Positive first-order logic on words

Denis Kuperberg

CNRS, LIP, ENS Lyon

Séminaire Méthodes Formelles, LaBRI, 28/09/21
This work was presented at LICS 2021
$\text{FO}^+$ and the powerset alphabet

A special language

Background: Lyndon’s theorem

Undecidability result
The $\text{FO}^+$ logic, words as structures

$\text{FO}^+$ Logic: $a$ ranges over $\Sigma$, no $\neg$

$\varphi, \psi := a(x) \mid x \leq y \mid x < y \mid \varphi \lor \psi \mid \varphi \land \psi \mid \exists x. \varphi \mid \forall x. \varphi$
**The FO\(^{+}\) logic, words as structures**

\(\text{FO}^{+}\) Logic: \(a\) ranges over \(\Sigma\), no \(\neg\)

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

Word on alphabet \(A = 2^\Sigma\):

\[
\begin{array}{ccccccc}
\emptyset & \{b\} & \{a, b\} & \emptyset & \{b\} \\
\bullet & \rightarrow & \bullet & \rightarrow & \bullet & \rightarrow & \bullet
\end{array}
\]
The $\text{FO}^+$ logic, words as structures

$\text{FO}^+$ Logic: $a$ ranges over $\Sigma$, no $\neg$

$$\phi, \psi := a(x) \mid x \leq y \mid x < y \mid \phi \lor \psi \mid \phi \land \psi \mid \exists x. \phi \mid \forall x. \phi$$

Word on alphabet $A = 2^\Sigma$:

$$\emptyset \rightarrow \{b\} \rightarrow \{a, b\} \rightarrow \emptyset \rightarrow \{b\}$$

Example: On $\Sigma = \{a, b\}$:

$$\exists x, y. (x \leq y) \land a(x) \land b(y) \leadsto A^*\{a\}A^*\{b\}A^* \cup A^*\{a, b\}A^*$$
The $\text{FO}^+$ logic, words as structures

$\text{FO}^+$ Logic: $a$ ranges over $\Sigma$, no $\neg$

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

Word on alphabet $A = 2^\Sigma$: $\emptyset \longrightarrow \{b\} \longrightarrow \{a, b\} \longrightarrow \emptyset \longrightarrow \{b\}$

Example: On $\Sigma = \{a, b\}$:

$$\exists x, y. (x \leq y) \land a(x) \land b(y) \sim A^*\{a\}A^*\{b\}A^* \cup A^*\{a, b\}A^*$$

Remark: $\emptyset^*$ undefinable in $\text{FO}^+$ (cannot say "$\neg a$").

The \( \text{FO}^+ \) logic, words as structures

\( \text{FO}^+ \) Logic: \( a \) ranges over \( \Sigma \), no \( \neg \)

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

\[ \emptyset \quad \{b\} \quad \{a, b\} \quad \emptyset \quad \{b\} \]

Word on alphabet \( A = 2^\Sigma \):

\[ \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \]

Example: On \( \Sigma = \{a, b\} \):

\[ \exists x, y. (x \leq y) \land a(x) \land b(y) \sim A^*\{a\}A^*\{b\}A^* \cup A^*\{a, b\}A^* \]

Remark: \( \emptyset^* \) undefinable in \( \text{FO}^+ \) (cannot say "\(-a\".")

More generally: \( \text{FO}^+ \) can only define monotone languages:

\[ u\alpha v \in L \text{ and } \alpha \subseteq \beta \Rightarrow u\beta v \in L \]
The $\text{FO}^+$ logic, words as structures

$\text{FO}^+$ Logic: $a$ ranges over $\Sigma$, no $\neg$

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

Word on alphabet $A = 2^\Sigma$: $\emptyset \rightarrow \{b\} \rightarrow \{a, b\} \rightarrow \emptyset \rightarrow \{b\}$

Example: On $\Sigma = \{a, b\}$:

$$\exists x, y. (x \leq y) \land a(x) \land b(y) \iff A^* \{a\} A^* \{b\} A^* \cup A^* \{a, b\} A^*$$

Remark: $\emptyset^*$ undefinable in $\text{FO}^+$ (cannot say "$\neg a$").

More generally: $\text{FO}^+$ can only define monotone languages:

$$u\alpha v \in L \text{ and } \alpha \subseteq \beta \Rightarrow u\beta v \in L$$

Question [Colcombet]: FO & monotone $\xRightarrow{?} \text{FO}^+$
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 $A = \{\emptyset, a, b, c, (a)_b, (b)_c, (c)_a, (a)_b, (b)_c, (c)_a\}$. Let $a^\uparrow = \{a, (a)_b, (c)_a\}$. 
A counter-example language

Our first result

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

Alphabet $A = \{\emptyset, a, b, c, (a)_b, (b)_c, (c)_a, (a)_b (c)_a\}$. Let $a^\uparrow = \{a, (a)_b, (c)_a\}$.

Language $L = (a^\uparrow b^\uparrow c^\uparrow)^* \cup A^* \left(\begin{pmatrix} a \\ b \\ c \end{pmatrix}\right) A^*$. 
A counter-example language

Our first result

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

Alphabet $A = \{\emptyset, a, b, c, (a_b), (b_c), (c_a), (a_b c_a)\}$. Let $a^\uparrow = \{a, (a_b), (c_a)\}$.

Language $L = (a^\uparrow b^\uparrow c^\uparrow)^* \cup A^* \left(\frac{a}{b c}\right) A^*$.

Lemma: $L$ is FO-definable.

Proof: is counter-free. (no cycle labelled $u \geq 2$)
A counter-example language

Our first result

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

Alphabet $A = \{\emptyset, a, b, c, (\frac{a}{b}), (\frac{b}{c}), (\frac{c}{a}), (\frac{a}{b}, \frac{b}{c}, \frac{c}{a})\}$. Let $a^\uparrow = \{a, (\frac{a}{b}), (\frac{c}{a})\}$.

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

Lemma: $L$ is FO-definable.

Proof: $a^\uparrow$ is counter-free. (no cycle labelled $u \geq 2$)

To prove $L$ is not $\text{FO}^+$-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 $\iff$ 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:

- $u = a2^k \in (aa)^*$: $a a a a a a a a a a$
- $v = a2^k - 1 \not\in (aa)^*$: $a a a a a a a a a$
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 $(aa)^\ast$ is not FO-definable:

- $u = a_k \in (aa)^\ast$: $a a a a a a a a a a$
- $v = a_k{-1} \not\in (aa)^\ast$: $a a a a a a a a a$
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 \( u \in L, v \notin L \) s.t. \( u \equiv_n v \).
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:
\[
\begin{align*}
u &= a^{2k} & \in (aa)^* : & a \ a \ a \ a \ a \ a \ a \ a \ a \\
v &= a^{2k-1} & \notin (aa)^* : & a \ a \ a \ a \ a \ a \ a \ a \ a
\end{align*}
\]
Proving $\text{FO}^+-\text{undefinability}$

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

**New rule:**
Letters in $u$ just have to be included in corresponding ones in $v$.

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

Theorem (Correctness of EF$^+$ games)

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

[Stolboushkin 1995+this work]

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

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

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

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

New rule:
Letters in \( u \) just have to be included in 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]
Proving $\text{FO}^+\text{-undefinability}$

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

New rule:
Letters in $u$ just have to be included in 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**

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

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

First-order logic on arbitrary structures, signature $(P_1, \ldots, P_k)$.

**Theorem (Lyndon 1959)**

Let $\varphi \in \text{FO}$, stable under making predicates true on more tuples. Then $\varphi$ is equivalent to a negation-free formula.

**Example:** If a language of graphs is FO-definable and closed under adding edges, then it is FO-definable without $\lnot$. 

---

Theorem Lyndon’s theorem fails on finite structures:

- [Ajtai, Gurevich 1987]
- [Stolboushkin 1995]
- [This work]

EF games on grid-like structures, involved

EF games on words, elementary thanks to $L$. 


**Background: Lyndon’s theorem**

First-order logic on arbitrary structures, signature \((P_1, \ldots, P_k)\).

**Theorem (Lyndon 1959)**

Let \(\varphi \in \text{FO}\), stable under making predicates true on more tuples. Then \(\varphi\) is equivalent to a negation-free formula.

**Example:** If a language of graphs is FO-definable and closed under adding edges, then it is FO-definable without \(\neg\).

**Theorem**

Lyndon’s theorem fails on finite structures:

- [Ajtai, Gurevich 1987]
  lattices, probabilities, number theory, topology, very hard

- [Stolboushkin 1995]
  EF games on grid-like structures, involved
Background: Lyndon’s theorem

First-order logic on arbitrary structures, signature \((P_1, \ldots, P_k)\).

**Theorem (Lyndon 1959)**

Let \(\varphi \in \text{FO}\), stable under making predicates true on more tuples. Then \(\varphi\) is equivalent to a negation-free formula.

**Example:** If a language of graphs is FO-definable and closed under adding edges, then it is FO-definable without \(\neg\).

**Theorem**

Lyndon’s theorem fails on finite structures:

- [Ajtai, Gurevich 1987]
  lattices, probabilities, number theory, topology, very hard

- [Stolboushkin 1995]
  EF games on grid-like structures, involved

- [This work]
  EF games on words, elementary thanks to L
Can we decide \( \text{FO}^+ \)-definability?

**Theorem**

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

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

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

**Theorem**

Given \( L \) regular 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 $\text{FO}^+-$definability?

**Theorem**

*Given* $L$ regular 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.

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

$$M \text{ mortal } \iff L \text{ is } \mathsf{FO}^+\text{-definable.}$$
**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 $(a^\uparrow b^\uparrow c^\uparrow)^*$, but:

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

- All transitions of $M$ follow the cycle: $C_1 \leftarrow C_2 \rightarrow C_3$

- $(a_b), (b_c), (c_a) \rightsquigarrow (u_1^u_2)$, exists iff $u_1 \xrightarrow{M} u_2$. 

Undecidability proof sketch

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

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

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: $C_1 \leftarrow \leftarrow C_3$

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

We choose

$$L := (C_1^+ \cdot C_2^+ \cdot C_3^+)^*$$
Undecidability proof sketch

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

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

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:

$$C_1 \leftarrow C_2 \rightarrow C_3$$

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

We choose

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

$u \in L \nRightarrow u$ 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 :

\[
\begin{align*}
&u \in L : 
&\ u_1 \ u_2 \ u_3 \ \ldots \ u_{n-1} \ u_n \\
&v \notin L : 
&\ (u_1 \ \ u_2) \ (u_2 \ \ u_3) \ (u_3 \ \ u_4) \ \ldots \ (u_{n-1} \ \ u_n)
\end{align*}
\]

\( \rightarrow L \) 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:

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

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

\[
\begin{align*}
u \in L : & \quad u_1 \ u_2 \ u_3 \ \ldots \ u_{n-1} \ u_n \\
v \notin L : & \quad (u_1 \ u_2) \ (u_2 \ u_3) \ (u_3 \ u_4) \ \ldots \ (u_{n-1} \ u_n)
\end{align*}
\]

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

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 $n = 5$):

<table>
<thead>
<tr>
<th>$u$</th>
<th>2</th>
<th>3</th>
<th>2</th>
<th>4</th>
<th>3</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>4</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v$</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>
The reduction

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

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

$\rightarrow L$ 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 $n = 5$):

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

 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:**
FO variants without negation 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:
FO variants without negation will often display this behaviour.

Thanks for your attention!