On the Satisfiability of Context-free String Constraints with Subword Ordering

C. Aiswarya
aiswarya@cmi.ac.in
Chennai Mathematical Institute
and CNRS IRL ReLaX
India

Soumodev Mal
soumodevmal@cmi.ac.in
Chennai Mathematical Institute
India

Prakash Saivasan
prakashs@imsc.res.in
The Institute of Mathematical Sciences
and CNRS IRL ReLaX
India

Context-free String Constraint with Subword Ordering

Defined over a set of Variables V and alphabet A,

It is a conjunction of

- Membership Constraints $x \in L$ ← Context-free
- Relational Constraints $x \preceq \mathtt{Shuffle}(y, y)$

Example

$V = \{x, y\}$ $\quad x \in a^n b^n c^*$ $\quad y \in a^* b^n a^n$

$A = \{a, b\}$ $\quad x \preceq y$ $\quad y \preceq x$

Solution $\quad x = y = a^n b^n c^n$

$bbabb \preceq \mathtt{Shuffle}(abab, baabaa)$

Satisfiability is Undecidable!

PCP instance over alphabet Σ:

Even for regular membership

Given two vector of strings U and V, each having n elements,

$\exists$ sequences of indices $i_1, i_2, \cdots, i_k$ such that $u_{i_1} \cdot u_{i_2} \cdots u_{i_k} = v_{i_1} \cdot v_{i_2} \cdots v_{i_k}$?

Regular String Constraint:

Variable set: $\{x_i, x_s, x_u, x_v\}$ Alphabet: $\Sigma \cup [n]$

captures the sequence of indices

captures the sequence of strings

Membership Constraints:

$x_i \in \left(\bigcup_{j \in [n]} j\right)^*$ $\qquad x_s \in \Sigma^*$

$x_u \in \left(\bigcup_{j \in [n]} j.u_j\right)^*$ $\qquad x_v \in \left(\bigcup_{j \in [n]} j.v_j\right)^*$

captures the index of each word by appending it as a prefix

Relational Constraints:

$x_i \preceq x_u$ $\qquad x_v \preceq \mathtt{Shuffle}(x_i, x_s)$

$x_s \preceq x_u$ $\qquad x_i \preceq x_v$

$x_u \preceq \mathtt{Shuffle}(x_i, x_s)$ $\qquad x_s \preceq x_v$

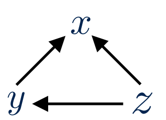
ensures that the string portion of X_u = string portion of X_v = X_s and the index portion of X_u = index portion of X_v = X_i

Acyclic Fragment

a subclass of our model where the relational constraints induce a DAG (partial order).

$x \preceq \mathtt{Shuffle}(y, z)$

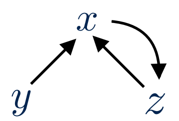
$y \preceq z$



Acyclic!

$x \preceq \mathtt{Shuffle}(y, z)$

$z \preceq x$



Cycle!

Theorem: Satisfiability of Acyclic String constraints is NEXPTIME complete!

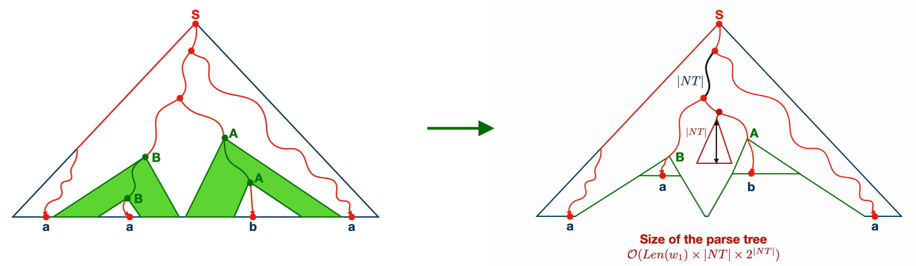
NEXPTIME Upperbound

If there is a satisfying assignment then there is a satisfying assignment of length bounded exponentially in the size of the input

Context-free

Lemma: if $\exists$ a word w_2 such that $w_1 \preceq w_2$ and $w_2 \in L$

then $\exists\, w_3 : w_1 \preceq w_3$, $w_3 \in L$ and w_3 has bounded size.

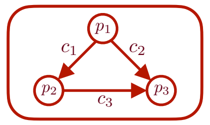


Parse tree of w_2 where grammar is in CNF

Parse tree of w_3

Acyclic Lossy Channel Pushdown Systems

Acyclic LPDS



Processes - Pushdown Systems

Lossy FIFO channels

[Atig et. al. 2008]

Mohamed Faouzi Atig, Ahmed Bouajjani, and Tayssir Touili. 2008. On the Reachability Analysis of Acyclic Networks of Pushdown Systems. In Concurrency Theory, 19th International Conference, CONCUR 2008, Toronto, Canada, August 19-22, 2008. Proceedings (Lecture Notes in Computer Science, Vol. 5201), Franck van Breugel and Marsha Chechik (Eds.). Springer. https://doi.org/10.1007/978-3-540- 85361- 9_29g

NEXPTIME Upper bound

A reduction from Acyclic LPDS to Acyclic String constraints

Variable set V

$\{\underbrace{x_{p_1}, x_{p_2}, x_{p_3}}_{\text{process for each process}}, \overbrace{x_{c_1}, x_{c_2}, x_{c_3}}^{\text{process for each channel}}\}$

Alphabet A

$\mathtt{Msgs} \times \{c_1, c_2, c_3\}$

message alphabet

tagged with the channel name

To convert a pushdown Transition System T_p for a process p in the Acyclic LCS to the membership constraint L_p for its corresponding variable in Acyclic String Constraints

For each transition in the Pushdown Transition System for a process p, we have a corresponding transition in the pushdown automata for the membership of the variable x_p as follows:

Pushdown Transition System	Membership Constraint
$s \xrightarrow{c?a} s'$	$s \xrightarrow{(a,c),\mathtt{nop}} s'$
$s \xrightarrow{c!a} s'$	$s \xrightarrow{(a,c),\mathtt{nop}} s'$
$s \xrightarrow{op} s'$	$s \xrightarrow{\epsilon, op} s'$

Note: $c?a$ - read "a" from channel c $\quad$ $c!a$ - write "a" from channel c
op - operations on stack i.e., `push()`, `pop()`, `nop`

Membership Constraints: $x_{p_i} \in L_{p_i}$ $\quad x_{c_i} \in B^*$

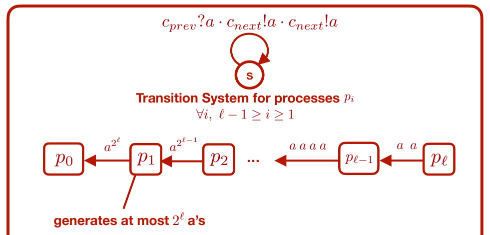
$\{$< letters marked by channels other than c_i >$\}$

Subword ordered Constraints: For each channel c

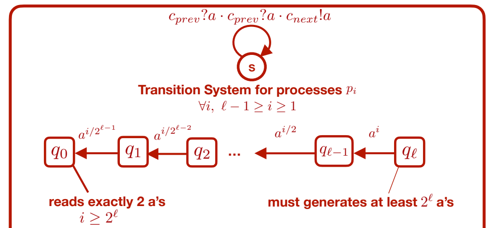
$x_p \preceq \mathtt{Shuffle}(x_q, x_c)$

where process p reads from channel c
process q writes on channel c

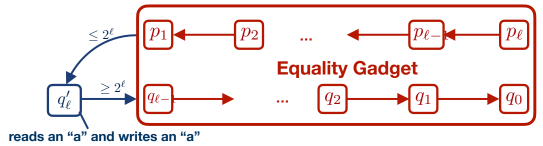
NEXPTIME lower bound

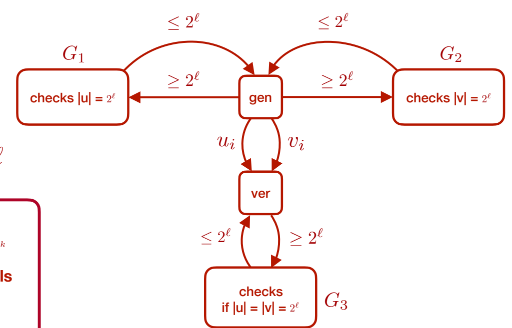


Gadget to count at most 2^ℓ



Gadget to count at least 2^ℓ





Topology for the PCP simulation

Reduction from a version of PCP instance bounded exponentially by an input integer ℓ

In one step, gen produces u_i and v_i for an index i in the two channels. It also ensures that the word $v_{i_1} \ldots v_{i_k}$ and $u_{i_1} \ldots u_{i_k}$ are of length 2^ℓ using the equality gadgets G_1 and G_2.
In one step, ver checks if the topmost letter in the two channels are equal or not. It also has access to an equality gadget G_3 that ensures that no message is lost while verifying their equality.

This implies that both the Satisfiability of Acyclic String Constraints and the Reachability of Acyclic Lossy Channel Pushdown Systems are NEXPTIME complete.

CHENNAI MATHEMATICAL INSTITUTE

ReLaX Research Lab in Computer Science