. Thus infinitely many tokens generated in $B(\xi)$ end up in a final state that holds in the current position. Since there are only finitely many of those and they are elements of $B^+(G_\varphi \cup F_\varphi)$, they are entailed by $X^*I(w)$ due to Lemma 4.

Proposition 7 (Soundness). If $Inf(A(\varphi)(w))$ is accepting w.r.t. $C(\varphi)$ then $w \models \varphi$.

Proof. Let $M := Inf(A(\varphi)(w))$ be a accepting for pair $P_I$. There is $i \in \mathbb{N}$ such that after reading $i$ letters we come to $Inf(A(\varphi)(w))$ and stay there from now on and, moreover, $w_i \models \psi$ for all $\psi \in I$ by Lemma 4 and definition of $C$. Denote the $ith$ state by $\langle \psi, R \rangle$. By the definition of $avoid$, we get $w_i \models \psi$. By Lemma 5, we thus get $w \models \varphi$.

5 Optimizations

We optimize the construction as follows. Instead of keeping track of states of each $B(\xi)$, only the currently relevant ones. E.g. after reading $\emptyset$ in $GFa \lor (b \land GFc)$, it is no more interesting to track if $c$ occurs infinitely often. Formally, define $RelRec(\psi)(\xi)$ to be true iff $\xi$ occurs in the Boolean combination $\psi$. When the first component of a state is $\psi$, the second component is restricted to the vector with coordinates in $RelRec(\psi)$. The same holds for the definition of $avoid$.

Further, since only the infinite behaviour of $B(\xi)$ is important and it has acyclic structure (except for the initial states), instead of the initial state we can
start in any subset of states. Therefore, we start in a subset that is most likely to occur repetitively and we thus omit unnecessary initial transient parts of $\mathcal{A}(\varphi)$. 
Pseudocode

A  Notation:

For LTL formula \( \varphi \), \( sf(\varphi) \) denotes the set of all subformulae (any Boolean combination is one formula). Further, we denote by \( T(\varphi) \) the set of all \( X \)-, \( F \)-, \( G \)- and \( U \)-subformulae of \( \varphi \). For a set \( S \), \( B^+(S) \) is the set of positive Boolean functions over \( S \). The closure of \( \varphi \) is then \( C(\varphi) := \{tt, ff\} \cup Ap \cup \{\neg a \mid a \in Ap\} \cup T(\varphi) \cup XT(\varphi) \) where \( XS = \{Xs \mid s \in S\} \) and further \( X^*S = \{X\cdots Xs \mid s \in S, n \in \mathbb{N}_0\} \).

B  Main algorithm:

input: \( \varphi \in LTL \)

1. if \( \varphi \) not in \( LTL_{GU} \) then return “not in the LTL fragment”
2. compute type 2 formulae:
   \( G_{\varphi} := \{G\psi \in sf(\varphi)\} \)
   \( F_{\varphi} := \{F\psi \in sf(\omega) \mid \text{for some } \omega \in G_{\varphi}\} \)
   \( \text{Rec} := \{\psi \mid G\psi \in G_{\varphi} \text{ or } F\psi \in F_{\varphi}\} \)
   /*go down the tree and take every child of \( G \), and further every child of \( F \)
   if you already saw \( G \) on this branch*/
   /* we can take progress formulae only */
3. foreach \( \xi \in \text{Rec} \) construct \( B(\xi) \)
4. construct \( A(\varphi) \)
5. construct GR acceptance condition \( C \)
6. if \( C \) empty then return “unsat”
7. else
8. output \( A(\varphi), C \)
9. perform Andreas’ degeneralization and output its result

C  Auxiliary functions:

\( \text{Unf} : B^+(C(\varphi)) \to B^+(C(\varphi)) \):

\[
\begin{align*}
\text{Unf}(a) &= a \\
\text{Unf}(\neg a) &= \neg a \\
\text{Unf}(\varphi \land \psi) &= \text{Unf}(\varphi) \land \text{Unf}(\psi) \\
\text{Unf}(\varphi \lor \psi) &= \text{Unf}(\varphi) \lor \text{Unf}(\psi) \\
\text{Unf}(X\varphi) &= X\varphi \\
\text{Unf}(F\varphi) &= \text{Unf}(\varphi) \lor XF\varphi \\
\text{Unf}(G\varphi) &= \text{Unf}(\varphi) \land XG\varphi \\
\text{Unf}(\varphi U\psi) &= \text{Unf}(\psi) \lor (\text{Unf}(\varphi) \land X(\varphi U\psi))
\end{align*}
\]
\( X^{-1} : B^+(C(\phi)) \rightarrow B^+(C(\phi)):\)

\[
\begin{align*}
X^{-1}(\psi_1 \land \psi_2) &= X^{-1}(\psi_1) \land X^{-1}(\psi_2) \\
X^{-1}(\psi_1 \lor \psi_2) &= X^{-1}(\psi_1) \lor X^{-1}(\psi_2) \\
X^{-1}(X\psi) &= \psi
\end{align*}
\]

\( X^{-1}(\psi) = \psi \) for all other types of formulae

Consider a formula \( \chi \in B^+(C(\phi)) \). For a set \( S \subseteq C(\phi) \), let \( \chi[S \mapsto \text{tt}] \) denote the formula where \( \text{tt} \) is substituted for elements of \( S \). As elements of \( C(\phi) \) are considered to be atomic expressions here, the substitution is only done on the propositional level and does not go through the modality, e.g. \( (a \land XG a)[\{a\} \mapsto \text{tt}] = \text{tt} \land XG a \), which is equivalent to \( XG a \) in the propositional semantics. For a valuation \( \nu \subseteq Ap \), we set \( \chi[\nu] := \chi[\nu \cup \{-a \mid a \in Ap \setminus \nu\} \mapsto \text{tt}] \).

\( \mathcal{R}_\text{elRec} : B^+(C(\phi)) \rightarrow 2^{\text{Rec}}: \)

\( \mathcal{R}_\text{elRec}(\psi)(\xi) \) iff \( \xi \) occurs in the Boolean combination \( \psi \).

**D Automaton \( B(\xi) \) construction:**

input: \( \xi \in \text{Rec} \)

output: \( B(\xi) = (Q_\xi, i_\xi, \delta_\xi, F_\xi) \) over \( 2^{Ap} \)

1. the initial state \( i_\xi := \xi \)
2. worklist := \( \{i_\xi\} \)
3. while worklist \( \neq \emptyset \)
   (a) pop \( q \in \text{worklist} \)
   (b) if \( q \notin F_\xi \) then foreach \( \nu \subseteq Ap \)
       new := \( X^{-1}(\chi[\nu]) \)
       add \((q, \nu, \text{new})\) to \( \delta_\xi \)
       if \( \text{new} \notin Q_\xi \) then add new to worklist and \( Q_\xi \)
       if (each atomic proposition has \( F \) or \( G \) as an ancestor in the syntactic tree of \( \text{new} \)) then add \( q \) to \( F_\xi / *i.e. no atomic propositions are guarded by only \( X^* \) and Boolean operators*/
4. foreach \( \nu \subseteq Ap \) add \((i, \nu, i)\) to \( \delta_\xi \)
E Automaton $\mathcal{A}(\varphi)$ construction:

output: $\mathcal{A}(\varphi) = (Q, i, \delta)$ over $\Sigma = 2^{Ap}$

1. for each $G_\xi \in G_\varphi$, 
pick $f_\xi$ to be (1) $tt$ if $tt \in F_\xi$ else (2) any $\psi \neq ff$ if $\psi \in F_\xi$ else (3) $ff$
2. for each $F_\xi \in F_\varphi$, 
pick $f_\xi$ to be (1) $ff$ if $ff \in F_\xi$ else (2) any $\psi \neq tt$ if $\psi \in F_\xi$ else (3) $tt$
3. for each $\xi \in \text{Rec}$, 
$S_\xi :=$ states on an arbitrary path from $i_\xi$ to $f_\xi$ including both
4. the initial state $i := \langle \text{Unif}(\varphi), (S_\xi)_{\xi \in \text{Rec}} \rangle$
5. worklist $:= \{i\}$
6. while worklist $\neq \emptyset$
   (a) pop $q := \langle \psi, (R_\xi)_{\xi \in \text{RelRec}(\psi)} \rangle \in$ worklist 
   (b) foreach $\nu \subseteq Ap$
      i. $new_\psi := \text{Unif}(X^{-1}(\psi[\nu]))$ /*and search for the label*/
      ii. foreach $\xi \in \text{RelRec}(\psi)$ do
          new$_\xi := \bigcup_{q \in R_\xi} \delta_\xi(q, \nu)$
      iii. new $:= \langle new_\psi, (new_\xi)_{\xi \in \text{RelRec}(new_\psi)} \rangle$ /*new vector can be smaller*/
      iv. add $(q, \nu, new)$ to $\delta$
   v. if new $\not\in Q$ then add new to worklist and $Q$

F GR acceptance condition $\mathcal{C}$ construction:

1. foreach $I \subseteq G_\varphi \cup F_\varphi$
   (a) avoid $:= \{\langle \psi, (R_\xi)_{\xi \in \text{RelRec}(\psi)} \rangle \in Q \mid X^*I \cup \bigcup_{\xi \in I: G_\xi \in \text{ist}(\varphi)} R_\xi \not\models \psi\}$
   (b) foreach $G_\chi \in I$
      avoid$_{G_\chi} := \{\langle \psi, (R_\xi)_{\xi \in \text{RelRec}(\psi)} \rangle \in Q \mid \exists q \in R_\chi \cap F_\chi : X^*I \not\models q\}$ /* e.g. states where the segment automaton for $\chi$ is in $ff$ */
   (c) foreach $F_\chi \in I$
      reach$_{F_\chi} := \{\langle \psi, (R_\xi)_{\xi \in \text{RelRec}(\psi)} \rangle \in Q \mid \exists q \in R_\chi \cap F_\chi : X^*I \models q\}$ /* e.g. states where the segment automaton for $\chi$ is in $tt$ or in some element of $I$ */
   (d) add $(\text{avoid} \cup \bigcup_{G_\chi \in I} \text{avoid}_{G_\chi}, \{\text{reach}_{F_\chi} \mid F_\chi \in I\})$ to $\mathcal{C}$
2. perform redundancy removals on $\mathcal{C}$ described on the webpage of Rabinizer (version 1) and return the result
   /* such as $([1, 2], \{\{2\}, \{3\}\})$ is redundant w.r.t. $([1], \{\{2, 3\}\})$ */
References

[CGK13] Krishnendu Chatterjee, Andreas Gaiser, and Jan Křetínský. Automata with
generalized Rabin pairs for probabilistic model checking and LTL synthesis.

[GKE12] Andreas Gaiser, Jan Křetínský, and Javier Esparza. Rabinizer: Small deter-
ministic automata for LTL(F,G). In ATVA, pages 72–76, 2012.

[KB06] Joachim Klein and Christel Baier. Experiments with deterministic omega-

[KE12] Jan Křetínský and Javier Esparza. Deterministic automata for the (F,G)-
fragment of LTL. In CAV, pages 7–22, 2012.