Mathematical Programming Modulo Strings

Ankit Kumar, Panagiotis Manolios
Northeastern University

FMCAD ‘21

DART: Directed Automated Random Testing

Patrice Godefroid Nils Klarlund Koushik Sen

A systematic analysis of XSS sanitization in web application frameworks

Paper Reference: Wei
web application framewo

Symbolic String Verification: Combining String

Saner: Com
Validat

An array-oriented language with static rank polymorphism

Davide l
E

er

CU Paros

Justin Slepak, Olin Shivers, and Panagiotis Manolios

Ahmed Rezine*, Philipp Rummer*, and Jari Stenman*

(Tools Pape

Koushik Sen and Gul
University of Illinois at Urbana-C

Symbolic String Verification: An Automata-based Approach

String Solving with Word Eq
Towards a Logic for Analysing

Fang Yu, Tevfik Bultan, Marco Cova, and Oscar H. Ibarra

Anthony W. Lin

Pablo Barceló

Nikolai Tillmann and Jonathan de Halleux

Quiz 1: Consider the example below, where

x, y are sequence variables

a,b are constants

Is this satisfiable?

$$xaxby = bybyx$$

From: Gutierrez. Solving Equations in Strings. 2006

Quiz 1: Consider the example below, where

x, y are sequence variables

a,b are constants

Is this satisfiable? Let $x = \epsilon$

$$xaxby = bybyx$$

$$aby = byby$$

From: Gutierrez. Solving Equations in Strings. 2006

Quiz 1: Consider the example below, where

x, y are sequence variables

a,b are constants

Is this satisfiable?

$$xaxby = bybyx$$

From: Gutierrez. Solving Equations in Strings. 2006

Quiz 1: Consider the example below, where

x, y are sequence variables

a,b are constants

Is this satisfiable? Let $x =$ b

$$xaxby = bybyx$$
$$babby = bybyb$$
$$abby = ybyb$$

From: Gutierrez. Solving Equations in Strings. 2006

Quiz 1: Consider the example below, where

x, y are sequence variables

a,b are constants

Is this satisfiable? Let $x = \text{b}, y = \epsilon$

$$\text{abb}y = y\text{b}y\text{b}$$

$$\text{abb} = \text{bb}$$

From: Gutierrez. Solving Equations in Strings. 2006

Quiz 1: Consider the example below, where

x, y are sequence variables

a,b are constants

Is this satisfiable? Let $x = \text{b}, y = \text{a}$

$$\text{abb}y = y\text{b}y\text{b}$$

$$\text{abba} = \text{abab}$$

From: Gutierrez. Solving Equations in Strings. 2006

Quiz 1: Consider the example below, where

x, y are sequence variables

a,b are constants

Is this satisfiable? Let $x =$ b, $y =$ ab

$$\text{abb}y = y\text{b}y\text{b}$$

$$\text{abbab} = \text{abbabb}$$

From: Gutierrez. Solving Equations in Strings. 2006

Quiz 1: Consider the example below, where

x, y are sequence variables

a,b are constants

Is this satisfiable?

$$xaxby = bybyx$$

From: Gutierrez. Solving Equations in Strings. 2006

Quiz 1: Consider the example below, where

x, y are sequence variables

a,b are constants

Is this satisfiable?

$$xaxby \quad bybyx$$

From: Gutierrez. Solving Equations in Strings. 2006

Quiz 1: Consider the example below, where

x, y are sequence variables

a,b are constants

Is xaxby=byb$y x$ satisfiable?

$$x\text{a}x\text{b}y$$

$$\text{b}y\text{b}yx$$

From: Gutierrez. Solving Equations in Strings. 2006

Quiz 1: Consider the example below, where

x, y are sequence variables

a,b are constants

Is xaxby=byb$y x$ satisfiable?

Decompose

$$x\text{a} \mid x\text{b}y$$

$$\text{b}y \mid \text{b}yx$$

From: Gutierrez. Solving Equations in Strings. 2006

Quiz 1: Consider the example below, where

x, y are sequence variables

a,b are constants

Is xaxby=byb$y x$ satisfiable?

Decompose

$$x\text{a}\ \ \text{b}y$$

$$x\text{b}y\ \ \text{b}yx$$

From: Gutierrez. Solving Equations in Strings. 2006

Quiz 1: Consider the example below, where

x, y are sequence variables

a,b are constants

Decompose

Is xaxby=byb$y x$ satisfiable?

$$x\mathrm{a} = \mathrm{b}y$$

$$x\mathrm{b}y = \mathrm{b}yx$$

From: Gutierrez. Solving Equations in Strings. 2006

Quiz 1: Consider the example below, where

x, y are sequence variables

a,b are constants

Is xaxby=byb$y x$ satisfiable?

Decompose

Rewrite

$$x\text{a} = \text{b}y$$

$$x\text{b}y = \text{b}yx$$

From: Gutierrez. Solving Equations in Strings. 2006

Quiz 1: Consider the example below, where

x, y are sequence variables

a,b are constants

Is xaxby=byb$y$$x$ satisfiable?

Decompose

Rewrite

$$x\mathrm{a} = \mathrm{b}y$$

$$x\mathrm{b}y = x\mathrm{a}x$$

From: Gutierrez. Solving Equations in Strings. 2006

Quiz 1: Consider the example below, where

x, y are sequence variables

a,b are constants

Is xaxby=byb$y$$x$ satisfiable?

$$x\mathrm{a} = \mathrm{b}y$$

$$\mathrm{b}y = \mathrm{a}x$$

Decompose

Rewrite

Trim

From: Gutierrez. Solving Equations in Strings. 2006

Quiz 1: Consider the example below, where

x, y are sequence variables

a,b are constants

Is xaxby=byb$y$$x$ satisfiable?

$$x\mathrm{a} = \mathrm{b}y$$

$$\mathrm{b}y = \mathrm{a}x$$

Decompose

Rewrite

Trim

ConstUnsat

From: Gutierrez. Solving Equations in Strings. 2006

Quiz 2: Consider the example below, where
u, v, w, x, y, z are sequence variables
a,b,c are constants

Is this satisfiable?

$$x\text{c}y\text{c}z\text{v}y\text{c}y\text{a} = y\text{ac}w\text{a}z\text{v}\text{b}u x$$

Too hard; let's ask existing string solvers. Just 6 variables, should be easy.

Quiz 2: Consider the example below, where
u, v, w, x, y, z are sequence variables
a,b,c are constants

Is this satisfiable?

$$x\text{c}y\text{c}z\text{v}y\text{c}y\text{a} = y\text{ac}w\text{a}z\text{v}\text{b}u x$$

Unsolved by CVC4, Norn, Z3Str2 and Z3Str3, even after 1,000 seconds

SeqSolve after <1 second

$$x\text{c}y\text{c}z\text{v}y\text{c}y\text{a} = y\text{ac}w\text{a}z\text{v}\text{b}u x$$

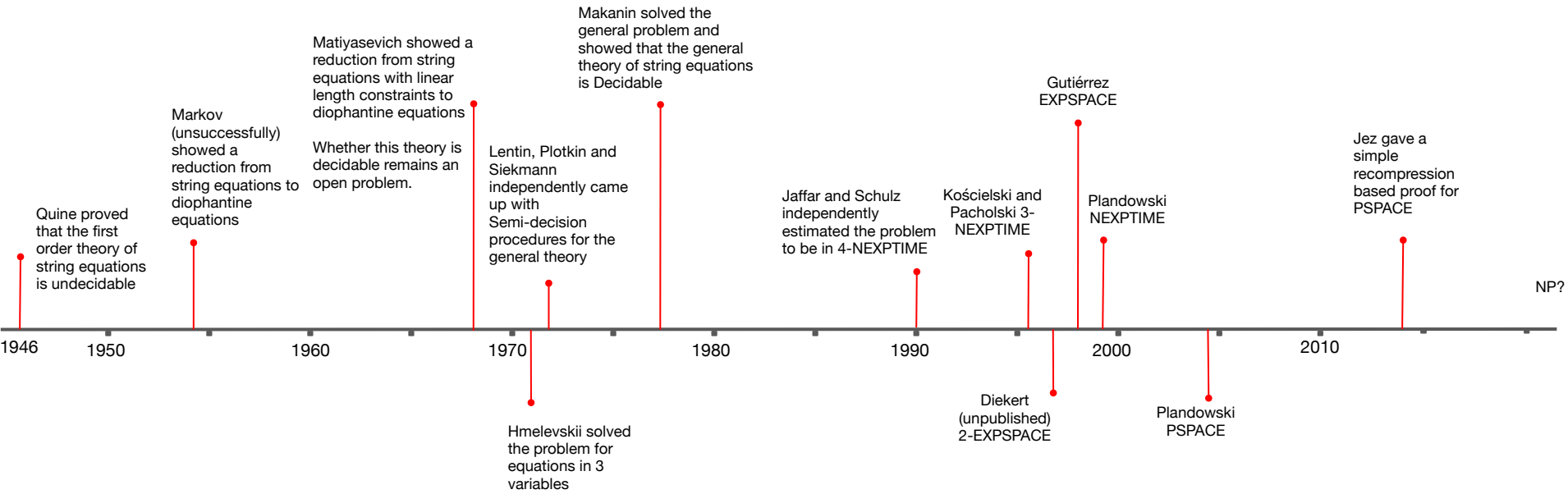
x = aba

y = ab

u = cabc

$v = w = z = \epsilon$

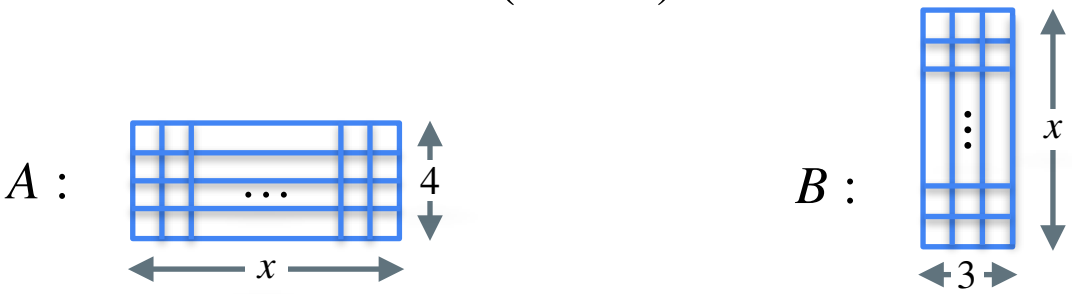
History of theoretical results on string solving



Motivation

Type inference in Remora, an Array Processing Language

$$(+\ A\ \ B)$$



$$x4 = 3x$$

J.Slepak, O.Shivers,and P.Manolios, “An array-oriented language with static rank polymorphism,” in European Symposium on Programming (ESOP), 2014.

Rules for solving sequence equations

$$\frac{\begin{array}{c}\{U{=}V,\ldots\} \qquad \langle \exists S,T :: SWT{=}U \wedge |W| > 1\rangle \\ \mathsf{Atoms}(W) \subseteq vars \qquad z \in \mathcal{X} \qquad z \notin vars \\ \mathsf{Atoms}(W) \cap \mathsf{Atoms}(\{U{=}V,\ldots\}\{W{:}z\}) = \emptyset\end{array}}{\begin{array}{c}\mathsf{LIANewVar}(z) \\ \sigma \leftarrow \sigma, W{:}z \\ \{U{=}V,\ldots\}\{W{:}z\}\end{array}}\ \mathsf{VarSubst}$$

$$\frac{\begin{array}{c}\{x^c(\alpha,l)U{=}(\beta,m)V,\ldots\} \qquad \alpha \neq \beta, c > 0 \\ x,y \in \mathcal{X} \qquad y \notin vars\end{array}}{\begin{array}{c}C_{len} \leftarrow C_{len},\ k > 0,\ (c-1) * k < m \leq c * k, \\ k < m \Rightarrow \epsilon_y = 1 \\ C_{words} \leftarrow C_{words},\ k < m \Rightarrow S^x_\beta = 1 \\ \{x{=}(\beta,k)y,\ x^c(\alpha,l)U{=}(\beta,m)V,\ldots\}\end{array}}\ \underline{\mathsf{VarSplit}}$$

Rules for solving sequence equations

Already seen Decompose, Rewrite, Trim and ConstUnsat

EqLength

$xa = x \rightarrow unsat$ as is $1 + |x| = |x|$

EqConsts

$ax = xb \rightarrow unsat$ as is $1 + n_a^x = n_a^x$

VarElim

$xa = ax, cy = x \rightarrow cya = acy \rightarrow unsat$

Rules for solving sequence equations

VarSplit

$xx\mathrm{a} = \mathrm{c}yx \rightarrow \mathrm{c}v\mathrm{c}v\mathrm{a} = \mathrm{c}y\mathrm{c}v \rightarrow v\mathrm{c}v\mathrm{a} = y\mathrm{c}v \rightarrow v\mathrm{c} = v, v\mathrm{a} = \mathrm{c}v \rightarrow unsat$

VarSubst

$wuzwuzwuz\mathrm{a}{=}\mathrm{c}ywuz \rightarrow xx\mathrm{a} = \mathrm{c}yx \rightarrow unsat$

EqWords

$x\mathrm{bca}y = y\mathrm{cba}x \rightarrow unsat$ (count of words in ca)

Transition System TranSeq

Accepts a conjunction of sequence equations Q_{init} and linear constraints C_{init} as input.

Generates an initial configuration K_{init}.

Applies enabled rules to configurations non-deterministically.

Generated linear constraints are incrementally pushed to a background Linear Integer Arithmetic (LIA) solver.

LIA solver is queried for satisfiability/decomposition.

Theoretically and practically complete for the general theory of string equations.

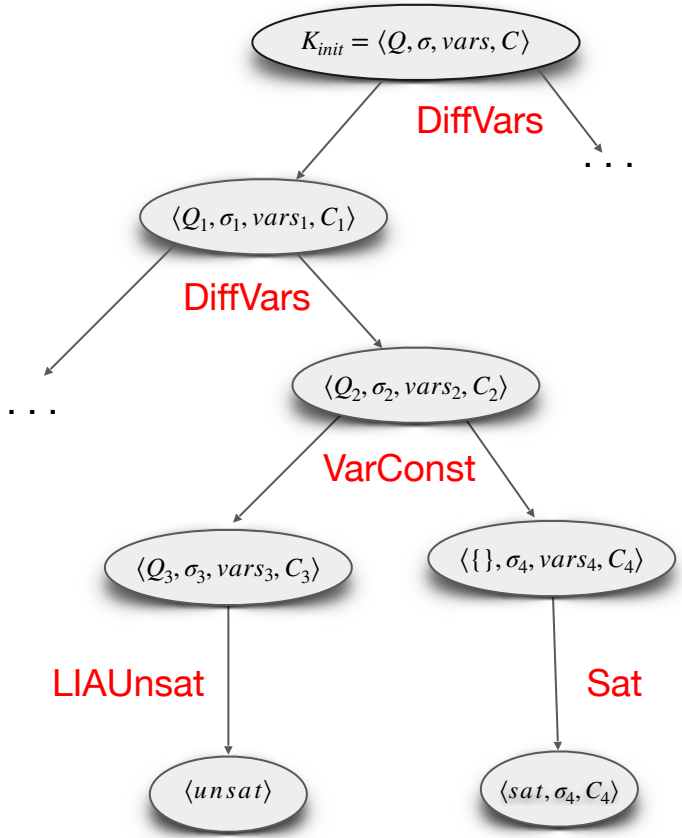
TranSeq based on Mathematical Programming Modulo Theories

MPMT [Manolios, Papavasileiou CAV2013] framework is used to combine decision procedures, like SMT, but instead of a SAT solving backend, utilizes an LIA solver.

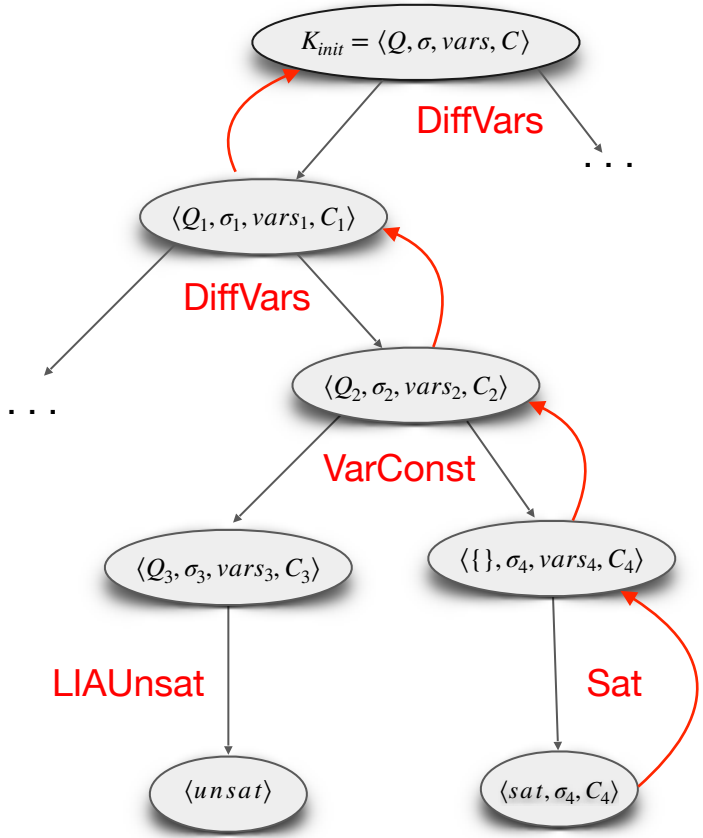
Highly non-deterministic system that allows branching and pruning of states while searching for a solution.

LIA solver communicates with background theory solver using interface variables.

Searching for a solution



Searching for a solution



Correctness

TranSeq returns either $sat, unsat$ or $unknown$. It is

1. Sound : never lies *i.e.*

   i) Returns $sat \Rightarrow$ satisfiable($Q_{init} \wedge C_{init}$) and returns a satisfying assignment

   ii) Returns $unsat \Rightarrow$ unsatisfiable($Q_{init} \wedge C_{init}$)

Correctness

TranSeq returns either sat, $unsat$ or $unknown$. It is

1. Sound : never lies
2. Partially Correct : sound and terminating but may return $unknown$

Termination depends on bounds on the lengths of minimal solutions of string equations by Plandowski. [STOC 1999]

Correctness

TranSeq returns either sat, $unsat$ or $unknown$. It is

1. Sound : never lies
2. Partially Correct : sound and terminating but may return $unknown$
3. Fully Correct in the absence of input linear constraints : partially correct and never returns $unknown$

Termination depends on bounds on the lengths of minimal solutions of string equations by Plandowski, STOC 1999

SeqSolve : Implementation of TranSeq

We implemented SeqSolve in ACL2s which allowed us to:

Easily define complex datatypes using defdata

```
(defdata a character)
(defdata svar (cons 'S symbol))

(defdata lexp (oneof nat
                     lvar
                     evar
                     (list op lexp lexp)))

(defdata block (cons a lexp))
(defdata elem (oneof a svar uvar block))
(defdata seq (listof elem))
(defdata sigma (alistof v seq))
(defdata eq (cons seq seq))
(defdata qs (listof eq))

(check= (seqp '(#\1 #\2 (#\1 . 1) (S . X))) t)
```

SeqSolve : Implementation of TranSeq

Rules on complex datatypes

```
(definecd varelim (qs :qs acc :qs sub :sigma) :qsig
  (match qs
    (() (cons acc sub))
    (((u . u) . qcs) (varelim qcs acc sub))
    (((u . v) . qcs)
     (cond ((and (endp (cdr u))
                 (vp (car u)))
            (varelim-help u v qs acc sub))
           ((and (endp (cdr v))
                 (vp (car v)))
            (varelim-help v u qs acc sub))
           (t (varelim qcs (cons (cons u v) acc) sub))))))
```

SeqSolve : Implementation of TranSeq

Define Z3 expression grammar

```
(defdata comp (oneof '= '<= '>= '< '>))

(defdata z3lexp (list comp lexp lexp))
```

SeqSolve : Implementation of TranSeq

Functions on complex datatypes

```
(definecd seqlen (seq :seq) :lexp
  (match seq
    ((c . d)
     (match c
       (() 0)
       (:svar `(+ ,(lvar c) ,(seqlen d)))
       (:uvar `(+ 1 ,(seqlen d)))
       (:block `(+ ,(cdr c) ,(seqlen d)))
       (& `(+ 1 ,(seqlen d)))))))
```

```
(definecd eq-len (u :seq v :seq) :z3lexp
  (list '= (seqlen u) (seqlen v)))
```

SeqSolve : Implementation of TranSeq

Interfacing with Z3, thanks to Andrew Walter's Z3-ACL2s interfacing library

```
(defun add-assertion (stmt)
  (let* ((lses (collect-vs stmt))
         (ls (first lses))
         (es (second lses)))
    (z3-assert-fn (solver->z3exp
                   (append (+int-vars-decl+ ls)
                           (+bool-vars-decl+ es)))
                  (solver->z3exp stmt))))
```

```
(solver-push)
(add-assertion (eq-len u v))
(check-sat)
(solver-pop)
```

Evaluation

Benchmark Problems

We evaluated our solver on all publicly available benchmarks we could find, restricted to the general theory of string equations and length constraints. Problems from Kaluza, StringFuzz and handcrafted benchmarks.

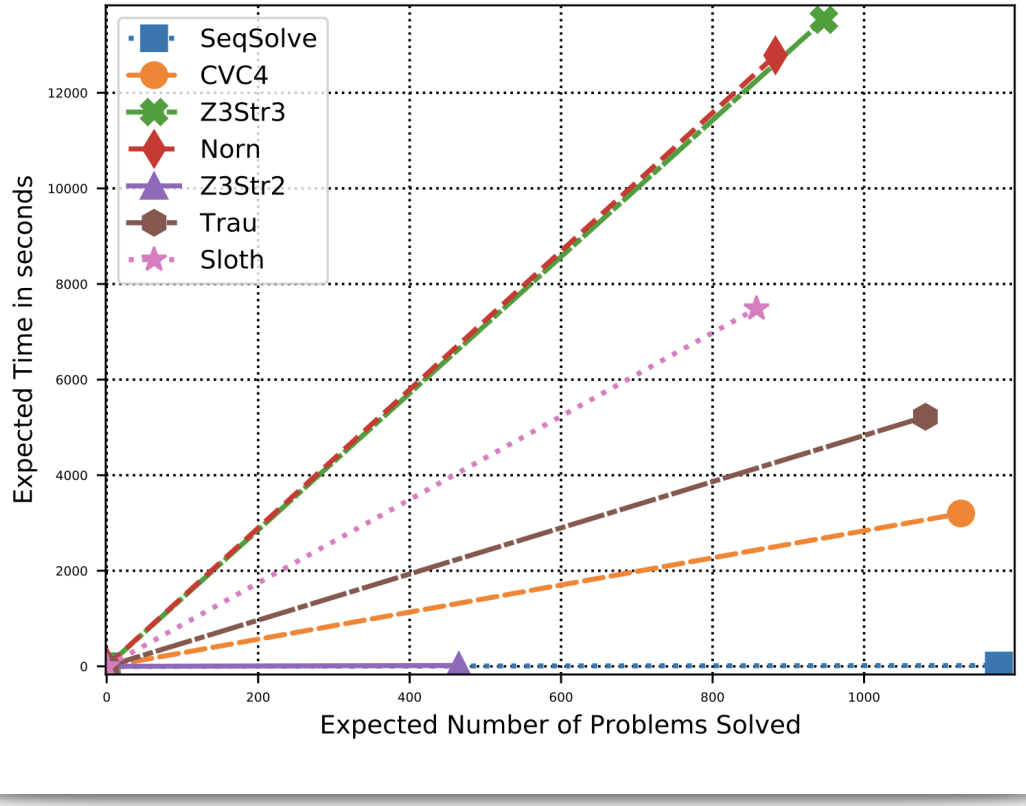
Tools

We compared our solver against all string solvers we know of that can solve problems in our theory of interest. These solvers are: CVC4, Norn, Z3Str2, Z3Str3, Sloth and TRAU.

Evaluation

Solver	Solved	Time (s)	Unknown	Timeout	X
SeqSolve	1,178: 780/344/54	176	0	0	0
CVC4	1,128: 736/344/48	3,200	0	50	0
Z3Str3	947: 552/344/51	13,527	6	225	0
Norn	883: 492/344/47	12,783	120	175	0
Z3Str2	465: 121/332/12	18	713	0	0
Trau	1,081: 692/344/45	5,223	18	78	1
Sloth	858: 462/344/52	7,486	0	319	64

Evaluation using Ray Plot



Conclusions and Future Work

- A new non-deterministic, branching transition system, TranSeq for deciding the satisfiability of conjunctions of string equations and length constraints.
- SeqSolve was faster and solved more problems than existing solvers
- Reason about the implementation utilizing ACL2s theorem prover.
- Plan to add support for a richer class of constraints in the future.

Questions

Try SeqSolve : **https://github.com/ankitku/SeqSolve**