Decision procedure for string constraints involving the integer data type

Abstract. In this note, we consider straight-line string constraints involving string and integer data types. We propose semantic conditions and a generic decision procedure for the path feasibility of the symbolic execution of programs satisfying the semantic conditions. Furthermore, we show that common string operations, including concat, replaceall, transducers, reverse, substring, indexof, and length, satisfy the semantic conditions. Our approach is based on a variant of cost register automata.

1 Introduction

2 Preliminaries

For $n \in \mathbb{N}$ with $n \geq 1$, we use $[n]$ to denote $\{1, \cdots, n\}$.

A string over Σ is a (possibly empty) sequence of elements from Σ. Let $w = a_1 \cdots a_n$ be a string. The reserve of w, denoted by $w^{(r)}$, is $a_n \cdots a_1$.

We consider two data types, the string data type and the integer data type. We will use $c, d, \dots$ to denote integer constants, $u, v, \dots$ to denote string constants, $i, j, \dots$ to denote the integer variables, and $x, y, \dots$ to denote the string variables.

A finite automaton (FA) $\mathcal{A}$ is a tuple $(Q, \Sigma, \delta, I, F)$, where Q is a finite set of states, Σ is a finite alphabet, $\delta \subseteq Q \times \Sigma \times Q$ is the transition relation, $I, F \subseteq Q$ are the set of initial and final states respectively. A string $w = \sigma_1 \cdots \sigma_n$ is accepted by $\mathcal{A}$ if there is a state sequence $q_0 \cdots q_n$ such that $q_0 \in I$, $q_n \in F$, and $(q_{i-1}, \sigma_i, q_i) \in \delta$ for each $i \in [n]$. In particular, an empty string ε is accepted by $\mathcal{A}$ if $I \cap F \neq \emptyset$. The language defined by $\mathcal{A}$, denoted by $\mathcal{L}(\mathcal{A})$, is defined as the set of strings accepted by $\mathcal{A}$.

A finite transducer (FT) T is a tuple $(Q, \Sigma, \delta, I, F)$, where Q is a finite set of states, Σ is a finite alphabet, δ is the transition relation, which is a finite subset of $Q \times \Sigma \times Q \times \Sigma^*$, $I, F \subseteq Q$ are the set of initial and final states respectively. For readability, we write a transition (q, σ, q', u) as $q \xrightarrow{\sigma, u} q'$. A run of T over a string $w = \sigma_1 \cdots \sigma_n$ is a state sequence of transitions $q_0 \xrightarrow{\sigma_1, u_1} q_1 \cdots q_n \xrightarrow{\sigma_n, u_n} q_n$. The run is accepting if $q_0 \in I$ and $q_n \in F$. The string $u_1 \cdots u_n$ is called the output of the run. We use $\mathcal{T}(T)$ to denote the set of pairs (w, u) such that there is an accepting run of T on w, with the output u.

A linear arithmetic formula ϕ is defined by the rules: $\phi ::= t\ o\ t \mid \neg\phi \mid \phi \vee \phi \mid \exists i.\ \phi$, where $o \in \{=, \neq, \leq, \geq, <, >\}$ and t is defined by the rules $t ::= i \mid c \mid ct \mid t + t$. For a quantifier free linear integer arithmetic formula ϕ that contains the free variables $i_1, \cdots, i_k$, we use $\mathcal{M}(\phi)$ to denote the set of models of ϕ, namely, $\{(c_1, \cdots, c_k) \mid \phi[c_1/i_1, \cdots, c_k/i_k] \text{ holds}\}$. An existential linear arithmetic formula is a linear arithmetic formula where all the existential quantifiers are under the scope of even number of negation symbols.

3 The logic SL$_{int}$

We consider two types of functions, string functions that return strings and integer functions that return integers. Specifically, we consider

- string functions $f(x_1, \boldsymbol{i_1}, \cdots, x_k, \boldsymbol{i_k})$, where f is of the arity $\Sigma^*\times\mathbb{Z}^{n_1}\times\cdots\times\Sigma^*\times\mathbb{Z}^{n_k} \rightarrow 2^{\Sigma^*}$, and
- integer functions $g(x_1, \boldsymbol{i_1}, \cdots, x_k, \boldsymbol{i_k})$, where g is of the arity $\Sigma^*\times\mathbb{Z}^{n_1}\times\cdots\times\Sigma^*\times\mathbb{Z}^{n_k} \rightarrow 2^{\mathbb{Z}}$.

Note that f and g can be nondeterministic.

We consider string constraints where the formulae are of the form $S \wedge A$ defined by the following rules,

$$\begin{array}{ll}
t & ::= i \mid c \mid g(x_1, \boldsymbol{t_1}, \cdots, x_k, \boldsymbol{t_k}) \mid ct \mid t + t, \\
S & ::= x := f(x_1, \boldsymbol{t_1}, \cdots, x_k, \boldsymbol{t_k}) \mid S\,; S, \\
A_r & ::= x \in \mathcal{A} \mid A_r \wedge A_r, \\
A_i & ::= t\ o\ t \mid A_i \wedge A_i \mid A_i \vee A_i, \\
A & ::= A_r \wedge A_i,
\end{array}$$

where f is a string function and g is an integer function, $\boldsymbol{t_j} = t_{j,1}, \cdots, t_{j,n_j}$ for each $j \in [k]$, $\mathcal{A}$ is a finite-state automaton, and $o \in \{=, \neq, \geq, \leq, >, <\}$.

The logic SL$_{int}$ is defined as straight-line fragment of the aforementioned string constraints, specifically, SL$_{int}$ is defined as the collection of the formulae $S \wedge A$ satisfying that S **is in single static assignment (SSA) form**. Note that in SL$_{int}$, the straight-line restriction is applied only on S, which contains only the assignments to string variables (but not integer variables). No restrictions are put on the integer constraints in A_i.

Example 1. The formula $x := y \cdot z \wedge y := \mathsf{substring}(y', \mathsf{indexOf}(x, c), j) \wedge y' \in (ab)^* \wedge z \in a^*cb^* \wedge j = 2\mathsf{indexOf}(x, c)$ belongs to SL$_{int}$.

In the next section, we specify the semantic conditions for SL$_{int}$ in order to achieve decision procedures. For this purpose, we need the concepts of cost-enriched regular languages and recognisable relations.

4 The semantic conditions

4.1 Cost-enriched regular languages and recognisable relations

A cost-enriched string is $(w, (n_1, \cdots, n_k))$ with a string w and $n_i \in \mathbb{Z}$ for all $i \in [k]$. A cost-enriched language L is a subset of $\Sigma^* \times \mathbb{Z}^k$ for some k. Note that all the cost-enriched strings in L have the same number of costs, namely k. A cost-enriched relation $\mathcal{R}$ is a subset of $\Sigma^* \times \mathbb{Z}^{k_1} \times \cdots \Sigma^* \times \mathbb{Z}^{k_l}$.

Definition 1 (Cost-enriched finite automata and regular languages). *A cost-enriched finite automaton (CEFA) $\mathcal{A}$ is a tuple $(Q, \Sigma, R, \delta, I, F)$ where Q, Σ, I, F are as in FA, $R = (r_1, \cdots, r_k)$ is a vector of (mutually distinct) registers, δ is the transition relation which is a finite set of tuples (q, σ, q', η) where $q, q' \in Q$, $\sigma \in \Sigma$, and $\eta : R \rightarrow \mathbb{Z}$ is* *the cost-update function. For convenience, we usually write $(q, \sigma, q', \eta) \in \Delta$ as $q \xrightarrow{\sigma,\eta} q'$. A cost-enriched string $(w, (n_1, \cdots, n_k)) \in \Sigma^* \times \mathbb{Z}^k$ with $w = \sigma_1 \cdots \sigma_m$ is accepted by $\mathcal{A}$ if there is a sequence of transitions $q_0 \xrightarrow{\sigma_1,\eta_1} q_1 \cdots q_{m-1} \xrightarrow{\sigma_m,\eta_m} q_m$ such that $n_i = \eta_1(r_i) + \cdots + \eta_m(r_i)$ for each $i \in [k]$. The set of cost-enriched strings accepted by $\mathcal{A}$ is denoted as $\mathcal{L}(\mathcal{A})$. A cost-enriched language $L \subseteq \Sigma^* \times \mathbb{Z}^k$ is called a cost-enriched regular language (CERL) if there is a CEFA $\mathcal{A}$ such that $L = \mathcal{L}(\mathcal{A})$.*

CEFA can be seen as a variant of CRA in [1], by allowing nondeterminism and disallowing partial final cost functions.

For a CEFA $\mathcal{A}$, we use $R(\mathcal{A})$ to denote the vector of registers occurring in $\mathcal{A}$. Moreover, for a CEFA $\mathcal{A}$ and a vector of integer variables $\boldsymbol{i}$ such that $R(\mathcal{A}) \cap \boldsymbol{i} = \emptyset$, we use $\mathcal{A}[\boldsymbol{i}/R(\mathcal{A})]$ to denote the CEFA obtained from $\mathcal{A}$ by replacing the registers in $R(\mathcal{A})$ with those from $\boldsymbol{i}$.

Given two CEFAs $\mathcal{A}_1 = (Q_1, \Sigma, R_1, \delta_1, I_1, F_1)$ and $\mathcal{A}_2 = (Q_2, \Sigma, \delta_2, R_2, I_2, F_2)$ with $R_1 \cap R_2 = \emptyset$, we define the product of $\mathcal{A}_1$ and $\mathcal{A}_2$, denoted by $\mathcal{A}_1 \times \mathcal{A}_2$, as $(Q_1 \times Q_2, \Sigma, R_1 \cup R_2, \delta, I_1 \times I_2, F_1 \times F_2)$ such that δ comprises the tuples $((q_1, q_2), \sigma, (q_1', q_2'), \eta)$ satisfying that $(q_1, \sigma, q_1', \eta_1) \in \delta_1$, $(q_2, \sigma, q_2', \eta_2) \in \delta_2$, and $\eta = \eta_1 \cup \eta_2$ for some η_1, η_2.

Definition 2 (LA-SAT w.r.t. CEFA). *Let $\mathcal{A}_1 = (Q_1, \Sigma, R_1, \delta_1, I_1, F_1)$, $\cdots$, $\mathcal{A}_k = (Q_k, \Sigma, R_k, \delta_k, I_k, F_k)$ be CEFAs and ϕ be a quantifier-free linear arithmetic formula whose free variables are from $R_1 \cup \cdots \cup R_k \cup X$ for some X such that $X \cap (R_1 \cup \cdots \cup R_k) = \emptyset$. Then ϕ is said to be satisfiable w.r.t. $\mathcal{A}_1, \cdots, \mathcal{A}_k$ if there are words $w_1, \cdots, w_k$ and an assignment function $\eta : R_1 \cup \cdots R_k \cup X \to \mathbb{Z}$ such that $(w_1, \eta(R_1)) \in \mathcal{L}(\mathcal{A}_1)$, $\cdots$, $(w_k, \eta(R_k)) \in \mathcal{L}(\mathcal{A}_k)$, and $\phi[\eta(R_1)/R_1, \cdots, \eta(R_k)/R_k, \eta(X)/X]$ holds.*

Note that in Definition 2, it may happen that $R_i \cap R_j \neq \emptyset$ for some $i, j \in [k]$ with $i \neq j$.

Theorem 1. *The LA-SAT w.r.t. CEFA problem is decidable.*

For the proof of Theorem 1, we state and prove the following lemma.

Lemma 1. *Let $\mathcal{A} = (Q, \Sigma, R, \delta, I, F)$ be a CEFA with $R = (r_1, \cdots, r_m)$. Then there is an existential linear arithmetic formula $\varphi_{\mathcal{A}}(r_1, \cdots, r_m)$ such that $\mathcal{M}(\varphi_{\mathcal{A}}) = \{(c_1, \cdots, c_m) \mid$ there exists w such that $(w, c_1, \cdots, c_m) \in \mathcal{L}(\mathcal{A})\}$.*

Proof. Let $\delta = \{\tau_1, \cdots, \tau_l\}$ such that $\tau_j = (p_j, \sigma_j, p_j', \eta_j)$ and $\eta_j(r_i) = c_{i,j}$ for each $j \in [l]$ and $i \in [m]$. According the results on FA, we know that for each pair of states $(q, q') \in I \times F$, an existential linear arithmetic formula $\varphi_{q,q'}(j_1, \cdots, j_l)$ can be computed in linear time such that $\mathcal{M}(\varphi_{q,q'})$ is the set of Parikh images of the transition sequences of $\mathcal{A}$ starting from q and ending at q'.

Then

$$\varphi_{\mathcal{A}}(r_1, \cdots, r_m) ::= \bigvee_{(q,q') \in I \times F} \exists j_1 \cdots \exists j_l. \left(\varphi_{q,q'}(j_1, \cdots, j_l) \wedge \bigwedge_{i \in [m]} r_i = \sum_{j \in [l]} c_{i,j} j_j \right).$$

□ □

Theorem 1. Let $\mathcal{A}_1 = (Q_1, \Sigma, R_1, \delta_1, I_1, F_1)$, $\cdots$, $\mathcal{A}_k = (Q_k, \Sigma, R_k, \delta_k, I_k, F_k)$ be CEFAs and ϕ be a quantifier-free linear arithmetic formula whose free variables are from $R_1 \cup \cdots \cup R_k \cup X$ for some X such that $X \cap (R_1 \cdots R_k) = \emptyset$. Suppose that for each $i \in [k]$, $R_i = (r_{i,1}, \cdots, r_{i,l_i})$. Then the satisfiability of ϕ w.r.t. $\mathcal{A}_1, \cdots, \mathcal{A}_k$ can be reduced to the satisfiability problem of the existential linear arithmetic formula $\phi \wedge \bigwedge\limits_{i \in [k]} \varphi_{\mathcal{A}_i}(r_{i,1}, \cdots, r_{i,l_i})$. □ □

Definition 3 (Cost-enriched recognisable relations). *A cost-enriched relation $\mathcal{R} \subseteq \Sigma^* \times \mathbb{Z}^{k_1} \times \cdots \times \Sigma^* \times \mathbb{Z}^{k_l}$ is a cost-enriched recognisable relation (CERR) if it is a finite union of products of cost-enriched regular languages, namely,*

$$\mathcal{R} = \bigcup_{i=1}^{n} L_{i,1} \times \cdots \times L_{i,l},$$

where $L_{i,1} \subseteq \Sigma^* \times \mathbb{Z}^{k_1}, \cdots, L_{i,l} \subseteq \Sigma^* \times \mathbb{Z}^{k_l}$ are CERL. A CEFA representation of $\mathcal{R}$ is a collection of CERA tuples $(\mathcal{A}_{i,1}, \cdots, \mathcal{A}_{i,l})_{i \in [n]}$ such that $\mathcal{L}(\mathcal{A}_{i,j}) = L_{i,j}$ for each $i \in [n]$ and $j \in [l]$.

4.2 The two semantic conditions

For specifying the semantic conditions, we introduce two additional concepts.

Definition 4 (CERR linear integer functions). *An integer function $g : \Sigma^* \times \mathbb{Z}^{k_1} \times \Sigma^* \times \mathbb{Z}^{k_l} \to 2^{\mathbb{Z}}$ is a CERR linear integer function if there is a pair $(\mathcal{R}, t)$ such that $\mathcal{R} \subseteq \Sigma^* \times \mathbb{Z}^{k_1+1} \times \Sigma^* \times \mathbb{Z}^{k_l+1}$ is a CERR and t a linear integer term over $r^{(1)}, \cdots, r^{(l)}$ such that for all $\boldsymbol{c_1} \in \mathbb{Z}^{k_1}, \cdots, \boldsymbol{c_l} \in \mathbb{Z}^{k_l}$, and $d_1 \in \mathbb{Z}, \cdots, d_l \in \mathbb{Z}$, it holds that $(w_1, (\boldsymbol{c_1}, d_1), \cdots, w_l, (\boldsymbol{c_l}, d_l)) \in \mathcal{R}$ iff $t[d_1/r^{(1)}, \cdots, d_l/r^{(l)}] \in g(w_1, \boldsymbol{c_1}, \cdots, w_l, \boldsymbol{c_l})$. For a CERR linear integer function g witnessed by the pair $(\mathcal{R}, t)$, a CEFA representation of g is a tuple $((\mathcal{A}_{i,1}, \cdots, \mathcal{A}_{i,l})_{i \in [n]}, t)$, where $(\mathcal{A}_{i,1}, \cdots, \mathcal{A}_{i,l})_{i \in [n]}$ is a CEFA representation of $\mathcal{R}$.*

Example 2. The string functions length and $\mathsf{indexOf}_u$ are CERR linear integer functions, whose CEFA representations can be found in Section 4.3.

Definition 5 (Cost enriched pre-image of CERL). *Suppose that $f : \Sigma^* \times \mathbb{Z}^{k_1} \times \cdots \times \Sigma^* \times \mathbb{Z}^{k_l} \to 2^{\Sigma^*}$ is a string function, $L \subseteq \Sigma^* \times \mathbb{Z}^n$ is a CERL, and $L = \mathcal{L}(\mathcal{A})$ for some CEFA $\mathcal{A} = (Q, R, \delta, I, F)$ where $R = (r_1, \cdots, r_n)$. Then the R-cost enriched pre-image of L under f is a pair $(\mathcal{R}, \boldsymbol{t})$ such that*

- *$\mathcal{R} \subseteq \Sigma^* \times \mathbb{Z}^{k_1+n} \times \cdots \times \Sigma^* \times \mathbb{Z}^{k_l+n}$,*
- *$\boldsymbol{t} = (t_1, \cdots, t_n)$ is a vector of linear integer terms where for each $i \in [n]$, t_i is a term over $\boldsymbol{r_i} = (r_i^{(1)}, \cdots, r_i^{(l)})$,*
- *and*
  $L = \{(f(w_1, \boldsymbol{c_1}, \cdots, w_l, \boldsymbol{c_l}), t_1[d_{1,1}/r_1^{(1)}, \cdots, d_{l,1}/r_1^{(l)}], \cdots, t_n[d_{1,n}/r_n^{(1)}, \cdots, d_{l,n}/r_n^{(l)}]) \mid (w_1, (\boldsymbol{c_1}, \boldsymbol{d_1}), \cdots, w_l, (\boldsymbol{c_l}, \boldsymbol{d_l})) \in \mathcal{R}\}$ (where $\boldsymbol{d_1} = (d_{1,1}, \cdots, d_{1,n})$, $\cdots$, and $\boldsymbol{d_l} = (d_{l,1}, \cdots, d_{l,n})$).

The R-cost enriched pre-image of L under f, say $(\mathcal{R}, \boldsymbol{t})$, is said to be CERR-definable if $\mathcal{R}$ is a CERR. If the R-cost enriched pre-image of L under f, say $(\mathcal{R}, \boldsymbol{t})$, is CERR-definable, then its CEFA representation is a tuple $((\mathcal{A}_{i,1}, \cdots, \mathcal{A}_{i,l})_{i\in[m]}, \boldsymbol{t})$, where $(\mathcal{A}_{i,1}, \cdots, \mathcal{A}_{i,l})_{i\in[m]}$ is a CEFA representation of $\mathcal{R}$.

Example 3. Let $\mathcal{A} = (Q, R, \delta, I, F)$. The R-cost enrichment of the pre-image of $\mathcal{L}(\mathcal{A})$ under substring is CERR-definable and its CEFA representation can be found in Section 4.3.

Now we are ready to state the two semantic conditions.

The 1st semantic condition. Each integer function g is a CERR linear integer function, moreover, a CEFA representation of g can be effectively computed from g.

The 2nd semantic condition. Each string function f satisfies that for each CERL L, the cost enriched pre-image of L under f, say $(\mathcal{R}, \boldsymbol{t})$, satisfies that $\mathcal{R}$ is a CERR, moreover, a CEFA representation can be effectively computed from f and L.

4.3 A string logic satisfying the semantic conditions

The string logic $\mathrm{SL}^{\dagger}_{int}$ defined by the following rules satisfies the two semantic conditions,

$$\begin{array}{rl} t & ::= i \mid c \mid \mathsf{length}(x) \mid \mathsf{indexOf}_u(x, i) \mid ct \mid t + t, \\ S & ::= x := y \cdot z \mid x := \mathsf{replaceAll}_{e,u}(y) \mid x := \mathsf{reverse}(y) \mid x := T(y) \mid \\ & \quad\; x := \mathsf{substring}(y, t_1, t_2) \mid S ; S, \\ A_r & ::= x \in \mathcal{A} \mid A_r \wedge A_r, \\ A_i & ::= t\ o\ t \mid A_i \wedge A_i \mid A_i \vee A_i, \\ A & ::= A_r \wedge A_i, \end{array}$$

where $u \in \Sigma^+$, e is a regular expression, T is a finite-state transducer, and $o \in \{=, \neq, \geq, \leq, >, <\}$.

In the following, we show that the integer and string operations in $\mathrm{SL}^{\dagger}_{int}$ satisfy the semantic conditions.

Let $\mathcal{A} = (Q, \Sigma, R, \delta, I, F)$ be a CEFA with $R = (r_1, \cdots, r_m)$.

Concatenation $x_1 \cdot x_2$.

Then $((\mathcal{A}_{I,q}, \mathcal{A}_{q,F})_{q\in Q}, \boldsymbol{t})$ is a CEFA representation of the R-cost enriched pre-image of $\mathcal{L}(\mathcal{A})$ under $\cdot$, where $\mathcal{A}_{I,q} = (Q, \Sigma, R^{(1)}, \delta^{(1)}, I, \{q\})$ and $\mathcal{A}_{q,F} = (Q, \Sigma, R^{(2)}, \delta^{(2)}, \{q\}, F)$ such that

- $R^{(1)} = (r_1^{(1)}, \cdots, r_m^{(1)})$, $R^{(2)} = (r_1^{(2)}, \cdots, r_m^{(2)})$,
- $\delta^{(1)}$ comprises the tuples (q, σ, q', η') satisfying that $(q, \sigma, q', \eta) \in \delta$ and for each $j \in [m]$, $\eta'(r_j^{(1)}) = \eta(r_j)$, similarly for $\delta^{(2)}$,

and $\boldsymbol{t} = (r_1^{(1)} + r_1^{(2)}, \cdots, r_m^{(1)} + r_m^{(2)})$.

Reverse $\mathsf{reverse}(x_1)$.

$(\mathcal{A}^{(r)}, (r_1^{(1)}, \cdots, r_m^{(1)}))$ is the CEFA representation of the R-cost enriched pre-image of $\mathcal{L}(\mathcal{A})$ under reverse, where $\mathcal{A}^{(r)}$ is $(Q, \Sigma, R, \delta', F, I)$ such that δ' comprises the set of tuples (q', σ, q, η) with $(q, \sigma, q', \eta) \in \delta$. Note that $\mathcal{L}(\mathcal{A}^{(r)}) = \{(w^{(r)}, \boldsymbol{c}) \mid (w, \boldsymbol{c}) \in \mathcal{L}(\mathcal{A})\}$.

Substring $\mathsf{substring}(x_1, i, j)$.

Intuitively, $\mathsf{substring}(x_1, i, j)$ returns the substring of x_1 starting from the position i and ending at the position j (assuming that $i < j$), with the letter at the position j excluded.

$(\mathcal{B}, (r_1^{(1)}, \cdots, r_m^{(1)}))$ is the CEFA representation of the R-cost enriched pre-image of $\mathcal{L}(\mathcal{A})$ under $\mathsf{substring}$, where $\mathcal{B} = (Q \times \{p_0, p_1, p_2\}, \Sigma, R', \delta', I \times \{p_0\}, F \times \{p_2\})$ such that $R' = (i, j, r_1^{(1)}, \cdots, r_m^{(1)})$ and δ' comprises

- the tuples $((q, p_0), \sigma, (q, p_0), \eta')$ such that $q \in I$ and $\eta' = \eta_0 \cup \{i \to 1, j \to 1\}$, where $\eta_0(r_j^{(1)}) = 0$ for each $j \in [m]$,
- the tuples $((q, p_0), \sigma, (q', p_1), \eta')$ such that $(q, \sigma, q', \eta) \in \delta$ and $\eta' = \eta^{(1)} \cup \{i \to 1, j \to 1\}$, where $\eta^{(1)}(r_j^{(1)}) = \eta(r_j)$ for each $j \in [m]$,
- the tuples $((q, p_1), \sigma, (q', p_1), \eta')$ such that $(q, \sigma, q', \eta) \in \delta$ and $\eta' = \eta^{(1)} \cup \{i \to 0, j \to 1\}$, where $\eta^{(1)}(r_j^{(1)}) = \eta(r_j)$ for each $j \in [m]$,
- the tuples $((q, p_1), \sigma, (q, p_2), \eta')$ such that $q \in F$ and $\eta' = \eta_0 \cup \{i \to 0, j \to 1\}$, where $\eta_0(r_j^{(1)}) = 0$ for each $j \in [m]$,
- the tuples $((q, p_2), \sigma, (q, p_2), \eta')$ such that $q \in F$ and $\eta' = \eta_0 \cup \{i \to 0, j \to 0\}$, where $\eta_0(r_j^{(1)}) = 0$ for each $j \in [m]$.

FT $T(x_1)$.

Let $T = (Q', \Sigma, \delta', I', F')$. Then $(\mathcal{B}, (r^{(1)}, \cdots, r_m^{(1)}))$ is the CEFA representation of the R-cost enriched pre-image of $\mathcal{L}(\mathcal{A})$ under T, where $\mathcal{B} = (Q \times Q', \Sigma, R^{(1)}, \delta'', I \times I', F \times F')$ such that $R^{(1)} = (r^{(1)}, \cdots, r_m^{(1)})$, δ'' comprises the tuples $((q_1, q_1'), \sigma, (q_2, q_2'), \eta')$ satisfying that $(q_1', \sigma, q_2', u) \in \delta'$ with $u = \sigma_1 \cdots \sigma_i$, and in $\mathcal{A}$, we have $p_1 \xrightarrow{\sigma_1, \eta_1} p_2 \cdots \xrightarrow{\sigma_i, \eta_i} p_{i+1}$ with $p_1 = q_1$ and $p_{i+1} = q_2$, and for each $j \in [m]$, $\eta'(r_j^{(1)}) = \eta_1(r_j) + \cdots + \eta_i(r_j)$.

ReplaceAll $\mathsf{replaceAll}_{e,u}(x)$.

Intuitively, $\mathsf{replaceAll}_{e,u}(x)$ is the string obtained by replacing every occurrence of e in x with the constant string u.

From the results in [2], we know that a FT $T_{e,u}$ can be constructed to simulate $\mathsf{replaceAll}_{e,u}$. Therefore, a CEFA representation of the R-cost enriched pre-image of $\mathcal{L}(\mathcal{A})$ under T can be constructed as in the FT case.

Length $\mathsf{length}(x_1)$.

$(\mathcal{B}, r^{(1)})$ is a CEFA representation of length, where $\mathcal{B} = (Q', \Sigma, R^{(1)}, \delta', I', F')$ such that $Q' = \{q_0'\}$, $I' = F' = \{q_0'\}$, $R^{(1)} = (r^{(1)})$, $\delta' = \{(q_0', \sigma, q_0', \eta) \mid \sigma \in \Sigma, \eta(r^{(1)}) = 1\}$.

IndexOf $\mathsf{indexOf}_u(x_1, i)$.

Suppose $u = \sigma_1 \cdots \sigma_j$. We use the concept of window profiles of positions w.r.t. u, which are elements of $\{\bot, \top\}^{j-1}$, to recognise the first occurrence of u after the position i.

For $\pi \in \{\bot, \top\}^{j-1}$ and $\sigma' \in \Sigma$, $upt(\boldsymbol{\pi}, \sigma')$ is the updated window profile after reading the letter σ', specifically, $upt(\boldsymbol{\pi}, \sigma') = \boldsymbol{\pi'}$ such that

- $\pi_1' = \top$ iff $\sigma' = \sigma_1$,
- for each $j' \in [j-2]$, $\pi_{j'+1}' = \top$ iff $\pi_{j'} = \top$ and $\sigma' = \sigma_{j'+1}$.

The set of window profiles of u, denoted by WP_u, is computed by setting $WP_0 := \{\perp^{j-1}\}$ and iterating the following procedure, until $WP_i = WP_{i+1}$:

$$WP_{i+1} := WP_i \cup \{upt(\boldsymbol{\pi}, \sigma') \mid \boldsymbol{\pi} \in WP_i, \sigma' \in \Sigma\}.$$

From the results in [2], we know that $|WP_u| \leq |u|$. Therefore, the aforementioned iteration terminates in at most $|u|$ steps.

Then $(\mathcal{B}, r^{(1)})$ is a CEFA representation of $\mathsf{indexOf}_u$, where $\mathcal{B} = (Q', \Sigma, R', \delta', I', F')$ such that $Q' = \{q'_0, q'_1\} \cup WP_u \cup WP_u \times [i]$, $R' = (i, r^{(1)})$, $I' = \{q'_0\}$, $F' = \{q'_1\}$, and δ' comprises

- the tuples $(q'_0, \sigma, q'_0, \eta)$ such that $\sigma \in \Sigma$, $\eta(i) = 1$, and $\eta(r^{(1)}) = 1$,
- the tuples $(q'_0, \sigma, \boldsymbol{\pi}, \eta)$ such that $\sigma \in \Sigma$, $\boldsymbol{\pi} = \theta\perp^{j-2}$ where $\theta = \top$ iff $\sigma = \sigma_1$, $\eta(i) = 1$, and $\eta(r^{(1)}) = 1$,
- the tuples $(\boldsymbol{\pi}, \sigma, upt(\boldsymbol{\pi}, \sigma), \eta)$ such that $\boldsymbol{\pi} \in WP_u$, $\sigma \in \Sigma$, $\pi_{j-1} = \perp$ or $\sigma \neq \sigma_j$, $\eta(i) = 0$, and $\eta(r^{(1)}) = 1$,
- the tuples $(\boldsymbol{\pi}, \sigma, (upt(\boldsymbol{\pi}, \sigma), 1), \eta)$ such that $\boldsymbol{\pi} \in WP_u$, $\sigma = \sigma_1$, $\pi_{j-1} = \perp$ or $\sigma \neq \sigma_j$, $\eta(i) = 0$, and $\eta(r^{(1)}) = 1$,
- the tuples $((\boldsymbol{\pi}, j'), \sigma, (upt(\boldsymbol{\pi}, \sigma), j'+1), \eta)$ such that $\boldsymbol{\pi} \in WP_u$, $j' \in [j-2]$, $\sigma = \sigma_{j'+1}$, $\pi_{j-1} = \perp$ or $\sigma \neq \sigma_j$, $\eta(i) = 0$, and $\eta(r^{(1)}) = 0$,
- the tuples $((\boldsymbol{\pi}, j-1), \sigma, q'_1, \eta)$ such that $\boldsymbol{\pi} \in WP_u$, $\sigma = \sigma_j$, $\eta(i) = 0$, and $\eta(r^{(1)}) = 0$,
- the tuples $(q'_1, \sigma, q'_1, \eta)$ such that $\sigma \in \Sigma$, $\eta(i) = 0$, and $\eta(r^{(1)}) = 0$.

5 Decision procedure

Let $S' := S$ and $A' := A$. Moreover, let $A'' := \mathsf{true}$. Then execute the following procedure to (partially) flatten the integer terms.

Step 1. Recursively apply the following transformation until $S' \wedge A'$ contains no more occurrences of integer functions: Select an occurrence of integer functions, say $g(x_1, \boldsymbol{t_1}, \cdots, x_k, \boldsymbol{t_k})$, such that *none* of $\boldsymbol{t_1}, \cdots, \boldsymbol{t_k}$ contains occurrences of integer functions, introduce a fresh integer variable i, let $S' \wedge A'$ be the formula obtained by replacing $g(x_1, \boldsymbol{t_1}, \cdots, x_k, \boldsymbol{t_k})$ with i, moreover, let $A'' := A'' \wedge i = g(x_1, \boldsymbol{t_1}, \cdots, x_k, \boldsymbol{t_k})$.

Step 2. It comprises the following two substeps.

1. For each occurrence of string functions in S', say $f(x_1, \boldsymbol{t_1}, \cdots, x_k, \boldsymbol{t_k})$, suppose $\boldsymbol{t_j} = (t_{j,1}, \cdots, t_{j,l_j})$ for each $j \in [k]$, introduce fresh integer variables $i_{j,j'}$ for $j \in [k]$ and $j' \in [l_j]$, replace $f(x_1, \boldsymbol{t_1}, \cdots, x_k, \boldsymbol{t_k})$ with $f(x_1, \boldsymbol{i_1}, \cdots, x_k, \boldsymbol{i_k})$ in S', where $\boldsymbol{i_j} = (i_{j,1}, \cdots, i_{j,l_j})$ for each $j \in [k]$, and let $A' := A' \wedge \bigwedge\limits_{j\in[k], j'\in[l_j]} i_{j,j'} = t_{j,j'}$.
2. For each occurrence of integer functions in A'', say $g(x_1, \boldsymbol{t_1}, \cdots, x_k, \boldsymbol{t_k})$, suppose $\boldsymbol{t_j} = (t_{j,1}, \cdots, t_{j,l_j})$ for each $j \in [k]$, introduce fresh integer variables $i_{j,j'}$ for $j \in [k]$ and $j' \in [l_j]$, replace $g(x_1, \boldsymbol{t_1}, \cdots, x_k, \boldsymbol{t_k})$ with $g(x_1, \boldsymbol{i_1}, \cdots, x_k, \boldsymbol{i_k})$ in A'', where $\boldsymbol{i_j} = (i_{j,1}, \cdots, i_{j,l_j})$ for each $j \in [k]$, and let $A' := A' \wedge \bigwedge\limits_{j\in[k], j'\in[l_j]} i_{j,j'} = t_{j,j'}$.

Step 3. Let $S := S'$ and $A := A'' \wedge A'$.

The aforementioned flattening procedure is a bit technical, for simplicity, we may assume that the integer terms are fully flattened, including the arithmetic operations.

Note that after the aforementioned flattening procedure, the resulting formula $S \wedge A$ satisfies the following property:

> The integer terms in all the occurrences of string and integer functions are integer variables, moreover, each integer variable occurs at most once in these string and integer functions. $(*)$

Therefore, in the sequel, we assume that $S \wedge A$ satisfies the property $(*)$.

Theorem 2. *Path feasibility of SL$_{int}$ satisfying the semantic conditions is decidable.*

Proof. In the following, we extend the generic decision procedure in [3], where NFA is replaced by CEFA.

Let $S \wedge A$ be an SL$_{int}$ formula (satisfying the property $(*)$).

For each occurrence of $i = g(x_1, \boldsymbol{i'_1}, \cdots, x_k, \boldsymbol{i'_k})$ in A with g an integer function, apply the following nondeterministic transformation to A:

> According to the 1st semantic condition, g is a CERR linear integer function and a CEFA representation of g, say $((\mathcal{A}_{j,1}, \cdots, \mathcal{A}_{j,k})_{j \in [m]}, t)$, can be computed effectively from g. Consider $((\mathcal{A}'_{j,1}, \cdots, \mathcal{A}'_{j,k})_{j \in [m]}, t')$, where $\mathcal{A}'_{j,1} = \mathcal{A}_{j,1}[\boldsymbol{i'_1}/R(\mathcal{A}_{j,1})], \cdots, \mathcal{A}'_{j,k} = \mathcal{A}_{j,k}[\boldsymbol{i'_k}/R(\mathcal{A}_{j,k})]$, and $t' = t[i^{(1)}/r^{(1)}, \cdots, i^{(k)}/r^{(k)}]$.
> Nondeterministically choose $j \in [m]$, and replace $i = g(x_1, \boldsymbol{i'_1}, \cdots, x_k, \boldsymbol{i'_k})$ by $x_1 \in \mathcal{A}'_{j,1} \wedge \cdots \wedge x_k \in \mathcal{A}'_{j,k} \wedge i = t'$ in A.

Note that after this transformation, $S \wedge A$ contains no occurrences of integer functions, moreover, as a result of the property $(*)$, for every variable x, all the CEFAs to which x belongs satisfy that their sets of registers are mutually disjoint.

Then repeat the following procedure until S becomes empty.

> Suppose $y := f(x_1, \boldsymbol{i_1}, \cdots, x_k, \boldsymbol{i_k})$ is the last assignment of S.
> Let $\rho := \{\mathcal{A}_1, \cdots, \mathcal{A}_s\}$ be the set of all CEFAs such that $y \in \mathcal{A}_j$ occurs in A for each $j \in [s]$. Construct $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_s$ (Recall that the sets of registers of $\mathcal{A}_1$, $\cdots$, $\mathcal{A}_s$ are mutually disjoint). Let the vector of registers in $\mathcal{A}$ be $R = (r'_1, \cdots, r'_n)$. Then according to the 2nd semantic condition, a CEFA representation of the R-cost enriched pre-image of $\mathcal{L}(\mathcal{A})$ under f, say $((\mathcal{B}_{j,1}, \cdots, \mathcal{B}_{j,k})_{j \in [\ell]}, \boldsymbol{t})$, can be effectively computed from $\mathcal{A}$ and f. Consider $((\mathcal{B}'_{j,1}, \cdots, \mathcal{B}'_{j,k})_{j \in [\ell]}, \boldsymbol{t'})$, where $\mathcal{B}'_{j,1} = \mathcal{B}_{j,1}[\boldsymbol{i_1}/R(\mathcal{B}_{j,1}), (\boldsymbol{r'})^{(\boldsymbol{1})}/\boldsymbol{r}^{(\boldsymbol{1})}], \cdots$, $\mathcal{B}'_{j,k} = \mathcal{B}_{j,k}[\boldsymbol{i_k}/R(\mathcal{B}_{j,k}), (\boldsymbol{r'})^{(\boldsymbol{k})}/\boldsymbol{r}^{(\boldsymbol{k})}]$ (with $\boldsymbol{r}^{(\boldsymbol{1})} = (r^1_1, \cdots, r^{(1)}_n)$, similarly for $\boldsymbol{r}^{(\boldsymbol{2})}$ and so on), and $\boldsymbol{t'} = \boldsymbol{t}[\boldsymbol{r'_1}/\boldsymbol{r_1}, \cdots, \boldsymbol{r'_n}/\boldsymbol{r_n}]$ (with $\boldsymbol{r_1} = (r^{(1)}_1, \cdots, r^{(k)}_1)$, similarly for $\boldsymbol{r}^{(\boldsymbol{2})}$ and so on).
> Nondeterministically choose $j \in [\ell]$ and let
>
> $$A := A \wedge x_1 \in \mathcal{B}'_{j,1} \wedge \cdots \wedge x_k \in \mathcal{B}'_{j,k} \wedge \bigwedge_{j' \in [n]} r'_{j'} = t'_{j'}.$$
>
> Remove $y := f(x_1, \boldsymbol{i_1}, \cdots, x_k, \boldsymbol{i_k})$ from S.

We would like to remark that if all the string functions f in $S \wedge A$ are *deterministic*, then the product of CEFAs before the pre-image computation can be avoided and the pre-image can be computed *distributively* for CEFAs in ρ.

In the end, we get a formula $S \wedge A$ where S is empty. Suppose $A = A_r \wedge A_i$, where A_r is a conjunction of atomic formulae of the form $x \in \mathcal{A}$, and A_i is linear arithmetic formula (containing no integer functions). By computing the product construction of CEFAs, A_r can be rewritten as $x_1 \in \mathcal{A}_1 \wedge \cdots \wedge x_n \in \mathcal{A}_n$, where $x_1, \cdots, x_n$ are mutually distinct. Therefore, the path feasibility of $S \wedge A$ is exactly the satisfiability of A_i w.r.t. the CEFAs $\mathcal{A}_1, \cdots, \mathcal{A}_n$. From Theorem 1, we conclude that the path feasibility of SL_{int} is decidable. □ □

Corollary 1. *Path feasibility of $SL^{\dagger}_{int}$ is decidable.*

References

1. R. Alur, L. D'Antoni, J. Deshmukh, M. Raghothaman, and Y. Yuan. Regular functions and cost register automata. In *Proceedings of the 2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '13, pages 13–22. IEEE Computer Society, 2013.
2. T. Chen, Y. Chen, M. Hague, A. W. Lin, and Z. Wu. What is decidable about string constraints with the replaceall function. *PACMPL*, 2(POPL):3:1–3:29, 2018.
3. T. Chen, M. Hague, A. W. Lin, P. Rümmer, and Z. Wu. Decision procedures for path feasibility of string-manipulating programs with complex operations. *CoRR*, abs/1811.03167, 2018.