Positive first-order logic on words

Denis Kuperberg

CNRS, LIP, ENS Lyon

IRIF séminaire automates
30 avril 2021

Monotone languages

Ordered alphabet: Finite alphabet A with partial order $\leq_A$.

Monotone languages

Ordered alphabet: Finite alphabet A with partial order $\leq_A$.

Example of powerset alphabet:
$A = 2^P$, Order $\leq_A$ is inclusion

Monotone languages

Ordered alphabet: Finite alphabet A with partial order $\leq_A$.

Example of powerset alphabet:
$A = 2^P$, Order $\leq_A$ is inclusion

Definition (monotone languages)

$L \subseteq A^*$ is monotone if $\forall$ words u, v and letters a, b,

$$uav \in L \text{ and } a \leq_A b \Longrightarrow ubv \in L.$$

Monotone languages

Ordered alphabet: Finite alphabet A with partial order $\leq_A$.

Example of powerset alphabet:
$A = 2^P$, Order $\leq_A$ is inclusion

Definition (monotone languages)

$L \subseteq A^*$ is monotone if $\forall$ words u, v and letters a, b,

$$uav \in L \text{ and } a \leq_A b \Longrightarrow ubv \in L.$$

Example

On $A = \{a, b\}$ with $a \leq_A b$:

- $A^* b A^*$ is monotone.
- Its complement a^* is not monotone.

Positive first-order logic

How to syntactically define monotone languages ?

Positive first-order logic

How to syntactically define monotone languages ?

Definition (FO$^+$)

$\varphi, \psi := a^{\uparrow}(x) \mid x \leq y \mid x < y \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \exists x.\varphi \mid \forall x.\varphi$

- No negation: all predicates appear positively.
- Atomic predicate $a^{\uparrow}(x)$ with $a \in A$: $a \leq_A \mathrm{label}(x)$.

Positive first-order logic

How to syntactically define monotone languages ?

Definition (FO$^+$)

$\varphi, \psi := a^{\uparrow}(x) \mid x \leq y \mid x < y \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \exists x.\varphi \mid \forall x.\varphi$

- No negation: all predicates appear positively.
- Atomic predicate $a^{\uparrow}(x)$ with $a \in A$: $a \leq_A \mathrm{label}(x)$.

Example

On alphabet $A = \{a, b\}$ with $a \leq_A b$.

- $\exists x.b^{\uparrow}(x)$ recognizes $A^* b A^*$.
- $\forall x.a^{\uparrow}(x)$ recognizes the full A^*. (not only a^*)

Positive first-order logic

How to syntactically define monotone languages ?

Definition (FO$^+$)

$\varphi, \psi := a^\uparrow(x) \mid x \leq y \mid x < y \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \exists x.\varphi \mid \forall x.\varphi$

- No negation: all predicates appear positively.
- Atomic predicate $a^\uparrow(x)$ with $a \in A$: $a \leq_A \mathrm{label}(x)$.

Example

On alphabet $A = \{a, b\}$ with $a \leq_A b$.

- $\exists x.b^\uparrow(x)$ recognizes A^*bA^*.
- $\forall x.a^\uparrow(x)$ recognizes the full A^*. (not only a^*)

Fact: L definable in FO$^+$ $\Rightarrow$ L monotone and FO-definable.

Positive first-order logic

How to syntactically define monotone languages ?

Definition (FO^+)

$\varphi, \psi := a^\uparrow(x) \mid x \leq y \mid x < y \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \exists x.\varphi \mid \forall x.\varphi$

- No negation: all predicates appear positively.
- Atomic predicate $a^\uparrow(x)$ with $a \in A$: $a \leq_A \mathrm{label}(x)$.

Example

On alphabet $A = \{a, b\}$ with $a \leq_A b$.

- $\exists x.b^\uparrow(x)$ recognizes $A^* b A^*$.
- $\forall x.a^\uparrow(x)$ recognizes the full A^*. (not only a^*)

Fact: L definable in $\mathrm{FO}^+ \Rightarrow L$ monotone and FO-definable.

T. Colcombet: Is the converse true ?

A counter-example language

Our first result

There is L monotone, FO-definable but not FO^+-definable.

A counter-example language

Our first result

There is L monotone, FO-definable but not FO^+-definable.

Alphabet $A = \{a, b, c, \binom{a}{b}, \binom{b}{c}, \binom{c}{a}\}$, with $i, j \leq_A \binom{i}{j}$.

A counter-example language

Our first result

There is L monotone, FO-definable but not FO^+-definable.

Alphabet $A = \{a, b, c, \binom{a}{b}, \binom{b}{c}, \binom{c}{a}\}$, with $i, j \leq_A \binom{i}{j}$.

Language $L = (a^\uparrow b^\uparrow c^\uparrow)^*$.

A counter-example language

Our first result

There is L monotone, FO-definable but not FO$^+$-definable.

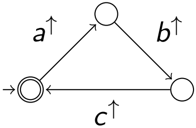
Alphabet $A = \{a, b, c, \binom{a}{b}, \binom{b}{c}, \binom{c}{a}\}$, with $i, j \leq_A \binom{i}{j}$.

Language $L = (a^\uparrow b^\uparrow c^\uparrow)^*$.

Lemma

L is FO-definable.

Proof: Verify that



is counter-free.

I.e. no word induces a non-trivial cycle.

A counter-example language

Our first result

There is L monotone, FO-definable but not FO$^+$-definable.

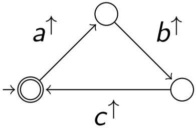
Alphabet $A = \{a, b, c, \binom{a}{b}, \binom{b}{c}, \binom{c}{a}\}$, with $i, j \leq_A \binom{i}{j}$.

Language $L = (a^{\uparrow} b^{\uparrow} c^{\uparrow})^*$.

Lemma

L is FO-definable.

Proof: Verify that



is counter-free.

I.e. no word induces a non-trivial cycle.

To prove L is not FO^+-definable: Ehrenfeucht-Fraïssé games.

Ehrenfeucht-Fraïssé games for FO

Definition (EF games)

Played on two words u, v. At each round i:

- Spoiler places token i in u or v.
- Duplicator must answer token i in the other word such that
  - the letter on token i is the same in u and v.
  - the tokens are in the same order in u and v.

Ehrenfeucht-Fraïssé games for FO

Definition (EF games)

Played on two words u, v. At each round i:

- Spoiler places token i in u or v.
- Duplicator must answer token i in the other word such that
  - the letter on token i is the same in u and v.
  - the tokens are in the same order in u and v.

Let us note $u \equiv_n v$ if Duplicator can survive n rounds on u, v.

Ehrenfeucht-Fraïssé games for FO

Definition (EF games)

Played on two words u, v. At each round i:

- Spoiler places token i in u or v.
- Duplicator must answer token i in the other word such that
  - the letter on token i is the same in u and v.
  - the tokens are in the same order in u and v.

Let us note $u \equiv_n v$ if Duplicator can survive n rounds on u, v.

Theorem (Ehrenfeucht,Fraïssé, 1950-1961)

L not FO-definable $\Leftrightarrow$ For all n, there are $u \in L$, $v \notin L$ s.t. $u \equiv_n v$.

Ehrenfeucht-Fraïssé games for FO

Definition (EF games)

Played on two words u, v. At each round i:

- Spoiler places token i in u or v.
- Duplicator must answer token i in the other word such that
  - the letter on token i is the same in u and v.
  - the tokens are in the same order in u and v.

Let us note $u \equiv_n v$ if Duplicator can survive n rounds on u, v.

Theorem (Ehrenfeucht,Fraïssé, 1950-1961)

L not FO-definable $\Leftrightarrow$ For all n, there are $u \in L$, $v \notin L$ s.t. $u \equiv_n v$.

Example

Proving $(aa)^*$ is not FO-definable:

$$\begin{array}{lll} u = a^{2k} & \in (aa)^* : & a\ a\ a\ a\ a\ a\ a\ a \\ v = a^{2k+1} & \notin (aa)^* : & a\ a\ a\ a\ a\ a\ a\ a\ a \end{array}$$

Proving FO^+-undefinability

Definition (EF^+ games)

New rule: we only ask letters in u to be $\leq_A$-smaller than corresponding ones in v.

We write $u \preceq_n v$ if Duplicator can survive n rounds.

Proving FO^+-undefinability

Definition (EF^+ games)

New rule: we only ask letters in u to be $\leq_A$-smaller than corresponding ones in v.

We write $u \preceq_n v$ if Duplicator can survive n rounds.

Theorem (Correctness of EF^+ games)

L not FO^+-definable $\Leftrightarrow \forall n$, there are $u \in L$, $v \notin L$ s.t. $u \preceq_n v$.
[Stolboushkin 1995+this work]

Proving FO^+-undefinability

Definition (EF^+ games)

New rule: we only ask letters in u to be $\leq_A$-smaller than corresponding ones in v.

We write $u \preceq_n v$ if Duplicator can survive n rounds.

Theorem (Correctness of EF^+ games)

L not FO^+-definable $\Leftrightarrow \forall n$, there are $u \in L$, $v \notin L$ s.t. $u \preceq_n v$.
[Stolboushkin 1995+this work]

Application: Proving L is not FO^+-definable

$$\begin{array}{lccccccccc} u \in L: & a & b & c & a & b & c & a & b & c \\ v \notin L: & \binom{a}{b} & \binom{b}{c} & \binom{c}{a} & \binom{a}{b} & \binom{b}{c} & \binom{c}{a} & \binom{a}{b} & \binom{b}{c} & \end{array}$$

Background: Lyndon's theorem

Zoom out: FO on arbitrary structures.

Theorem (Lyndon 1959)

FO-definable and monotone $\Leftrightarrow$ *FO*$^{+}$*-definable.*

φ *preserved by surjective morphisms* $\Leftrightarrow$ *equivalent to a positive formula.*

Background: Lyndon's theorem

Zoom out: FO on arbitrary structures.

Theorem (Lyndon 1959)

FO-definable and monotone $\Leftrightarrow$ FO$^+$-definable.

φ preserved by surjective morphisms $\Leftrightarrow$ equivalent to a positive formula.

Theorem

Lyndon's theorem fails on finite structures:

- *[Ajtai, Gurevich 1987]*
  lattices, probabilities, number theory, topology, very hard
- *[Stolboushkin 1995]*
  EF^+ *games on grids, involved*

Background: Lyndon's theorem

Zoom out: FO on arbitrary structures.

Theorem (Lyndon 1959)

FO-definable and monotone $\Leftrightarrow$ FO$^+$-definable.

φ preserved by surjective morphisms $\Leftrightarrow$ equivalent to a positive formula.

Theorem

Lyndon's theorem fails on finite structures:

- *[Ajtai, Gurevich 1987]*
  lattices, probabilities, number theory, topology, very hard
- *[Stolboushkin 1995]*
  EF^+ *games on grids, involved*
- *[This work]*
  EF^+ *games on words, easy*

Can we decide FO^+-definability?

Theorem

Given a regular L on an ordered alphabet, we can decide

- *whether L is monotone (e.g. automata inclusion)*
- *whether L is FO-definable [Schützenberger, McNaughton, Papert]*

Can we decide whether L is FO^+-definable ?

Can we decide FO^+-definability?

Theorem

Given a regular L on an ordered alphabet, we can decide

- *whether L is monotone* (e.g. automata inclusion)
- *whether L is FO-definable* [Schützenberger, McNaughton, Papert]

Can we decide whether L is FO^+-definable ?

Our second result

FO^+-definability is undecidable for regular languages.

Can we decide FO^+-definability?

Theorem

Given a regular L on an ordered alphabet, we can decide

- *whether L is monotone (e.g. automata inclusion)*
- *whether L is FO-definable [Schützenberger, McNaughton, Papert]*

Can we decide whether L is FO^+-definable ?

Our second result

FO^+-definability is undecidable for regular languages.

Reduction from *Turing Machine Mortality*:

A deterministic TM M is *mortal* if there a uniform bound n on the runs of M from *any* configuration.

Undecidable [Hooper 1966].

Undecidability proof sketch

Given a TM M, we build a regular language L such that

$$M \text{ mortal } \Leftrightarrow L \text{ is } \mathrm{FO}^{+}\text{-definable.}$$

Undecidability proof sketch

Given a TM M, we build a regular language L such that

$$M \text{ mortal } \Leftrightarrow L \text{ is } \mathrm{FO}^+\text{-definable.}$$

Building L:
Inspired from $(a^\uparrow b^\uparrow c^\uparrow)^*$, but:

- $a, b, c \rightsquigarrow$ Words from C_1, C_2, C_3 encoding configs of M.
- All transitions of M follow the cycle:
$$\begin{tikzcd} & C_2 \arrow[dr] & \\ C_1 \arrow[ur] & & C_3 \arrow[ll] \end{tikzcd}$$
- $\binom{a}{b}, \binom{b}{c}, \binom{c}{a} \rightsquigarrow \binom{u_1}{u_2}$, exists iff $u_1 \xrightarrow{M} u_2$.

Undecidability proof sketch

Given a TM M, we build a regular language L such that

$$M \text{ mortal } \Leftrightarrow L \text{ is } \mathrm{FO}^+\text{-definable.}$$

Building L:
Inspired from $(a^\uparrow b^\uparrow c^\uparrow)^*$, but:

- $a, b, c \rightsquigarrow$ Words from C_1, C_2, C_3 encoding configs of M.
- All transitions of M follow the cycle: $\begin{tikzcd} & C_2 \arrow[dr] & \\ C_1 \arrow[ur] & & C_3 \arrow[ll] \end{tikzcd}$
- $\binom{a}{b}, \binom{b}{c}, \binom{c}{a} \rightsquigarrow \binom{u_1}{u_2}$, exists iff $u_1 \xrightarrow{M} u_2$.

We choose

$$L := (C_1^\uparrow \cdot C_2^\uparrow \cdot C_3^\uparrow)^*$$

Undecidability proof sketch

Given a TM M, we build a regular language L such that

$$M \text{ mortal } \Leftrightarrow L \text{ is } \mathrm{FO}^{+}\text{-definable.}$$

Building L:
Inspired from $(a^{\uparrow} b^{\uparrow} c^{\uparrow})^{*}$, but:

- $a, b, c \rightsquigarrow$ Words from C_1, C_2, C_3 encoding configs of M.
- All transitions of M follow the cycle:

$$\begin{tikzcd} & C_2 \arrow[dr] & \\ C_1 \arrow[ur] & & C_3 \arrow[ll] \end{tikzcd}$$

- $\binom{a}{b}, \binom{b}{c}, \binom{c}{a} \rightsquigarrow \binom{u_1}{u_2}$, exists iff $u_1 \xrightarrow{M} u_2$.

We choose

$$L := (C_1^{\uparrow} \cdot C_2^{\uparrow} \cdot C_3^{\uparrow})^{*}$$

$u \in L \not\Rightarrow u$ encodes a run of M.

The reduction

If M not mortal:
Let $u_1, u_2, \ldots, u_n$ a long run of M, and play Duplicator in :

$$\begin{array}{lcccccc} u \in L: & u_1 & u_2 & u_3 & \ldots & u_{n-1} & u_n \\ v \notin L: & \binom{u_1}{u_2} & \binom{u_2}{u_3} & \binom{u_3}{u_4} & \ldots & \binom{u_{n-1}}{u_n} & \end{array}$$

$\to$ L is not FO^+-definable.

The reduction

If M not mortal:
Let $u_1, u_2, \ldots, u_n$ a long run of M, and play Duplicator in :

$$\begin{array}{llllllll} u \in L: & u_1 & u_2 & u_3 & \ldots & u_{n-1} & u_n \\ v \notin L: & \binom{u_1}{u_2} & \binom{u_2}{u_3} & \binom{u_3}{u_4} & \ldots & \binom{u_{n-1}}{u_n} \end{array}$$

$\rightarrow$ L is not FO^+-definable.

If M mortal with bound n:
Abstract u_i by the length of the run of M starting in it (at most n).

The reduction

If M not mortal:
Let $u_1, u_2, \dots, u_n$ a long run of M, and play Duplicator in :

$$\begin{array}{lcccccc} u \in L: & u_1 & u_2 & u_3 & \dots & u_{n-1} & u_n \\ v \notin L: & \binom{u_1}{u_2} & \binom{u_2}{u_3} & \binom{u_3}{u_4} & \dots & \binom{u_{n-1}}{u_n} & \end{array}$$

$\rightarrow$ L is not FO^+-definable.

If M mortal with bound n:
Abstract u_i by the length of the run of M starting in it (at most n).

Play Spoiler in the abstracted game (here $n = 5$):

$$\begin{array}{lcccccccccc} u: & 2 & 3 & 2 & 4 & 3 & 5 & 4 & 3 & 4 & 4 \\ v: & \binom{2}{1} & \binom{3}{2} & \binom{2}{1} & \binom{4}{3} & \binom{3}{2} & \binom{5}{4} & \binom{4}{3} & \binom{5}{4} & \binom{5}{4} & \end{array}$$

The reduction

If M not mortal:

Let $u_1, u_2, \ldots, u_n$ a long run of M, and play Duplicator in :

$$\begin{array}{lcccccc} u \in L : & u_1 & u_2 & u_3 & \ldots & u_{n-1} & u_n \\ v \notin L : & \binom{u_1}{u_2} & \binom{u_2}{u_3} & \binom{u_3}{u_4} & \ldots & \binom{u_{n-1}}{u_n} & \end{array}$$

$\rightarrow$ L is not FO^+-definable.

If M mortal with bound n:

Abstract u_i by the length of the run of M starting in it (at most n).

Play Spoiler in the abstracted game (here $n = 5$):

$$\begin{array}{lcccccccccc} u : & 2 & 3 & 2 & 4 & 3 & 5 & 4 & 3 & 4 & 4 \\ v : & \binom{2}{1} & \binom{3}{2} & \binom{2}{1} & \binom{4}{3} & \binom{3}{2} & \binom{5}{4} & \binom{4}{3} & \binom{5}{4} & \binom{5}{4} & \end{array}$$

Spoiler always wins in $2n$ rounds $\rightarrow$ L is FO^+-definable.

Ongoing work

With Thomas Colcombet:
Exploring the consequences of this in other frameworks:

- regular cost functions,
- logics on linear orders,
- ...

Slogan:
Variants of FO not closed under complement will often display this behaviour.

Ongoing work

With Thomas Colcombet:
Exploring the consequences of this in other frameworks:

- regular cost functions,
- logics on linear orders,
- ...

Slogan:
Variants of FO not closed under complement will often display this behaviour.

Thanks for your attention !

Paper at arxiv.org/pdf/2101.01968, to be presented at LICS 2021.