Extending Two-Variable Logic on Trees

(joint work with Witold Charatonik and Emanuel Kieroński)

Bartosz Bednarczyk

bbednarczyk@stud.cs.uni.wroc.pl
Institute of Computer Science
University of Wrocław
Wrocław, Poland

CSL 2017
Stockholm, August 24, 2017

Agenda

- Classical results on $\mathcal{FO}^2$ and related logics
- Logics on restricted classes of structures (words and trees)
- The main results of the paper
  - namely decidability and complexity of some tree logics
- Proof ideas
- Our current research

Historical results

Facts about SAT and $\mathcal{FO}^2$ on arbitrary structures

- We are interested in (finite) satisfiability problems
- Models = relational structures, no constants, no functions

Facts about SAT and $\mathcal{FO}^2$ on arbitrary structures

- We are interested in (finite) satisfiability problems
- Models = relational structures, no constants, no functions
- Some classical results:
  - $\mathcal{FO}$ undecidable (Church, Turing; 1930s)

Facts about SAT and $\mathcal{FO}^2$ on arbitrary structures

- We are interested in (finite) satisfiability problems
- Models = relational structures, no constants, no functions
- Some classical results:
  - $\mathcal{FO}$ undecidable (Church, Turing; 1930s)
  - $\mathcal{FO}^3$ undecidable (Kahr, Moore, Wang; 1959)

Facts about SAT and $\mathcal{FO}^2$ on arbitrary structures

- We are interested in (finite) satisfiability problems
- Models = relational structures, no constants, no functions
- Some classical results:
  - $\mathcal{FO}$ undecidable (Church, Turing; 1930s)
  - $\mathcal{FO}^3$ undecidable (Kahr, Moore, Wang; 1959)
  - $\mathcal{FO}^2$ decidable (Mortimer; 1975)
  - $\mathcal{FO}^2$ exponential model property (Gradel, Kolaitis, Vardi; 1997) - NEXPTIME-completess

Facts about SAT and $\mathcal{FO}^2$ on arbitrary structures

- We are interested in (finite) satisfiability problems
- Models = relational structures, no constants, no functions
- Some classical results:
  - $\mathcal{FO}$ undecidable (Church, Turing; 1930s)
  - $\mathcal{FO}^3$ undecidable (Kahr, Moore, Wang; 1959)
  - $\mathcal{FO}^2$ decidable (Mortimer; 1975)
  - $\mathcal{FO}^2$ exponential model property (Gradel, Kolaitis, Vardi; 1997) - NEXPTIME-completess
  - Connection between $\mathcal{FO}^2$ and modal, temporal, descriptive logics; many applications in verification and databases

Example formula:
from each element there exists a path of length 3

$$\forall x \exists y \, (E(x, y) \wedge \exists x \, (E(y, x) \wedge \exists y \; E(x, y)))$$

Logics on trees

Possible variations

There are several scenarios which may influence decidability/complexity. E.g., we may consider:

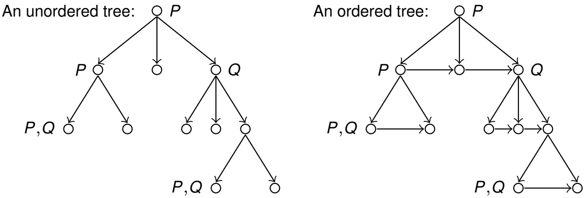
- Ordered vs Unordered trees
- Ranked vs Unranked trees
- Finite vs Infinite trees
- With unary alphabet restriction (UAR) or without UAR
  - precisely one unary predicate holds at each node
- ...

In this talk: Finite, Ordered, Unranked, No UAR Trees

Structures

Signature $\tau = \tau_0 \cup \tau_{nav} \cup \tau_{bin}$

- τ_0 – unary symbols (usually P, Q, etc.)
- τ_{nav} – *navigational* binary symbols with fixed interpretation
  - unordered trees: $\downarrow$ (child), $\downarrow_+$ (descendant, TC of $\downarrow$)
  - ordered trees: $\downarrow$, $\downarrow_+$, $\rightarrow$ (next sibling), $\rightarrow^+$ (TC of $\rightarrow$)
- τ_{bin} – additional *uninterpreted* binary symbols (may be empty)



Complexity results

- The complexity of $\mathcal{FO}$ and $\mathcal{MSO}$ on words and trees is non-elementary even for $\mathcal{FO}^3$ (Stockmeyer; 1974).

Complexity results

- The complexity of $\mathcal{FO}$ and $\mathcal{MSO}$ on words and trees is non-elementary even for $\mathcal{FO}^3$ (Stockmeyer; 1974).
- $\mathcal{FO}^2$ on finite words
  - $\leq$ is a linear word order and $+1$ is its induced successor relation
  - $\mathcal{FO}^2[+1, \leq]$ is NExpTime-complete (Etessami, Vardi, Wilke; 2002)
  - Equally expressive to Unary Temporal Logic
  - $\mathcal{FO}^2[+1, \leq, \tau_{bin}]$ is NExpTime-complete too (Thomas Zeume, Frederik Harwath 2016).

Complexity results

- The complexity of $\mathcal{FO}$ and $\mathcal{MSO}$ on words and trees is non-elementary even for $\mathcal{FO}^3$ (Stockmeyer; 1974).
- $\mathcal{FO}^2$ on finite words
  - $\leq$ is a linear word order and $+1$ is its induced successor relation
  - $\mathcal{FO}^2[+1, \leq]$ is NEXPTIME-complete (Etessami, Vardi, Wilke; 2002)
  - Equally expressive to Unary Temporal Logic
  - $\mathcal{FO}^2[+1, \leq, \tau_{bin}]$ is NEXPTIME-complete too (Thomas Zeume, Frederik Harwath 2016).
- $\mathcal{FO}^2$ on finite trees
  - $\mathcal{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ on trees is EXPSPACE-complete (Benaim, Benedikt, Charatonik, Kieronski, Lenhardt, Mazowiecki, Worrell; 2013).
  - Equally expressive to Navigational XPath.

Our results

Our settings

We work with two extensions of $\mathcal{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$.

- $\mathcal{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+, \tau_{bin}]$– extends $\mathcal{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ with additional uninterpreted binary symbols (τ_{bin})
- $\mathcal{C}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$– extends $\mathcal{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ with counting quantifiers of the form $\exists^{\leq n}, \exists^{\geq n}$ (n encoded in binary)
- We also combine these logic into $\mathcal{C}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+, \tau_{bin}]$.
- Recall that:
  - $\downarrow$ is a child relation
  - $\downarrow^+$ is a descendant relation
  - $\rightarrow$ is a right sibling relation
  - $\rightarrow^+$ is it's transitive closure

Our contribution

Theorem (FINSAT)

The finite satisfiability problem for $\mathcal{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+, \tau_{bin}]$ *and* $\mathcal{C}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ *is* EXPSPACE*-complete.*

Our contribution

Theorem (FINSAT)

The finite satisfiability problem for $\mathcal{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+, \tau_{bin}]$ *and* $\mathcal{C}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ *is* EXPSPACE*-complete.*

Theorem (Expressive power)

- $\mathcal{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ *and* $\mathcal{C}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ *are equally expressive.*
- $\mathcal{C}^2[\downarrow, \downarrow_+]$ *is more expressive than* $\mathcal{FO}^2[\downarrow, \downarrow_+]$.

Our contribution

Theorem (FINSAT)

The finite satisfiability problem for $\mathcal{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+, \tau_{bin}]$ *and* $\mathcal{C}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ *is* EXPSPACE*-complete.*

Theorem (Expressive power)

- $\mathcal{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ *and* $\mathcal{C}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ *are equally expressive.*
- $\mathcal{C}^2[\downarrow, \downarrow_+]$ *is more expressive than* $\mathcal{FO}^2[\downarrow, \downarrow_+]$*.*

Theorem (Combining two extensions)

The finite satisfiability problem for $\mathcal{C}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+, \tau_{bin}]$ *is at least as hard as checking non-emptiness for VATA/BVASS.*

Proof ideas

Expressive power

Theorem

$\mathcal{C}^2[\downarrow, \downarrow_+]$ is more expressive than $\mathcal{FO}^2[\downarrow, \downarrow_+]$.

Expressive power

Theorem

$\mathcal{C}^2[\downarrow, \downarrow_+]$ *is more expressive than* $\mathcal{FO}^2[\downarrow, \downarrow_+]$.

Proof.

Consider the formula $\exists x\ \exists^{\geq 3} y\ x \downarrow^+ y$. Easy to observe that Duplicator has a simple winning strategy in the standard two-pebble game of any length played on $\mathfrak{T}_3$ and $\mathfrak{T}_2$. □

Expressive power

Theorem

$\mathcal{C}^2[\downarrow, \downarrow_+]$ is more expressive than $\mathcal{FO}^2[\downarrow, \downarrow_+]$.

Proof.

Consider the formula $\exists x\ \exists^{\geq 3} y\ x \downarrow^{+} y$. Easy to observe that Duplicator has a simple winning strategy in the standard two-pebble game of any length played on $\mathfrak{T}_3$ and $\mathfrak{T}_2$. □

Theorem

$\mathcal{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ and $\mathcal{C}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ are equally expressive.

Proof.

Structural induction with elimination of counting quantifiers. □

Finite satisfiability cooking recipe

Finite satisfiability cooking recipe

- Step 1. Transform your formula into a normal form

Finite satisfiability cooking recipe

- Step 1. Transform your formula into a normal form
- Step 2. Desing a right notion of a type
  - And prove that your notion is ”correct” . . .

Finite satisfiability cooking recipe

- Step 1. Transform your formula into a normal form
- Step 2. Desing a right notion of a type
  - And prove that your notion is ”correct” . . .
- Step 3. Show small model property
  - Restrict your attention to trees with:
    - exponential degree of every nodes and
    - exponentially long paths
  - Do it by cutting out too long $\downarrow$ and $\rightarrow$-paths

Finite satisfiability cooking recipe

- Step 1. Transform your formula into a normal form
- Step 2. Desing a right notion of a type
  - And prove that your notion is ”correct” . . .
- Step 3. Show small model property
  - Restrict your attention to trees with:
    - exponential degree of every nodes and
    - exponentially long paths
  - Do it by cutting out too long $\downarrow$ and $\rightarrow$-paths
- Step 4. Present an alternating algorithm
  - in this case AEXPTIME

Finite satisfiability cooking recipe

- Step 1. Transform your formula into a normal form
- Step 2. Desing a right notion of a type
  - And prove that your notion is "correct" . . .
- Step 3. Show small model property
  - Restrict your attention to trees with:
    - exponential degree of every nodes and
    - exponentially long paths
  - Do it by cutting out too long $\downarrow$ and $\rightarrow$-paths
- Step 4. Present an alternating algorithm
  - in this case AEXPTIME

During this talk

- we will concentrate on $\mathcal{C}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$

Finite satisfiability cooking recipe

- Step 1. Transform your formula into a normal form
- Step 2. Desing a right notion of a type
  - And prove that your notion is ”correct” . . .
- Step 3. Show small model property
  - Restrict your attention to trees with:
    - exponential degree of every nodes and
    - exponentially long paths
  - Do it by cutting out too long $\downarrow$ and $\rightarrow$-paths
- Step 4. Present an alternating algorithm
  - in this case AEXPTIME

During this talk

- we will concentrate on $\mathcal{C}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$
- because that was my bachelor thesis

Finite satisfiability cooking recipe

- Step 1. Transform your formula into a normal form
- Step 2. Desing a right notion of a type
  - And prove that your notion is "correct" ...
- Step 3. Show small model property
  - Restrict your attention to trees with:
    - exponential degree of every nodes and
    - exponentially long paths
  - Do it by cutting out too long $\downarrow$ and $\rightarrow$-paths
- Step 4. Present an alternating algorithm
  - in this case AEXPTIME

During this talk

- we will concentrate on $\mathcal{C}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$
- because that was my bachelor thesis
- successfully defended in Feb 2017 :)

Order formulas

Order formulas specify the relative position of a pair of distinct elements in a tree. Assuming $\tau_{nav} = \{\downarrow, \downarrow_+, \rightarrow, \rightarrow^+\}$ there are ten of them:

- $\theta_=$: $x = y$,
- $\theta_\downarrow$: $x{\downarrow}y$,
- $\theta_\uparrow$: $y{\downarrow}x$,
- $\theta_{\downarrow\downarrow^+}$: $x{\downarrow_+}y \wedge \neg(x{\downarrow}y)$,
- $\theta_{\uparrow\uparrow^+}$: $y{\downarrow_+}x \wedge \neg(y{\downarrow}x)$,
- $\theta_\rightarrow$, $\theta_{\rightrightarrows^+}$, $\theta_{\leftleftarrows^+}$, $\theta_\leftarrow$ similar to the above for sibling relations
- $\theta_{\not\sim}$: $x{\not\sim}y$, (none of the above positions hold)

Scott normal form for $\mathcal{C}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$

- We translate a $\mathcal{C}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ formula into the following shape:

$$\varphi = \forall x \forall y\ \chi(x, y) \wedge \bigwedge_{i \in I} \forall x \exists^{\bowtie C_i} y\ \chi_i(x, y)$$

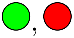
- $\bowtie_i \in \{\leq, \geq\}$ and the formulas χ, χ_i are quantifier-free
- Main property: quantifier depth is at most two
- Such form is polynomially computable and
- requires introducing some fresh unary symbols

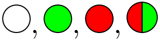
Atomic 1-types

- 1-type over a signature τ_0 is simply a subset of τ_0.
- We usually denote 1-types by α and their set by $\boldsymbol{\alpha}$
- A 1-type α can be identified with the conjunction

$$tp(x) = \bigwedge_{P \in \alpha} P(x) \wedge \bigwedge_{Q \notin \alpha} \neg Q(x)$$

- the number of 1-types is bounded exponentially in $|\tau|$
- Example:

Unary symbols $\tau_0 = \{$  $\} = \{$Green(), Red()$\}$

Possible 1-types $\boldsymbol{\alpha}_{\tau_0} = \{$  $\}$

- 1-type stores the information about a single node

A new ingredient - Full type - definition

- Recall that:
  - 1-types α store the color of a node

$$tp(x) = \bigwedge_{P \in \alpha} P(x) \wedge \bigwedge_{Q \notin \alpha} \neg Q(x)$$

  - Positions (assuming $\tau_{vav} = \{\downarrow, \downarrow^{+}, \rightarrow, \rightarrow^{+}\}$)

$$\Theta = \{\theta_{=}, \theta_{\downarrow}, \theta_{\uparrow}, \theta_{\downarrow\downarrow_{+}}, \theta_{\uparrow\uparrow^{+}}, \theta_{\rightarrow}, \theta_{\leftarrow}, \theta_{\rightrightarrows^{+}}, \theta_{\leftleftarrows^{+}}, \theta_{\not\sim}\}$$

A new ingredient - Full type - definition

- Recall that:
  - 1-types α store the color of a node

$$tp(x) = \bigwedge_{P \in \alpha} P(x) \wedge \bigwedge_{Q \notin \alpha} \neg Q(x)$$

  - Positions (assuming $\tau_{vav} = \{\downarrow, \downarrow^+, \rightarrow, \rightarrow^+\}$)

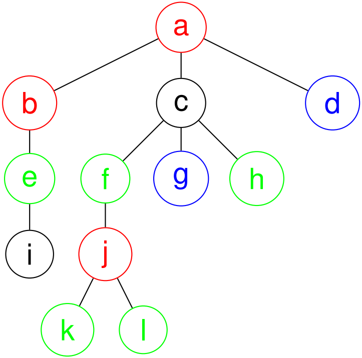
$$\Theta = \{\theta_=, \theta_\downarrow, \theta_\uparrow, \theta_{\downarrow\downarrow^+}, \theta_{\uparrow\uparrow^+}, \theta_\rightarrow, \theta_\leftarrow, \theta_{\rightrightarrows^+}, \theta_{\leftleftarrows^+}, \theta_{\not\sim}\}$$

- *C*-Full type stores the information about nodes, at the relative positions and their colors, from the vertex point of view. Formally:

$$C\text{-}ftp(x) :: \Theta \rightarrow \alpha \rightarrow \{0, 1, 2, \ldots, C, C+1, \infty\}$$

Full type example

$$\boldsymbol{\alpha} = \{ ●, ●, ●, ● \} \quad \mathit{ftp}(c) = ?$$

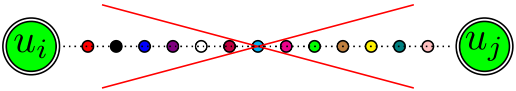


	●	●	●	●
$\theta_{=}$	0	0	0	1
$\theta_{\downarrow}$	1	0	2	0
$\theta_{\downarrow\downarrow^{+}}$	0	1	2	0
$\theta_{\not\sim}$	0	0	1	1
...				

Key lemma for the proof

Lemma (Pumping lemma)

Let φ be a normal form formula and let $C = \max_i C_i$ from the normal form. Let $\mathfrak{T} \models \varphi$. If there are two nodes on a root-to-leaf path of $\mathfrak{T}$ having the same C-full-type then we can remove all the vertices between them with subtrees rooted at them and obtain a shorter model $\mathfrak{T}'$.



$\mathcal{C}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ - further steps in the proof

- Unfortunately the number of different full types are doubly-exponential, so we obtain only doubly-exponential bound on the length of paths and degree of nodes.
- We introduced reduced full types:
  - We join below, above and free positions in the following way:

$$A = \theta_{\uparrow} \cup \theta_{\uparrow\uparrow^+}, B = \theta_{\downarrow} \cup \theta_{\downarrow\downarrow_+}, F = \bigcup \text{other}$$

$$C\text{-}rftp(x) :: \{A, B, F\} \rightarrow \boldsymbol{\alpha} \rightarrow \{0, 1, 2, \ldots, C, C+1, \infty\}$$

  - There are still doubly exponentially many reduced full types.
  - But they behave monotonically along root-to-leaf paths.
  - There are some problems with pumping lemma - details in the paper.

$\mathcal{C}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ - further steps in the proof

- Unfortunately the number of different full types are doubly-exponential, so we obtain only doubly-exponential bound on the length of paths and degree of nodes.
- We introduced reduced full types:
  - We join below, above and free positions in the following way:

$$A = \theta_{\uparrow} \cup \theta_{\uparrow\uparrow^+}, B = \theta_{\downarrow} \cup \theta_{\downarrow\downarrow_+}, F = \bigcup \text{other}$$

$$C\text{-}rftp(x) :: \{A, B, F\} \rightarrow \boldsymbol{\alpha} \rightarrow \{0, 1, 2, \ldots, C, C+1, \infty\}$$

  - There are still doubly exponentially many reduced full types.
  - But they behave monotonically along root-to-leaf paths.
  - There are some problems with pumping lemma - details in the paper.
- Conclusion: Exponentially long $\rightarrow$ and $\downarrow$-paths in trees.
- Next step: Algorithm - see the paper or ask for more details

Related logics over trees - current research

The following retain relatively low complexity

- $\mathcal{F}_1$ – one-dimensional fragment
  - fragment of $\mathcal{FO}$ in which blocks of existential (universal) quantifiers leave at most one variable free
  - $\exists y_1, \ldots, y_k \varphi(x, y_1, \ldots, y_k)$
  - 2-EXPTIME-complete, EXPSPACE-complete if the only navigational symbol is $\downarrow_+$
  - Just presented at MFCS 2017 in Aalborg :)

Related logics over trees - current research

The following retain relatively low complexity

- $\mathcal{F}_1$ – one-dimensional fragment
  - fragment of $\mathcal{FO}$ in which blocks of existential (universal) quantifiers leave at most one variable free
  - $\exists y_1, \ldots, y_k \varphi(x, y_1, \ldots, y_k)$
  - 2-EXPTIME-complete, EXPSPACE-complete if the only navigational symbol is $\downarrow_+$
  - Just presented at MFCS 2017 in Aalborg :)
- $\mathrm{FO}^2_{\mathrm{MOD}}$– $\mathcal{FO}^2$ + modulo counting quantifiers
  - allows quantifiers of the form $\exists^{=k (\mathrm{mod}\ l)}\ y\ \varphi(x, y)$
  - 2-EXPTIME-complete even when k, ls are binary coded
  - Submitted.
- The same decidability schema as for $\mathcal{C}^2$!

Questions?

Thank you for your attention