Undecidability of Stratification of Commutativity Relations

AZADEH FARZAN, University of Toronto, Canada

1 INTRODUCTION

This note sets the record straight on a previously published paper [1] that wrongly claimed the decidability of an inclusion check up to the stratification of several commutativity relations.

In particular, this note proves that the inclusion check [1, Equation 4], for which an algorithm is proposed in [1],

$$\mathrm{red}^{I_1}_{\leq}\Big(\mathrm{red}^{I_2}_{\leq}(P) \cup \Pi\Big) \subseteq \Pi \qquad \text{(RedStratification Union Inclusion)}$$

is in general *undecidable*. We show that it suffices to use a lexicographical order $\leq$, that is induced by a total alphabet order, to define the reduction to get the undecidability result. The proposed algorithm of [1] remains *sound* for deciding the inclusion [4]: If the algorithm says yes, then

$$P \subseteq \mathrm{cl}_{I_1}(\mathrm{cl}_{I_2}(\Pi))$$

The only erroneous part is the fact that the check in the algorithm coincides with the specific RedStratification Union Inclusion check.

The RedStratification Union Inclusion check was devised as a modification of an original check

$$\mathrm{red}^{I_1}_{\leq}\Big(\mathrm{red}^{I_2}_{\leq}(P)\Big) \subseteq \Pi \qquad \text{(RedStratification Inclusion)}$$

which was not settled in [1] as decidable or not. The modification was meant as the adjustment that preserved the purpose of the check in a verification context, and yet yielded decidability; which we now know is wrong. In this note, we settle the second question as well. We show that this original check is also *undecidable*. However, for this second proof of undecidability, we have to rely on using a *contextual* order [2] for $\leq$.

Both undecidability proofs are through a reduction to the well-known result of Gibbons and Rytter [3], on undecidability of checking the universality of the trace closure of a regular language. Both proofs have been formalized and checked in Lean. This formalization, however, excludes the proof of undecidability of the Gibbons/Rytter result, and simply trusts it.

2 DEFINITIONS

Let I be a symmetric, irreflexive relation on a finite alphabet Σ. *Trace equivalence*, $u \equiv_I v$, is defined as u being transformable to v by repeatedly swapping adjacent letters related by I.

$$\mathrm{cl}_I(L) = \{v \in \Sigma^* \mid \exists u \in L :\ u \equiv_I v\}.$$

We use a special notation $[u]_I$ to define the equivalence class containing the word $u \in \Sigma^*$:

$$[u]_I = \mathrm{cl}_I(\{u\})$$

For a total lexicographic order $\leq$ on Σ^*, the general reduction of an arbitrary language L is defined as

$$\mathrm{red}^I_{\leq}(L) = \{\min_{\leq}\big([u]_I \cap L\big) \mid u \in L\}.$$

Author's address: Azadeh Farzan, University of Toronto, Toronto, Canada, azadeh@cs.toronto.edu.

3 UNDECIDABILITY OF RedStratification Union Inclusion CHECK

Gibbons and Rytter assert that there exists an alphabet Γ and a commutativity relation J on Γ, such that the following problem is undecidable [3, Theorem 2]:

Given a regular language $L \subseteq \Gamma^*$, decide whether $\mathrm{cl}_J(L) = \Gamma^*$.

Equivalently, the test asks if every J-equivalence class of Γ^* contains at least one word of L. Below, we refer to L, J, and Γ, and we always mean them to be the ingredients from the above construction.

3.1 The Reduction from a Universality Check

Define a new alphabet Σ based on the aforementioned Γ from [3, Theorem 2]:

$$\Sigma = \Gamma \cup \{a, b, c, \#\}.$$

Choose a total order on letters extending

$$a < c < b,$$

and let $\preceq$ be its ordinary lexicographic extension to words. Define

$$I_1 = J \cup \{(a, b), (b, a)\},$$
$$I_2 = \{(b, c), (c, b)\}.$$

All unspecified pairs are dependent; in particular, # commutes with nothing.

Now define the program language P as

$$P = (bac \ \cup \ abc \ \cup \ acb)\, \Gamma^*\, \#.$$

For the language $L \subseteq \Gamma^*$, define the proof language

$$\Pi_L = abc\, L\, \# \ \cup \ acb\, \Gamma^*\, \#.$$

Note that P and Π are both regular by construction and $\Pi_L \subseteq P$. Furtheremore, for all $u \in \Gamma^*$, we have

$$bac\, u\, \# \equiv_{I_1} abc\, u\, \# \equiv_{I_2} acb\, u\, \#, \qquad acb\, u\, \# \prec abc\, u\, \# \prec bac\, u\, \#.$$

Lemma 3.1. *P is closed under both I_1 and I_2.*

Proof. Clear by construction. Γ^* is closed under every relation, and P includes all commutations for the prefixes over $\{a, b, c\}$, which do not commute against anything in Γ. □

Observe that

$$\mathrm{red}_{\preceq}^{I_2}(P) = (bac \ \cup \ acb)\, \Gamma^*\, \#.$$

Consequently,

$$\mathrm{red}_{\preceq}^{I_2}(P) \cup \Pi_L = bac\, \Gamma^*\, \# \ \cup \ acb\, \Gamma^*\, \# \ \cup \ abc\, L\, \#.$$

Lemma 3.2 (Outer reduction). *For every regular $L \subseteq \Gamma^*$,*

$$\mathrm{red}_{\preceq}^{I_1}(\mathrm{red}_{\preceq}^{I_2}(P) \cup \Pi_L) \subseteq \Pi_L \quad \Longleftrightarrow \quad \mathrm{cl}_J(L) = \Gamma^*. \tag{1}$$

Proof. Recall that the right-hand side of Equation 1 can alternatively be interpreted as

$$\forall u \in \Gamma^* : \ [u]_J \cap L \neq \emptyset$$

We are interested in I_1-equivalence classes of

$$\mathrm{red}_{\preceq}^{I_2}(P) \cup \Pi_L = bac\, \Gamma^*\, \# \ \cup \ acb\, \Gamma^*\, \# \ \cup \ abc\, L\, \#.$$

For any $u \in \Gamma^*$, the J-equivalence class of u, namely $[u]_J$ induces the following two I_1 equivalence classes over the above set:

$$\begin{aligned} F_u &= acb\,[u]_J\,\# \\ E_u &= bac\,[u]_J\,\# \;\cup\; abc\,([u]_J \cap L)\,\#. \end{aligned}$$

Observe that $F_u \subseteq \Pi_L$, hence any representative chosen from F_u always belongs to Π_L.

Assume $[u]_J \cap L \neq \emptyset$. Then, $abc\,([u]_J \cap L)\,\# \neq \emptyset$. This implies that for all $w \in bac\,[u]_J\,\#$

$$\exists w' \in abc\,([u]_J \cap L)\,\# : \; w \equiv_{I_1} w' \;\wedge\; w' \prec w$$

Hence any representative chosen by $\mathrm{red}^{I_1}_{\leq}$ from E_u must belong to $abc\,([u]_J \cap L)\,\#$ and hence to Π_L.

Assume $[u]_J \cap L = \emptyset$. Then $E_u = bac\,[u]_J\,\#$, and its lexicographically least representative, which starts with bac, cannot belong to Π_L.

Therefore the inclusion on the left-hand side of Equation 1 holds exactly when $\mathrm{cl}_J(L) = \Gamma^*$. □

Hence, we can conclude:

Theorem 3.3 (Undecidability of RedStratification Union Inclusion Check). *There exist a finite alphabet Σ, regular program and proof languages $P, \Pi \subseteq \Sigma^*$, commutativity relations I_1, I_2, and a (flat) lexicographic order $\preceq$ such that the* (RedStratification Union Inclusion) *check is undecidable.*

Note that the result remains true under the premise that P is closed under both relations, $\Pi \subseteq P$, and (I_1, I_2) is stratifiable.

3.2 Linking The Results Back to The Setup of [1]

First, we observe that the proof language used in the reduction construction satisfies the constraints used in [1].

Proposition 3.4. *For every regular L, the language Π_L can be presented as a deterministic proof automaton with exactly one accepting state, and the sequence (I_1, I_2) can be made stratifiable in the sense of Definition 5.4 of [1].*

Proof. Since every word in Π_L ends in the fresh symbol #, construct a DFA for the prefix language $abcL \cup acb\Gamma^*$, send every successful #-transition to one common accepting state, and send all missing transitions to a rejecting sink. Transitions out of the accepting state also go to the sink. The result is deterministic and has exactly one accepting state.

To realize it as a proof automaton, interpret every letter as a syntactically distinct no-op statement. Label the DFA states by syntactically distinct tautologies, and use tautologies as pre- and postconditions. Every transition then denotes a valid Hoare triple. Under this interpretation every word is correct, so

$$\mathrm{cl}_{I_2}\big(\mathrm{cl}_{I_1}(\Pi_L)\big) \subseteq \Sigma^* = \text{correct}.$$

Thus (I_1, I_2) is stratifiable. Together with Lemma 3.1, this gives the premises used in the stated finite-state input model. □

The regular language P is also the trace language of a fixed bounded-thread program. The program runs $a; c$ in parallel with b, producing exactly the three prefixes bac, abc, acb. After a join, it runs three nondeterministically terminating loops x^*, for any $x \in \Gamma$ in parallel; their shuffles produce Γ^*. After a second join, it executes #. All pairs in $I_1 \cup I_2$ can therefore be chosen between statements of different threads.

4 UNDECIDABILITY OF RedStratification Inclusion CHECK

For proving the undecidability of this inclusion

$$\mathrm{red}_{\preceq}^{I_1}\left(\mathrm{red}_{\preceq}^{I_2}(P)\right) \subseteq \Pi$$

we use contextual lexicographical orders [2]. More precisely, we show that there are choices for a finite alphabet Σ°, regular program and proof languages $P^\circ, \Pi^\circ \subseteq (\Sigma^\circ)^*$, commutativity relations I_1°, I_2° (under which P° is closed), where the inclusion becomes undecidable.

4.1 The Reduction from a Universality Check

Let $a, b, d, e, \#$ be fresh alphabet symbols (i.e. disjoint from Γ) and let

$$\Sigma^\circ = \Gamma \cup \{a, b, d, e, \#\}.$$

Define the program and proof languages (through regular expressions) as

$$P^\circ = (ab \cup ba)\,\Gamma^*\,(de \cup ed)\,\#,$$
$$\Pi^\circ = ab\,\Gamma^*\,(de \;\cup\; ed)\,\#.$$

Define

$$I_1^\circ = J \cup \{(a, b), (b, a)\},$$
$$I_2^\circ = \{(d, e), (e, d)\}.$$

Lemma 4.1. *The language P° is closed under both I_1° and I_2°.*

Proof. Trivial by construction and the fact that nothing across $\{a, b, d, e, \#\}$ and Γ commutes. □

We define a *contextual* alphabet order [2] as follows:

- At ϵ, let $a < b$.
- After a prefix abu, for any $u \in \Gamma^*$, let

$$d < e \iff \delta(q_0, u) \in F \iff u \in L.$$

- After a prefix abu, when $u \notin L$, let $e < d$.
- After a prefix bau, for any $u \in \Gamma^*$, let $d < e$.

Extend these comparisons arbitrarily to total orders on Σ° at every context. We refer to the lexicographical word order imposed by this contextual order as $\prec^\circ$ in the rest of this argument.

Before, we use the order, let us make an observation that this order is in some precise sense *finite-state* representable, or an instance of what [1] calls a *positional order*.

Proposition 4.2. *$\prec^\circ$ is a* positional order *in the sense of [1].*

Proof. Let $\mathcal{A} = (Q, \Gamma, \delta, q_0, F)$ be the (minimal) DFA recognizing the source language L. Construct a deterministic finite automaton $\mathcal{A}_{P,L}$ for P° which, during the Γ^*-phase, remembers both the initial branch ab versus ba and the current state $\delta(q_0, u)$ of $\mathcal{A}$. This product observer does not change the accepted language:

$$L(A_{P,L}) = P^\circ.$$

We can then assign the order as specified above to the states of this automaton in a trival way. □

We are now ready to formulate the reduction.

Lemma 4.3 (Inner reduction). *We have*

$$\mathrm{red}_{\prec^\circ}^{I_2^\circ}(P^\circ) = ba\Gamma^* de\# \cup abLde\# \cup ab\overline{L}ed\#.$$

$\overline{L}$ is the standard notation for $(\Gamma^* \setminus L)$.

Proof. This is a straightforward consequence of the definition of $\prec^\circ$ and I_2. $\square$

Lemma 4.4 (Outer pure reduction). *For every regular $L \subseteq \Gamma^*$,*

$$\mathrm{red}^{I_1^\circ}_{\preceq^\circ}\left(\mathrm{red}^{I_2^\circ}_{\preceq^\circ}(P^\circ)\right) \subseteq \Pi^\circ \quad \Longleftrightarrow \quad \mathrm{cl}_J(L) = \Gamma^*. \tag{2}$$

Proof. Recall that

$$\mathrm{red}^{I_2^\circ}_{\prec^\circ}(P^\circ) = ba\Gamma^* de\# \cup abLde\# \cup ab\overline{L}ed\#.$$

and we are interested in I_1-equivalence classes of it. Since e, d do not commute under I_1, we safely partition the words from the above set into the ones ending in de or ed.

First, the words ending in ed from the inner language have the form

$$D_u = ab([u]_J \setminus L)ed\#.$$

Every element of D_u begins with ab. Therefore, every representative (from their I_1 equivalence classes) also starts with ab and therefore is in Π°. Therefore, these words cannot falsify the inclusion on the right-hand side of Equation 2.

Second, let us focus on the words ending in de. Define

$$E_u = ba[u]_J de\# \cup ab([u]_J \cap L)de\#.$$

If $[u]_J \cap L \neq \emptyset$, then similar to proof of Lemma 3.2, the words from the second term of E_u win in the lexicographical order over the ones from the first term. Hence, any representative from this class starts with ab and belongs to Π°.

If $[u]_J \cap L = \emptyset$, then $E_u = ba[u]_J de\#$, whose least element begins with ba and therefore does not belong to Π°.

$\square$

Hence, we can conclude:

Theorem 4.5 (Undecidability of The Inclusion in RedStratification Inclusion). *There exist a finite alphabet Σ, regular program and proof languages $P, \Pi \subseteq \Sigma^*$, commutativity relations I_1, I_2, and a (contextual) lexicographic order $\preceq$ such that the* (RedStratification Inclusion) *check is undecidable.*

REFERENCES

[1] Azadeh Farzan, Dominik Klumpp, and Andreas Podelski. Stratified commutativity in verification algorithms for concurrent programs. *Proceedings of the ACM on Programming Languages*, 7(POPL), Article 49, 2023. doi:10.1145/3571242.

[2] Azadeh Farzan and Anthony Vandikas. Reductions for Safety Proofs. *Proc. ACM Program. Lang.* 4, POPL (Jan. 2020), 1–28. doi:10.1145/3371081

[3] Alan M. Gibbons and Wojciech Rytter. On the decidability of some problems about rational subsets of free partially commutative monoids. *Theoretical Computer Science*, 48:329–337, 1986. doi:10.1016/0304-3975(86)90101-5.

[4] Dominik Klumpp. From Commutativity to Proofs and Back Again: Reduction and Verification of Concurrent Programs. *PhD Thesis*, February 2026. University of Freiburg, Germany. .