Grid Structures and Undecidable Constraint Theories

Franck Seynhaeve
Marc Tommasi

LIFL
Lille, France

Ralf Treinen

LRI & CNRS
Orsay, France

Results of this Paper

The following are undecidable:

1. $\exists^*\forall^*$-fragment of the theory of set constraints
2. $\exists^*\forall^*$-fragment of the theory of one-step rewriting for shallow linear rewrite systems
3. emptiness for TRAFIC automata

All results obtained with a uniform method.

Set Constraints

Used in program analysis [Heintze & Jaffar]

Given: Σ: signature

Language: — function symbols of the signature

— binary predicate $\subseteq$

Structure S_Σ:

Universe: $2^{T(\Sigma)}$

$f^S(X_1, \ldots, X_n) \quad := \{f(x_1, \ldots, x_n) \mid x_i \in X_i\}$

$\subseteq^S$ subset

The Theory of Set Constraints

The set of all first-order sentences over set constraints that hold in S_Σ.

$$\exists x(x \subseteq f(x) \Rightarrow \forall y(y \subseteq x \vee \exists z(g(z) \subseteq h(x,y))))$$

Satisfiability of quantifier-free constraints: decidable [Aiken, Kozen & Wimmers 95], [Charatonik & Pacholski 94], [Gilleron, Tison & Tommasi 94]

Question: Does decidability extend to the first-order theory?

Quantifier complexity

$$\exists x(x \subseteq f(x) \Rightarrow \forall y(y \subseteq x \vee \exists z(g(z) \subseteq h(x,y))))$$

In Prenex Normal Form:

$$\exists x \forall y \exists z(x \subseteq f(x) \Rightarrow (y \subseteq x \vee g(z) \subseteq h(x,y))$$

belongs to the $\exists^*\forall^*\exists^*$-fragment.

Result on the theory of set constraints

Question (more precise): What is the simplest undecidable fragment of the theory of set constraints?

Result: The $\exists^*\forall^*$-fragment is undecidable.

(The $\exists^*$-fragment is known to be decidable.)

Method: Halting Problem

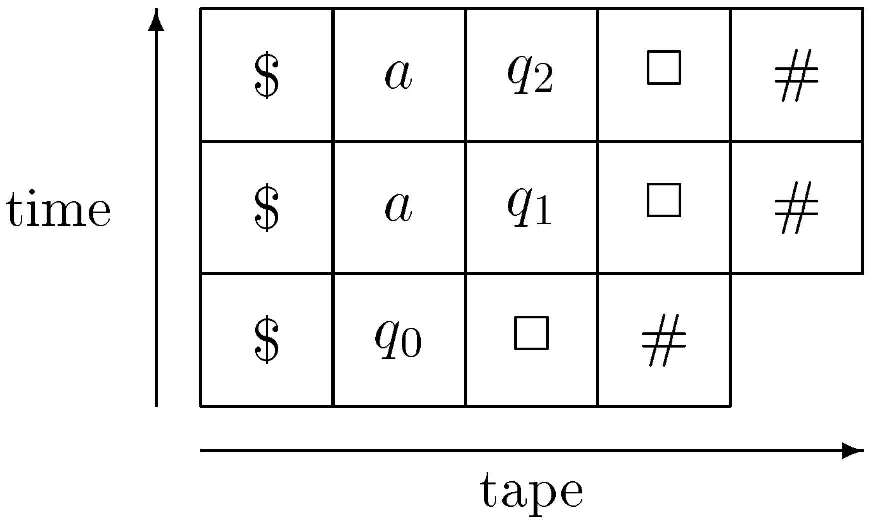
Reduce the halting problem of Turing Machines:

There is a sequence of TM configurations

- starting with an initial configuration
- ending in a final configuration
- consecutive configurations licensed by TM rules.

Method: Grids

Arrange sequences of TM configurations on a grid:



A grid is a TM run if

- certain 2-dimensional patterns match
- certain 2-dimensional patterns do not match

Method: Examples of forbidden pattern

TM production $(q, a) \to (p, b, \mapsto)$

$$\begin{array}{c|c} x & y \\ \hline q & a \end{array}$$

where $x \neq b$ or $y \neq p$

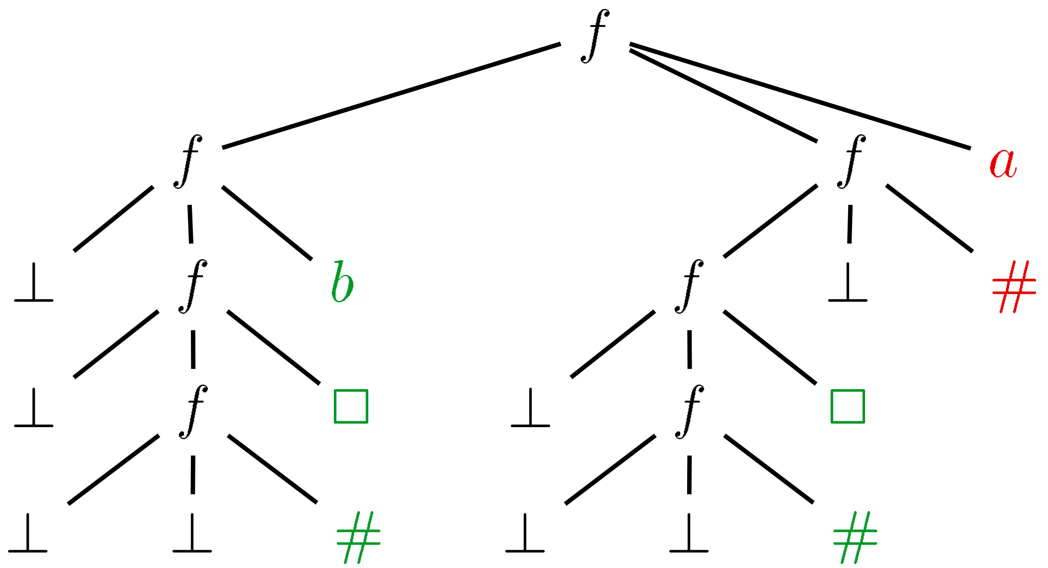
Tape symbols change only when under head:

$$\begin{array}{c|c} & z \\ \hline x & y \end{array}$$

where y tape symbol, x not a state, $z \neq y$

Method: Representing Grids

b	$\square$	$\#$
a	$\#$	



Method: Things to code

How to express

- that a term t represents a grid:

  If $f(f(\_, x, \_), f(y, \_, \_), \_)$ subterm of t, then $x = y$.

- that a certain linear pattern matches t.

Set constraints: Expressivity

$empty(X)$

$$f(X) \subseteq a$$

X and Y are disjoint ($X \parallel Y$)

$$\forall Z (Z \subseteq X \wedge Z \subseteq Y) \Rightarrow empty(Z)$$

X is a singleton ($singleton(X)$)

$$\neg empty(X) \wedge \forall Z\ Z \subseteq X \Rightarrow (empty(Z) \vee X \subseteq Z)$$

Set constraints: Grids 1

Idea: Define the set of all f-headed subterms of t:

$$
\begin{array}{ll}
 & singleton(X) \\
\wedge & X \subseteq S \\
\wedge & S \subseteq f(S, S, TapeSymbols) \cup \bot \\
\wedge & \forall U \; (U \subseteq f(U, U, TapeSymbols) \cup \bot \wedge X \subseteq U) \\
 & \qquad \Rightarrow S \subseteq U
\end{array}
$$

Set constraints: Grids 2

Enforce the equalities:

$$
\begin{array}{lcl}
\forall \bar{Y} & & \bigwedge_i \neg empty Y_i \\
& \wedge & f(f(Y_1, Y_2, Y_3), f(Y_4, Y_5, Y_6), Y_7) \subseteq S \\
& \Rightarrow & Y_2 = Y_4
\end{array}
$$

(and some more properties ...)

Set constraints: Patterns 2

X

b	$\square$	$\#$
a	$\#$	

S

b	$\square$	$\#$
a	$\#$	

b	$\square$	$\#$

$\square$	$\#$
$\#$	

$\square$	$\#$

$\#$

Set constraints: Patterns 1

Use S to check for patterns:

q_f occurs on the grid

$$\exists X, Y\ \neg empty(X) \wedge \neg empty(Y) \wedge f(X, Y, q_f) \subseteq S$$

One-Step Rewriting

Given: Σ: signature

R: Σ-rewrite system

Language: — binary predicate $\rightarrow$

— no equality

— no function symbols

Structure $A_{\Sigma,R}$: Universe: $T(\Sigma)$

$$A_{\Sigma,R} \models t \rightarrow s \quad \Leftrightarrow \quad t \rightarrow_R s$$

Result on the theory of one-step rewriting

There is no algorithm that decides the $\exists^*\forall^*$-fragment of $Th(A_{\Sigma,R})$ for any linear and shallow R.

non-linear:	$f(x, x) \to \ldots$	or	$\ldots \to f(x, x)$
non-shallow:	$f(a, g(x)) \to \ldots$	or	$\ldots \to f(a, g(x))$

One step rewriting: Motivation

- encompasses decidable properties of term rewriting systems (strong ground confluence, ground reducibility)
- decidability results for special cases:
  - unary signature
  - ground systems [Dauchet & Tison 90]
  - encompassment only [Caron, Coquide & Dauchet 93]
  - equality instead of rewriting [Comon, Haberstrau & Jouannaud, 94]
  - positive existential fragment [Niehren, Pinkal & Ruhrberg 97]

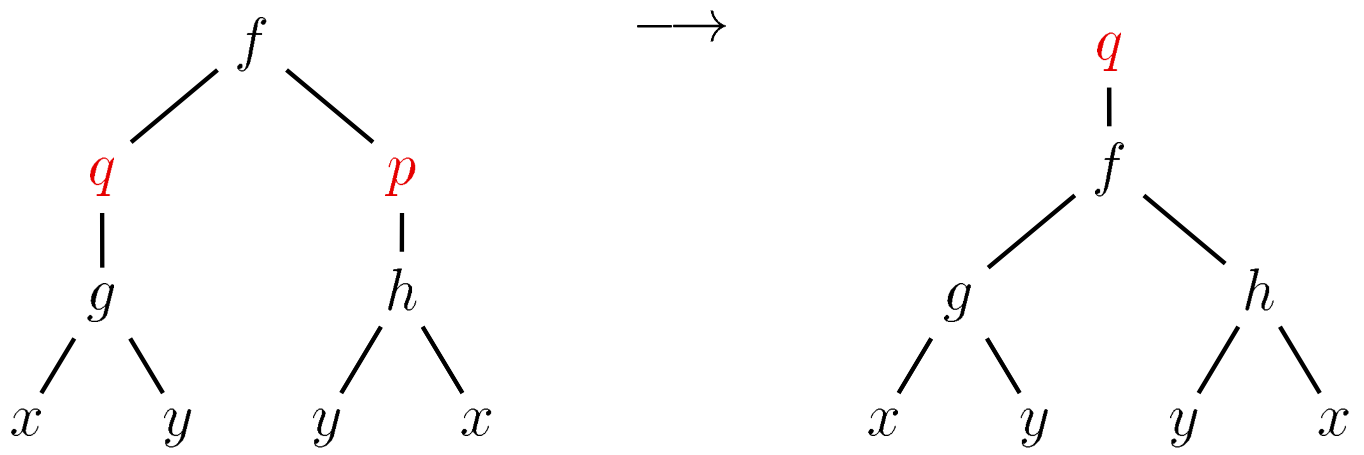
One step rewriting: Undecidability results

for particular classes of rewriting systems:

- linear [Treinen 96]
- right ground [Marcinkowski 97]
- linear and terminating [Vorobyov 97]

TRAFIC automata

Equality tests only between cousins:



Result: Emptyness undecidable.

Automata: related results

Decidability of emptyness:

- Equality tests only between siblings [Bogaert & Tison 92]
- Restriction on the number of equality tests in a run [Caron, Coquide & Dauchet 93]

Undecidability of emptyness:

- Arbitrary equality tests [Dauchet & Mongy 81]

Summary

All three problems: week in expressing equalities between subterms (coreferences)

Advantage of the grid method:

- one coreference (between cousins) must hold everywhere in the term
- matching of linear patterns