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 $\leq_A$. 

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 $a, b$, $uav \in L$ and $a \leq_A b \Rightarrow ubv \in L$.

Example
On $A = \{a, b\}$ with $a \leq_A b$:
$\rightarrow A^*bA^*$ is monotone.
$\rightarrow$ Its complement $a^*$ is not monotone.
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 \( a, b \),

\[ uav \in L \text{ and } a \leq_A b \implies ubv \in L. \]
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 $a, b$,

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

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$^+$)
\[\varphi, \psi := a^\uparrow(x) \mid x \leq y \mid x < y \mid \varphi \lor \psi \mid \varphi \land \psi \mid \exists x. \varphi \mid \forall x. \varphi\]

- No negation: all predicates appear positively.
- Atomic predicate \(a^\uparrow(x)\) with \(a \in A\): \(a \leq_A \text{label}(x)\).
Positive first-order logic

How to syntactically define monotone languages?

**Definition (FO⁺)**

\[ \varphi, \psi := \mathbf{a}^\uparrow(x) | x \leq y | x < y | \varphi \lor \psi | \varphi \land \psi | \exists x. \varphi | \forall x. \varphi \]

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

**Example**

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

- \( \exists x. \mathbf{b}^\uparrow(x) \) recognizes \( A^* b A^* \).
- \( \forall x. \mathbf{a}^\uparrow(x) \) recognizes the full \( A^* \). *(not only \( a^* \)*)
Positive first-order logic

How to syntactically define monotone languages?

**Definition (FO⁺)**

\[ \varphi, \psi := a^\uparrow(x) \mid x \leq y \mid x < y \mid \varphi \lor \psi \mid \varphi \land \psi \mid \exists x. \varphi \mid \forall x. \varphi \]

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

**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 \( \text{FO}^+ \) \( \Rightarrow \) \( L \) monotone and \( \text{FO}-\text{definable} \).
Positive first-order logic

How to syntactically define monotone languages?

**Definition (FO+)**
\[ \varphi, \psi := a^\uparrow(x) \mid x \leq y \mid x < y \mid \varphi \lor \psi \mid \varphi \land \psi \mid \exists x. \varphi \mid \forall x. \varphi \]

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

**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 \( \text{FO}^+ \) \( \Rightarrow \) \( L \) monotone and \( \text{FO} \)-definable.

**T. Colcombet:** Is the converse true?
### A counter-example language

#### Our first result

There is $L$ monotone, FO-definable but not $\text{FO}^+$-definable.
A counter-example language

Our first result

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

Alphabet $A = \{a, b, c, (a)_b, (b)_c, (c)_a\}$, with $i, j \leq_A (i)$. 
A counter-example language

Our first result

There is $L$ monotone, FO-definable but not $\text{FO}^+$-definable.

Alphabet $A = \{a, b, c, (a)_b, (b)_c, (c)_a\}$, with $i, j \leq_A (i)$. Language $L = (a^\uparrow b^\uparrow c^\uparrow)^*$. 

Lemma $L$ is FO-definable.

Proof: Verify that $a^\uparrow b^\uparrow c^\uparrow$ is counter-free. I.e. no word induces a non-trivial cycle.

To prove $L$ is not $\text{FO}^+$-definable: Ehrenfeucht-Fraïssé games.
A counter-example language

Our first result

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

Alphabet \( A = \{a, b, c, (a)_b, (b)_c, (c)_a\} \), with \( i, j \leq_A (i) \).

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

**Lemma**

\( L \) is FO-definable.

**Proof**: Verify that \( a\uparrow b\uparrow c\uparrow \) 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 \( A = \{a, b, c, (\frac{a}{b}), (\frac{b}{c}), (\frac{c}{a})\} \), with \( i, j \leq_A \binom{i}{j} \).

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

**Lemma**

\( L \) is FO-definable.

**Proof**: Verify that \( a^\uparrow b^\uparrow c^\uparrow \) 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 \( v \).

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 \iff \forall n, \exists u \in L, v \notin L \ s.t. u \equiv_n v. \)

Example

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

\( u = a^2k \in (aa)^*: \quad a \ a \ a \ a \ a \ a \ a \ a \ a \)

\( v = a^{2k+1} \notin (aa)^*: \quad a \ a \ a \ a \ a \ a \ a \ a \ a \)
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 \iff For all $n$, there are $u \in L, v \not\in L$ s.t. $u \equiv_n v$.

Example

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

- $u = a2^{k} \in (aa)^*$:
  - $a \ a \ a \ a \ a \ a \ a $.
- $v = a2^{k}+1 \not\in (aa)^*$:
  - $a \ a \ a \ a \ a \ a \ a $.
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 $\iff$ For all $n$, there are $u \in L$, $v \not\in L$ 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 $\iff$ For all $n$, there are $u \in L$, $v \notin L$ s.t. $u \equiv_n v$.

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

\[
\begin{align*}
  u &= a^{2k} & \in (aa)^*: & a \ a \ a \ a \ a \ a \ a \ a \\
  v &= a^{2k+1} & \notin (aa)^*: & a \ a \ a \ a \ a \ a \ a \ a \ a
\end{align*}
\]
Proving $\text{FO}^+-\text{undefinability}$

**Definition (EF$^+$ games)**

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

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

Application: Proving $L$ is not $\text{FO}^+-\text{definable}$

$u \in L$: $a\ b\ c\ a\ b\ c\ a\ b\ c$

$v \notin L$: $(a\ b)\ (b\ c)\ (c\ a)\ (a\ b)\ (b\ c)\ (c\ a)\ (a\ b)$
Proving $\text{FO}^+$-undefinability

**Definition (EF$^+$ games)**

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

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

**Theorem (Correctness of EF$^+$ games)**

$L$ not $\text{FO}^+$-definable $\iff \forall n$, there are $u \in L$, $v \notin L$ s.t. $u \leq_n v$.

[Stolboushkin 1995 + this work]
Proving $\text{FO}^+\text{-undefinability}$

**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 $\text{FO}^+$-definable $\iff \forall n$, there are $u \in L$, $v \notin L$ s.t. $u \preceq_n v$.

[Stolboushkin 1995+this work]

**Application: Proving $L$ is not $\text{FO}^+$-definable**

$u \in L : \ a \ b \ c \ a \ b \ c \ a \ b \ c$

$v \notin L : \ (a) \ (b) \ (c) \ (a) \ (b) \ (c) \ (a) \ (b) \ (c)$
Background: Lyndon’s theorem

Zoom out: FO on arbitrary structures.

Theorem (Lyndon 1959)

*FO*-definable and monotone $\iff$ *FO*+-definable.

$\varphi$ preserved by surjective morphisms $\iff$ equivalent to a positive formula.
Background: Lyndon’s theorem

Zoom out: FO on arbitrary structures.

Theorem (Lyndon 1959)

*FO-definable and monotone* ⇔ *FO⁰⁺-definable.*

ϕ preserved by surjective morphisms ⇔ 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 ⇔ FO⁺-definable.*

ϕ preserved by surjective morphisms ⇔ 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 $\text{FO}^+$-definability?

**Theorem**

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

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

Can we decide whether $L$ is $\text{FO}^+$-definable?
Can we decide $\text{FO}^+$-definability?

**Theorem**

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

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

Can we decide whether $L$ is $\text{FO}^+$-definable?

**Our second result**

$\text{FO}^+$-definability is undecidable for regular languages.
Can we decide \( \mathbf{FO}^+ \)-definability?

**Theorem**

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

- whether \( L \) is monotone (e.g. automata inclusion)
- whether \( L \) is \( \mathbf{FO} \)-definable \([\text{Schützenberger, McNaughton, Papert}]\)

Can we decide whether \( L \) is \( \mathbf{FO}^+ \)-definable?

**Our second result**

\( \mathbf{FO}^+ \)-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 \([\text{Hooper 1966}]\).
Undecidability proof sketch

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

$$M \text{ mortal } \iff L \text{ is } \FO^+\text{-definable.}$$
Undecidability proof sketch

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

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

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

- $a, b, c \rightsquigarrow$ Words from $C_1, C_2, C_3$ encoding configs of $M$.

- All transitions of $M$ follow the cycle:

  $$C_1 \leftarrow C_2 \rightarrow C_3$$

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

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


Undecidability proof sketch

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

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

**Building $L$:**

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

- $a, b, c \sim \text{ Words from } C_1, C_2, C_3 \text{ encoding configs of } M$.

- All transitions of $M$ follow the cycle:

  $C_1 \sim C_2 \sim C_3$.

- $(a^\uparrow, b^\uparrow, c^\uparrow) \sim (u_1^\uparrow, u_2^\uparrow)$, 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

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

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_1 & C_2 & C_3 \\
  & \leftarrow & \\
  \\
\end{tikzcd}

- $(a^\downarrow, b^\downarrow, c^\downarrow) \rightsquigarrow (u_1^\downarrow, u_2^\downarrow)$, exists iff $u_1^M \rightarrow u_2$.

We choose

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

$u \in L \nRightarrow u$ encodes a run of $M$. 

\[\text{\textbf{Warning}}\quad u \in L \nRightarrow u \text{ encodes a run of } M.\]
The reduction

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

$$u \in L : \ u_1 \ u_2 \ u_3 \ \ldots \ \ u_{n-1} \ u_n$$
$$v \notin L : \ (u_1 \ u_2) \ (u_2 \ u_3) \ (u_3 \ u_4) \ \ldots \ (u_{n-1} \ u_n)$$

$\rightarrow L$ is not $\text{FO}^+$-definable.
The reduction

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

\[
\begin{align*}
&u \in L: \quad u_1 \ u_2 \ u_3 \ \ldots \ \ u_{n-1} \ u_n \\
&v \not\in L: \quad (u_1 \ u_2) \ (u_2 \ u_3) \ (u_3 \ u_4) \ \ldots \ (u_{n-1} \ u_n) \\
\rightarrow & \ L \text{ is not FO}^+\text{-definable.}
\end{align*}
\]

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

$$u \in L : \quad u_1 \quad u_2 \quad u_3 \quad \ldots \quad u_{n-1} \quad u_n$$
$$v \notin L : \quad \left(\frac{u_1}{u_2}\right) \left(\frac{u_2}{u_3}\right) \left(\frac{u_3}{u_4}\right) \ldots \left(\frac{u_{n-1}}{u_n}\right)$$

$\rightarrow L$ is not $\mathsf{FO}^+$-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 $n = 5$): 

$$u : \quad 2 \quad 3 \quad 2 \quad 4 \quad 3 \quad 5 \quad 4 \quad 3 \quad 4 \quad 4$$
$$v : \quad \left(\frac{2}{1}\right) \left(\frac{3}{2}\right) \left(\frac{2}{1}\right) \left(\frac{4}{3}\right) \left(\frac{3}{2}\right) \left(\frac{5}{4}\right) \left(\frac{4}{3}\right) \left(\frac{5}{4}\right) \left(\frac{5}{4}\right)$$

Spoiler always wins in $2n$ rounds $\rightarrow L$ is $\mathsf{FO}^+$-definable.
The reduction

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

$$u \in L : \quad u_1 \quad u_2 \quad u_3 \quad \ldots \quad u_{n-1} \quad u_n$$
$$v \notin L : \quad (u_1 \ u_2) \quad (u_2 \ u_3) \quad (u_3 \ u_4) \quad \ldots \quad (u_{n-1} \ u_n)$$

$\rightarrow L$ is not $\text{FO}^+$-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 $n = 5$):

$$u : \quad 2 \quad 3 \quad 2 \quad 4 \quad 3 \quad 5 \quad 4 \quad 3 \quad 4 \quad 4$$
$$v : \quad (2 \quad \ 3) \quad (2 \quad \ 1) \quad (4 \quad \ 3) \quad (3 \quad \ 2) \quad (5 \quad \ 4) \quad (4 \quad \ 3) \quad (5 \quad \ 4)$$

Spoiler always wins in $2n$ rounds $\rightarrow L$ is $\text{FO}^+$-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!