Supporting Materials for: Triggers for Reactive Synthesis Specifications

Gal Amram[1], Dor Ma'ayan[1], Shahar Maoz[1], Or Pistiner[1], and Jan Oliver Ringert[2]

[1] Tel Aviv University, Israel
[2] Bauhaus University Weimar, Germany

Abstract. This document provides supporting materials for the ICSE'23 paper titled Triggers for Reactive Synthesis Specifications. We provide a proof for the correctness of the construction from Sect. V. We provide a proof for the optimality of the encoding.

1 Construction (extended with special cases)

Recall that Step I translates a trigger into two DSFAs $\mathcal{A}^1$ and $\mathcal{A}^2$. We now repeat Step II with two special cases (the paper shows the most general case — case (3)).

Step II (extended with special cases (1) and (2)) Given the two DSFAs, $\mathcal{A}^1$ and $\mathcal{A}^2$, we consider three cases.

Case 1. $\varepsilon \in L(\mathcal{A}^2)$. In this case, the trigger is trivially satisfied by any infinite play. Hence, we do not need to add anything to the game structure and to the GR(1) formula.

Case 2. $\varepsilon \notin L(\mathcal{A}^2)$, but $\varepsilon \in L(\mathcal{A}^1)$. In this case, an infinite play π satisfies the trigger iff it can be written as $\pi = w_0 w_1 w_2 \dots$ where each $w_i \in \mathit{first}(L(\mathcal{A}^2))$. Hence, intuitively, we use $\mathcal{A}^2$ to track the play, and each time $\mathcal{A}^2$ accepts, we wish to go back to the initial state, to check whether it will reach an accepting state again. Hence, one may suggest to redirect every transition to $\mathcal{F}^2$ into q_0^2 instead, and to mark q_0^2 as a single accepting state. However, this suggestion is incorrect since it may produce an SFA that accepts plays that do not satisfy the trigger. This happens when $q_0^2 \xrightarrow{w} q_0^2$ for some $w \neq \varepsilon$. By marking q_0^2 as accepting, the SFA will accept the play $www\cdots$, although it does not satisfy the trigger.

To fix this problem, we add a new initial state q_{ini} to the DSFA, with the same outgoing transitions as q_0^2, and redirect transitions to $\mathcal{F}^2$, into the added initial state. The construction is as follows. Let

- $I = \{(q_{ini}, assrt, q) : (q_0^2, assrt, q) \in \delta^2\}$,
- $\delta^2_{\mathcal{F}^2} = \{(q, assrt, p) \in \delta^2 : q \in Q^2 \setminus \mathcal{F}^2, p \in \mathcal{F}^2\}$, and
- $\delta^2_{q_{ini}} = \{(q, assrt, q_{ini}) : \exists p((q, assrt, p) \in \delta^2_{\mathcal{F}^2})\}$.

The trigger DSFA is $\mathcal{A}_{trig} = (Q_{trig} = (Q^2 \setminus \mathcal{F}^2) \cup \{q_{ini}\}, \delta_{trig} = (\delta^2 \setminus \delta^2_{\mathcal{F}^2}) \cup \delta^2_{q_{ini}} \cup I, q_0 = q_{ini}, \mathcal{F}_{trig} = \{q_{ini}\})$.

Case 3. $\varepsilon \notin L(\mathcal{A}^2)$ and $\varepsilon \notin L(\mathcal{A}^1)$. In this case, an infinite play π satisfies the trigger iff one of the following holds:

(3.1) π can be written as $\pi = w_0 w_1 w_2 \cdots$, where each $w_{2i} \in \mathit{first}(L(\mathcal{A}^1))$ and each $w_{2i+1} \in \mathit{first}(L(\mathcal{A}^2))$.

(3.2) π can be written as $w_0 w_1 \cdots w_{2j+1} w'$, where each $w_{2i} \in \mathit{first}(L(\mathcal{A}^1))$, each $w_{2i+1} \in \mathit{first}(L(\mathcal{A}^2))$, and no prefix of w' is in $\mathit{first}(L(\mathcal{A}^1))$.

Therefore, in this case, we concatenate the prefix and suffix DSFAs. Whenever $\mathcal{A}^1$ accepts, we go to the initial state of the suffix DSFA, and whenever $\mathcal{A}^2$ accepts, we go to the initial state of $\mathcal{A}^1$. A play π satisfies the trigger iff its computation (a) traverses between $\mathcal{A}^1$ and $\mathcal{A}^2$ infinitely often, or (b) from some point, never leaves $\mathcal{A}^1$. Hence, we redirect transitions to $\mathcal{F}^1$ into q^2_0, redirect transitions to $\mathcal{F}^2$ into q^1_0, and mark all states of $\mathcal{A}^1$ as accepting.

The formal construction is as follows. Let

- $\delta^1_{\mathcal{F}^1} = \{(q, assrt, q') \in \delta^1 : q \in Q^1 \setminus \mathcal{F}^1, q' \in \mathcal{F}^1\}$,
- $\delta^1_{q^2_0} = \{(q, assrt, q^2_0) : \exists q'((q, assrt, q') \in \delta^1_{\mathcal{F}^1})\}$,
- $\delta^2_{\mathcal{F}^2} = \{(q, assrt, q') \in \delta^2 : q \in Q^2 \setminus \mathcal{F}^2, q' \in \mathcal{F}^2\}$, and
- $\delta^2_{q^1_0} = \{(q, assrt, q^1_0) : \exists q'((q, assrt, q') \in \delta^2_{\mathcal{F}^2})\}$.

Then, $\mathcal{A}_{trig} = (Q_{trig} = (Q^1 \cup Q^2) \setminus (\mathcal{F}^1 \cup \mathcal{F}^2), \delta_{trig} = ((\delta^1 \setminus \delta^1_{\mathcal{F}^1}) \cup \delta^1_{q^2_0}) \cup ((\delta^2 \setminus \delta^2_{\mathcal{F}^2}) \cup \delta^2_{q^1_0}), q_0 = q^1_0, \mathcal{F}_{trig} = Q^1 \setminus \mathcal{F}^1)$.

2 A Correctness Proof for the Construction in Section V

We start by proving that $\mathcal{A}_{trig}$ is indeed deterministic.

Lemma 1. *$\mathcal{A}_{trig}$ is a DSFA.*

Proof. We prove only for case 3, as the proof for case 2 is similar. Assume, towards a contradiction, that $(q, assrt_1, p_1), (q, assrt_2, p_2) \in \delta_{trig}$, $p_1 \neq p_2$, but $(assrt_1 \wedge assrt_2) \not\equiv \texttt{false}$. W.l.o.g., assume that $q \in Q^1$. Since $\mathcal{A}^1$ is deterministic, these are not "old" transitions. In addition, since $p_1 \neq p_2$, by the construction, not both are "new" transitions either thus, w.l.o.g., $(q, assrt_1, p_1) \in \delta^1$ ("old") and $(q, assrt_2, p_2) \in \delta^1_{q^2_0}$ ("new"). Therefore, for some $p \in \mathcal{F}^1$, $(q, assrt_2, p) \in \delta^1$. In addition, since $(q, assrt_2, p_1) \in \delta_{trig}$, $(q, assrt_2, p_1) \notin \delta^1_{\mathcal{F}^1}$ (othwerwise, the construction removes it) thus $p_1 \neq p$. We get that $\mathcal{A}^1$ is not deterministic, in contradiction to our assumptions.

Now, we argue that our construction is correct.

Lemma 2. *Let π be an infinite play. $\pi \models trig$ iff the computation of $\mathcal{A}_{trig}$ on π traverses $\mathcal{F}$ infinitely often.*

Proof. We prove only for case 3, as case 1 is trivial, and the proof for case 2 is similar. First, we note that since $\varepsilon \notin L(\mathcal{A}^1) \cup L(\mathcal{A}^2)$, $q_0^1 \notin \mathcal{F}^1$, and $q_0^2 \notin \mathcal{F}^2$ (and thus, in particular, the constructions are well defined). Now, by the construction of $\mathcal{A}_{trig}$, we observe that the following hold for a word $w \in V(\mathcal{V})^*$:

- $w \in \mathit{first}(L(\mathit{prefix}))$ iff $q_0 = q_0^1 \xrightarrow{w} q_0^2$, and, excluding the final state q_0^2, the computation does not traverse any state of Q^2.
- $w \in \mathit{first}(L(\mathit{suffix}))$ iff $q_0^2 \xrightarrow{w} q_0^1 = q_0$, and, excluding the final state q_0^1, the computation does not traverse any state of Q^1.

Consequently, for an infinite play π, we have

- $\pi = w_0 w_1 \cdots$ where each $w_{2i} \in \mathit{first}(L(\mathit{prefix}))$, and each $w_{2i+1} \in \mathit{first}(L(\mathit{suffix}))$ iff the computation of $\mathcal{A}_{trig}$ on π traverses $Q_0^1 \setminus \mathcal{F}^1$ and $Q_0^2 \setminus \mathcal{F}^2$ infinitely often.
- $\pi = w_0 w_1 \cdots w_{2j+1} w'$ where each $w_{2i} \in \mathit{first}(L(\mathit{prefix}))$, each $w_{2i+1} \in \mathit{first}(L(\mathit{suffix}))$, and no prefix of w' belongs to $\mathit{first}(L(\mathit{prefix}))$ iff, from some point, the computation of $\mathcal{A}_{trig}$ on π never leaves $Q_0^1 \setminus \mathcal{F}^1$.

Therefore, $\pi \models \mathit{trig}$ iff the computation of $\mathcal{A}_{trig}$ on π traverses $\mathcal{F}_{trig} = Q^1 \setminus \mathcal{F}^1$ infinitely often, as required.

3 Optimality of the Encoding: Proof of Theorem 4.5

To provide a rigorous proof, we first provide precise definitions.

Definition 1. *Let $\mathcal{V}$ be a set of Boolean variables and $P \subseteq (2^{\mathcal{V}})^\omega$ a property over $\mathcal{V}$. An encoding of P is an LTL formula ψ over $\mathcal{V} \cup AUX$, where AUX is a set of boolean variables, disjoint to $\mathcal{V}$, such that:*

1. *For $\pi \in (2^{\mathcal{V} \cup AUX})^\omega$, if $\pi = s_0, s_1, \ldots \models \psi$, then $\pi|_{\mathcal{V}} = s_0 \cap \mathcal{V}, s_1 \cap \mathcal{V}, \cdots \in P$.*
2. *For $\pi \in (2^{\mathcal{V}})^\omega$, if $\pi = s_0, s_1, \cdots \in P$, then π can be extended into $\pi' = s_0 \cup a_0, s_1 \cup a_1, \ldots$ over $\mathcal{V} \cup AUX$, such that $\pi' = \psi$.*[3]

Definition 2. *A GR(1) encoding, a special case of LTL encoding, is a formula of the form $init \wedge safe \wedge justice$ where:*

1. *$init$ is an assertion over $\mathcal{V} \cup AUX$.*
2. *$safe$ is an assertion over $\mathcal{V} \cup AUX \cup \{\boldsymbol{X}(v) : v \in \mathcal{V} \cup AUX\}$.*
3. *$justice = \wedge_{i=1}^{k} \boldsymbol{GF}(asrt_i)$ where each $asrt_i$ is an assertion over $\mathcal{V} \cup AUX$.*

We are ready to show that our encoding is optimal.

Theorem. No GR(1) encoding adds less than $\log(|Q^1| + |Q^2| - 2)$ auxiliary variables.

[3] For synthesis purposes, it makes sense to require that a *unique* extension exists. We did not add this assumption as our proof works without it.

Proof. Take $\mathcal{V} = \{v\}$. For $k > 1$, consider the trigger $[\mathtt{true}]\{2^k{-}1\}|{=}{>}[v]$. That is, the trigger formulates the requirement that v must hold every 2^k steps. The left RE is accepted by a 2^k-state DSFA, and the right RE by a 2-state DSFA. Our encoding adds $k = \log(2^k)$ variables (since the construction omits the two accepting states), and, to prove the claim, we show that every GR(1) encoding adds, at least, k variables.

Assume towards a contradiction that $\psi = \mathit{init} \wedge \mathit{safe} \wedge \mathit{jutice}$ is a GR(1) encoding that adds $k{-}1$ variables, AUX. Take $\pi = s_0, s_1, \cdots \in (2^{\mathcal{V}})^\omega$ such that $s_i = v$ iff $i \mod 2^k = 2^k{-}1$. In more detail, $s_i = \{v\}$ when $i \mod 2^k = 2^k{-}1$, and $s_i = \emptyset$ otherwise. Therefore, $\pi \models [\mathtt{true}]\{2^k{-}1\}|{=}{>}[v]$ and thus can be extended into $\pi' = (s_0 \cup a_0), (s_1 \cup a_1), \cdots \in (2^{\mathcal{V} \cup AUX})^\omega$, $\pi' \models \psi$.

Consider the states $(s_{2^k-1} \cup a_{2^k-1}), (s_{2^k} \cup a_{2^k}), \ldots, (s_{2^{k+1}-1} \cup a_{2^{k+1}-1})$. In this subsequence of π', only the first and last states satisfy v. Furthermore, by the pigeonhole principle, the states $a_{2^k}, \ldots, a_{2^{k+1}-2}$ are not all different. That is, for some $0 \leq t_1 < t_2 \leq 2^k - 2$, $a_{2^k+t_1} = a_{2^k+t_2}$. To achieve a contradiction, we construct a third play π'', by duplicating the cycle that starts with $(s_{2^k+t_1} \cup a_{2^k+t_1})$ and ends with $(s_{2^k+t_2} \cup a_{2^k+t_2})$. This is indeed a cycle since $a_{2^k+t_1} = a_{2^k+t_2}$, and $s_{2^k+t_1} = s_{2^k+t_2} = \emptyset$. The resulting play satisfies ψ, but it does not satisfy the trigger.

Formally, we take

$$\begin{aligned}\pi'' = (s_0 \cup a_0), \ldots, (s_{2^k-1} \cup a_{2^k-1}), \ldots,\\ (s_{2^k+t_1} \cup a_{2^k+t_1}), \ldots, (s_{2^k+t_2-1} \cup a_{2^k+t_2-1}),\\ \boldsymbol{(s_{2^k+t_1} \cup a_{2^k+t_1}), \ldots, (s_{2^k+t_2-1} \cup a_{2^k+t_2-1}),}\\ (s_{2^k+t_2} \cup a_{2^k+t_2}), \ldots, (s_{2^{k+1}-1} \cup a_{2^{k+1}-1}), \ldots\end{aligned}$$

By duplicating the subsequence, $(s_{2^k+t_1} \cup a_{2^k+t_1}), \ldots, (s_{2^k+t_2-1} \cup a_{2^k+t_2-1})$, the assertion $[v]$ does not hold for $2^k - 1 + t_2 - t_1 \geq 2^k$ consecutive states, and thus $\pi'' \not\models [\mathtt{true}]\{2^k{-}1\}|{=}{>}[v]$. However, we still have $\pi'' \models \psi$ (note that $\pi'' \models \mathit{safe}$, since every transition in π'' also occurs in π'). Therefore, ψ does not encodes the trigger, in contradiction to the assumption. □

We do not know of lower bounds for general LTL encoding of triggers. However, we note a sort of a trade-off between the number of auxiliary variables and the "hardness" of the encoding. Specifically, the trigger $[\mathtt{true}]^*((([\mathtt{true}]^*[a])\{k\})\&([!b]^*))|{=}{>}[\mathtt{false}]$ adds $\log(k)$ variables with our GR(1) encoding. As the property it expresses can be written in LTL (see Sect. 3.3), it does not require any auxiliary variable with LTL encoding. However, as we mentioned in Sect. 3.3, any LTL formula that is equivalent to this trigger has k nested instances of the until operator, and thus constitutes a difficult instance for writing and for an LTL synthesizer. This suggests that the deprivation of auxiliary variables via LTL encoding may not be worthwhile.