Decision procedure for string constraints involving the integer data type

Anonymous Author(s)

Abstract

We consider straight-line string constraints involving string and integer data types, specifically, we consider string constraints including concat, replaceall, transducers, reverse, substring, indexof, and length. We design the decision procedure based on a variant of cost register automata.

Keywords keyword1, keyword2, keyword3

1 Introduction

As discussed, the whole point is to use INCRA in replace of FT

2 Preliminaries

Definition for NFA, NFT.

3 The logic SL$_{int}$

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.

3.1 The concrete version

SL$_{int}$ comprises all the formulae $S \wedge A$ defined by the following rules,

$$
\begin{array}{lll}
t & :: & i \mid c \mid \mathsf{length} x \mid \mathsf{indexOf}_u x, i \mid t \;\; t, \\
S & :: & i : t \mid x : \mathsf{substring} y, i, j \mid x : y \cdot z \mid x : \mathsf{replaceAll}_{e,u} y \mid \\
 & & x : \mathsf{reverse} y \mid x : T y \mid S ; S, \\
A & :: & x \in \mathcal{A} \mid t \; o \; t \mid A \wedge A,
\end{array}
$$

where c is an integer constant, e is a regular expression, u is a string constant, T is an NFT, $\mathcal{A}$ is an NFA, and $o \in \{, \neq, \leq, \geq, <, >\}$. Note that $\mathsf{replaceAll}_{e,u}$ is the replaceAll function where e and u are the pattern and the replacement arguments.

We assume that S is in single static assignment (SSA) form. Moreover, for technical convenience, we assume that all the assignments $i : t$ in S satisfy that $t \;\; \mathsf{length} x$, $t \;\; \mathsf{indexOf}_u x, j$, or t contains no occurrences of length or $\mathsf{indexOf}$ functions. We also assume that all the variables in A also occur in S.

3.2 The abstract version

To be more abstract, we consider string formulae $S \wedge A$, where A is as above and S is defined by the following rules,

$$
\begin{array}{lll}
t & :: & i \mid c \mid g x_1, i_1, \cdots, x_k, i_k \mid t \;\; t, \\
S & :: & i : t \mid x : f x_1, i_1, \cdots, x_k, i_k \mid S ; S,
\end{array}
$$

where $i_j \;\; i_{j,1}, \cdots, i_{j,n_j}$ for each $j \in k$, f is of the arity $\Sigma^* \times \mathbb{Z}^{n_1} \times \cdots \times \Sigma^* \times \mathbb{Z}^{n_k} \to 2^{\Sigma^*}$ and g is of the arity $\Sigma^* \times \mathbb{Z}^{n_1} \times \cdots \times \Sigma^* \times \mathbb{Z}^{n_k} \to 2^{\mathbb{Z}}$ (note that f resp. g can be nondeterministic).

We assume that S is in single static assignment (SSA) form. Moreover, for technical convenience, we assume that all the assignments $i : t$ in S satisfy that either $t \;\; g x_1, i_1, \cdots, x_k, i_k$, or t contains no occurrences of the functions $g x_1, i_1, \cdots, x_k, i_k$. We also assume that all the variables in A also occur in S.

4 Incremental nondeterministic cost register automata (INCRA)

Definition 4.1. An Incremental nondeterministic cost register automaton (INCRA) $\mathcal{A}$ is a tuple $\Sigma, Q, I, F, R, \delta$, where Σ is a finite alphabet, Q is a finite set of states, $I \subseteq Q$ is a set of initial states, F is a set of final states, $R \;\; r_1 \cdots r_m$ is a vector of registers, δ comprises the tuples q, σ, q', η such that η is the update function satisfying that for each $r \in R$, $\eta r \;\; r \;\; c$ for some integer constant c. For readability, we write a transition q, σ, q', η as $q \xrightarrow{\sigma, \eta} q'$.

Note that for the purpose of this note, INCRA are adapted from CRA in [**?**] by allowing the nondeterminism and discarding the partial final cost function μ.

Let $\mathcal{A} \;\; \Sigma, Q, I, F, R, \delta$ be an INCRA with $R \;\; r_1 \cdots r_m$. Over an input word $w \;\; \sigma_1 \cdots \sigma_n \in \Sigma$, a run of $\mathcal{A}$ on w is a sequence $q_0 \xrightarrow{\sigma_1, \eta_1} q_1 \cdots q_{n-1} \xrightarrow{\sigma_n, \eta_n} q_n$ such that $q_0 \in I$ and $q_{i-1}, \sigma_i, \eta_i, q_i \in \delta$ for each $i \in n$. A run is accepting if $q_n \in F$. The output of an accepting run of $\mathcal{A}$ on w is a tuple $i_1, \cdots, i_m$, where $i_j \;\; \eta_n r_j \cdots \eta_1 r_j 0 \cdots$ for each $j \in m$. Note that the initial value of each register r_j is zero. We define $\mathcal{A} w$ as the set of outputs of the accepting runs of $\mathcal{A}$ on w (possibly it is an empty set). Note the in general, an output of an INCRA is a tuple, instead of a single integer. Moreover, we also use $\mathcal{L} \mathcal{A}$ to denote $\{w \in \Sigma^* \mid \mathcal{A} w \neq \emptyset\}$ and $\mathcal{R} w \;\; \{w, n \mid n \in \mathcal{A} w\}$.

Given two INCRAs $\mathcal{A}_1 \;\; \Sigma, Q_1, I_1, F_1, R_1, \delta_1$ and $\mathcal{A}_2 \;\; \Sigma, Q_2, I_2, F_2, R_2, \delta_2$ 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 $\Sigma, Q_1 \times Q_2, I_1 \times I_2, F_1 \times F_2, R_1 \cdot R_2, \delta$ such that δ comprises the tuples $q_1, q_2, \sigma, q'_1, q'_2, \eta$ such that there are η_1, η_2 satisfying that $q_1, \sigma, q'_1, \eta_1 \in \delta_1$, $q_2, \sigma, q'_2, \eta_2 \in \delta_2$, and $\eta \;\; \eta_1 \cup \eta_2$.

Given an INCRA $\mathcal{A} \;\; \Sigma, Q, I, F, R, \delta$, the inverse of $\mathcal{A}$, denoted by $\mathcal{A}^r$, is $\Sigma, Q, F, I, R, \delta'$ where δ' comprises the set of tuples q', σ, q, η such $q, \sigma, q', \eta \in \delta$. Note that according to the definition, over each word w, $\mathcal{A}^r w \;\; \mathcal{A} w$.

Definition 4.2 (INCRA-LA SAT)**.** Let $\mathcal{A}_1 \;\; \Sigma, Q_1, I_1, F_1, R_1, \delta_1, \cdots, \mathcal{A}_k \;\; \Sigma, Q_k, I_k, F_k, R_k, \delta_k$ be INCRAs with $R_i \cap R_j \;\; \emptyset$ for each $i \neq j \in k$ and ϕ be a

PL'18, January 01–03, 2018, New York, NY, USA
2018.

(quantifier-free) linear arithmetic formula over $R_1 \cdot \cdots \cdot R_k$. 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 $c_1 \in \mathcal{A}_1 w_1, \ldots, c_k \in \mathcal{A}_k w_k$ such that $\phi c_1, \cdots, c_k$ holds.

Theorem 4.3. *The INCRA-LA SAT problem is decidable.*

For the proof of Theorem 4.3, we state and prove the following lemma.

Lemma 4.4. *Let $\mathcal{A}\ \ \Sigma, Q, I, F, R, \delta$ be an INCRA 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 $\{c_1, \cdots, c_m \mid \varphi_{\mathcal{A}} c_1, \cdots, c_m \text{ holds}\}$ $_{w \in \Sigma^*} \mathcal{A} w$.*

Proof. Let $\delta\ \ \{\tau_1, \cdots, \tau_l\}$ such that $\tau_j\ \ p_j, \sigma_j, p'_j, \eta_j$ and $\eta_j r_i$ $r_i\ \ c_{i,j}$ for each $j \in l$ and $i \in m$. For each pair of states $q, q' \in I \times F$, it is not hard to compute a Presburger arithmetic formula $\varphi_{q,q'}\, j_1, \cdots, j_l$ such that $\{c_1, \cdots, c_l \mid \varphi_{q,q'}\, c_1, \cdots, c_l \text{ holds}\}$ defines the Parikh image of the sequence of transitions of $\mathcal{A}$ starting from q and ending at q'.

Then

$$\varphi_{\mathcal{A}} :: \ {}_{q,q' \in I \times F}\ j_1 \cdots j_l.\ \varphi_{q,q'}\, j_1, \cdots, j_l \wedge {}_{i \in m}\ r_i\ \ {}_{j \in l}\ c_{i,j} j_j.$$

□ □

Theorem 4.3. Suppose for each $i \in k$, $R_i\ \ r_{i,1} \cdots r_{i,r_i}$. Then we reduce the INCRA-LA SAT problem to the satisfiability of the following existential linear arithmetic formula

$$\phi \wedge {}_{i \in k}\ \varphi_{\mathcal{A}_i} r_{i,1}, \cdots, r_{i,r_i}.$$

□ □

5 Decision procedure

Definition 5.1 (Pre-image of f)**.** Let $f x_1, i_1, \cdots, x_k, i_k : \Sigma^* \times \mathbb{Z}^{n_1} \times \cdots \times \Sigma^* \times \mathbb{Z}^{n_k} \to 2^{\Sigma^*}$ and $\mathcal{A}$ be an INCRA. Then the pre-image of f under $\mathcal{A}$, denoted by $f^{-1}\mathcal{A}$, is defined as

$$f^{-1}\mathcal{A}\ \ \{w_1, c_1, \cdots, w_k, c_k \mid w \in \Sigma^*.\ w \in f w_1, c_1, \cdots, w_k, c_k \text{ and } \mathcal{A} w \neq \emptyset\}.$$

Definition 5.2 (Cost-preserving INCRA-representation of pre-image of f)**.** Let $f x_1, i_1, \cdots, x_k, i_k : \Sigma^* \times \mathbb{Z}^{n_1} \times \cdots \times \Sigma^* \times \mathbb{Z}^{n_k} \to 2^{\Sigma^*}$ and $\mathcal{A}\ \ \Sigma, Q, I, F, R, \delta$ be an INCRA with $R\ \ r_1 \cdots r_m$. Then a cost-preserving representation of $f^{-1}\mathcal{A}$ is a pair $\mathcal{B}_{j,1}, \cdots, \mathcal{B}_{j,k\, j \in \ell}, t$ (where $\ell \geq 1$) such that

- for each $j \in \ell$ and $j' \in k$, $\mathcal{B}_{j,j'}$ $\Sigma, Q'_{j,j'}, I'_{j,j'}, F'_{j,j'}, R'_{j,j'}, \delta'_{j,j'}$ with $R'_{j,j'}\ \ i_{j'} \cdot r'_j$, where $r'_1\ \ r'_{1,1} \cdots r'_{1,m}, \cdots, r'_k\ \ r'_{k,1} \cdots r'_{k,m}$ are mutually distinct fresh registers,
- $t\ \ t_1, \cdots, t_m$ such that for each $j'' \in m$, $t_{j''}$ is a linear combination of $r'_{1,j''}, \cdots, r'_{k,j''}$,
- for each $w_1, c_1, \cdots, w_k, c_k \in f^{-1}\mathcal{A}$, we have

$$\begin{gathered}{}_{w \in f w_1, c_1, \cdots, w_k, c_k} \mathcal{A} w \\ \left\{ t d_1 r'_1, \cdots, d_k r'_k \mid w_1, c_1 \cdot d_1, \cdots, w_k, c_k \cdot d_k \in {}_{j \in \ell}\ \mathcal{R}\mathcal{B}_{j,1} \times \cdots \times \mathcal{R}\mathcal{B}_{j,k} \right\}\end{gathered}$$

Definition 5.3 (INCRA-representation of g)**.** Let $g x_1, i_1, \cdots, x_k, i_k : \Sigma^* \times \mathbb{Z}^{n_1} \times \cdots \times \Sigma^* \times \mathbb{Z}^{n_k} \to 2^{\mathbb{Z}}$ and r be a register. Then a representation of g w.r.t. r is a pair $\mathcal{B}_{j,1}, \cdots, \mathcal{B}_{j,k\, j \in \ell}, t$ such that

- for each $j \in \ell$ and $j' \in k$, $\mathcal{B}_{j,j'}$ $\Sigma, Q'_{j,j'}, I'_{j,j'}, F'_{j,j'}, R'_{j,j'}, \delta'_{j,j'}$ with $R'_{j,j'}\ \ i_{j'} \cdot r'_j$, where $r'_1, \cdots, r'_k$ are mutually distinct fresh registers,
- t is a linear combination of $r'_1, \cdots, r'_k$,
- for each $w_1, c_1, \cdots, w_k, c_k \in \Sigma^* \times \mathbb{Z}^{n_1} \times \cdots \times \Sigma^* \times \mathbb{Z}^{n_k}$, we have

$$\begin{gathered} g w_1, c_1, \cdots, w_k, c_k \\ \left\{ t d_1 r'_1, \cdots, d_k r'_k \mid w_1, c_1 \cdot d_1, \cdots, w_k, c_k \cdot d_k \in {}_{j \in \ell}\ \mathcal{R}\mathcal{B}_{j,1} \times \cdots \times \mathcal{R}\mathcal{B}_{j,k} \right\}\end{gathered}$$

Semantic conditions of SL_{int}. S satisfies the following two conditions,

- for each function f occurring in S, there is an effective procedure to compute for a given INCRA $\mathcal{A}$, a cost-preserving INCRA-representation of $f^{-1}\mathcal{A}$,
- for each function g occurring in S, there is an effective procedure to compute for a given register r, an INCRA-representation of g w.r.t. r.

Note that z $\mathsf{replaceAll}_e x, y$ does not satisfy the semantic conditions, since the length of z is nonlinear w.r.t. the lengths of x and y in general.

Theorem 5.4. *Satisfiability of abstract SL_{int} satisfying the semantic conditions is decidable.*

Proof idea: Backward computation. Record relationship between integer variables in A.

The decision procedure

Corollary 5.5. *Satisfiability of concrete SL_{int} is decidable.*

Let $\mathcal{A}\ \ \Sigma, Q, I, F, X, \delta$ be an INCRA.

- Then $\cdot^{-1}\mathcal{A}$ is defined as $\mathcal{A}_{I,q}, \mathcal{A}_{q,F\, q \in Q}$ where $\mathcal{A}_{I,q}$ $\Sigma, Q, I, \{q\}, X, \delta$ and $\mathcal{A}_{q,F}\ \ \Sigma, Q, \{q\}, F, X, \delta$.
- $\mathsf{reverse}^{-1}\mathcal{A}\ \ \mathcal{A}^r$.
- $\mathsf{substring}^{-1}\mathcal{A}$ is the INCRA $\Sigma, Q \times \{p_0, p_1, p_2\}, I \times \{p_0\}, F \times \{p_2\}, X \cup \{y_1, y_2\}, \delta'$ such that δ' comprises
  - the tuples $q, p_0, \sigma, q', p_0, \eta'$ such that $q, \sigma, q', \eta \in \delta$, $\eta'\ \ \eta \cup \{y_1 \to y_1\ \ 1, y_2 \to y_2\ \ 1\}$,
  - the tuples $q, p_0, \sigma, q', p_1, \eta'$ such that $q, \sigma, q', \eta \in \delta$, and $\eta'\ \ \eta \cup \{y_1 \to y_1\ \ 1, y_2 \to y_2\ \ 1\}$,
  - the tuples $q, p_1, \sigma, q', p_1, \eta'$ such that $q, \sigma, q', \eta \in \delta$, $\eta'\ \ \eta \cup \{y_1 \to y_1, y_2 \to y_2\ \ 1\}$,
  - the tuples $q, p_1, \sigma, q', p_2, \eta'$ such that $q, \sigma, q', \eta \in \delta$, and $\eta'\ \ \eta \cup \{y_1 \to y_1, y_2 \to y_2\ \ 1\}$,
  - the tuples $q, p_2, \sigma, q', p_2, \eta'$ such that $q, \sigma, q', \eta \in \delta$, $\eta'\ \ \eta \cup \{y_1 \to y_1, y_2 \to y_2\}$.
- Let $T\ \ \Sigma, Q', I', F', \delta'$ such that $\delta' \subseteq Q' \times \Sigma \times Q' \times \Sigma^*$. Then $T^{-1}\mathcal{A}\ \ \Sigma, Q \times Q', I \times I', F \times F', \delta''$ such that 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 $u\ \ \sigma_1 \cdots \sigma_i$, $p_1 \xrightarrow{\sigma_1, \eta_1} p_2 \cdots \xrightarrow{\sigma_i, \eta_i} p_{i1}$ with $p_1\ \ q_1$ and $p_{i1}\ \ q_2$, and $\eta'\ \ \eta_1 \circ \cdots \circ \eta_i$.

- Let $T_{e,u}$ be the NFT corresponding to $\mathsf{replaceAll}_{e,u}$. Then $\mathsf{replaceAll}_{e,u}^{-1}\mathcal{A}$ $T_{e,u}^{-1}\mathcal{A}$.

Moreover, we know that i : $\mathsf{length}x$ and i : $\mathsf{indexOf}_u x, j$ can be captured by INCRA.

Let S be a program where all the assignments i : t are flattened in the sense that they are of the form i : j c, or i : j j', or i : $\mathsf{length}x$, or i : $\mathsf{indexOf}_u x, j$. The decision procedure for the path feasibility of S goes backwards iteratively as follows:

- if the last assignment of the current program of the form neither i : j c nor i : j j' is x : $\mathsf{substring}y, i, j$, x : $y \cdot z$, x : $\mathsf{replaceAll}_{e,u}y$, x : $\mathsf{reverse}y$, or x : Ty, then construct the product INCRA $\mathcal{A}$ of all the INCRA for x, and replace the last assignment with $y \in \mathsf{substring}^{-1}\mathcal{A}$, or $y \in \mathcal{A}_{I,q}; z \in \mathcal{A}_{q,F}$ for some $q \in Q$, or $y \in \mathsf{replaceAll}_{e,u}^{-1}\mathcal{A}$, or $y \in \mathsf{reverse}^{-1}\mathcal{A}$, or $y \in T^{-1}\mathcal{A}$,
- if the last assignment of the current program of the form neither i : j c nor i : j j' is i : t such that t $\mathsf{length}x$ or t $\mathsf{indexOf}_u x, j$, then replace i : t with $y \in \mathcal{A}_{\mathsf{length},i}$, or $x \in \mathcal{A}_{\mathsf{indexOf}_u,i,j}$.

Then after the above procedure, we get a program S' where all the assignments are of the form i : j c or i : j j' and all the other statements are of the form $x \in \mathcal{A}$ for some INCRA $\mathcal{A}$. Then the path feasibility of S' is reduced to the INCRA LA-SAT problem. The decidability follows from Theorem 4.3.

A Appendix

Text of appendix ...