Positive first-order logic on words

Denis Kuperberg

CNRS, LIP, ENS Lyon

Warwick DIMAP Seminar, 03/11/21
This work was presented at LICS 2021

First-order logic, words

The FO^+ logic

A special language

Background: Lyndon's theorem

Undecidability result

First-order logic

Signature:
Predicate symbols $(P_1, \ldots, P_n)$ with arities $k_1, \ldots, k_n$.

Syntax of First-Order Logic on a signature $(P_1, \ldots, P_n)$:

$$\varphi, \psi := P_i(x_1, \ldots, x_{k_i}) \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \neg\varphi \mid \exists x.\varphi \mid \forall x.\varphi$$

First-order logic

Signature:
Predicate symbols $(P_1, \ldots, P_n)$ with arities $k_1, \ldots, k_n$.

Syntax of First-Order Logic on a signature $(P_1, \ldots, P_n)$:

$$\varphi, \psi := P_i(x_1, \ldots, x_{k_i}) \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \neg\varphi \mid \exists x.\varphi \mid \forall x.\varphi$$

Semantics of φ: *Set of Structures* $(X, R_1, \ldots, R_n)$.

First-order logic

Signature:
Predicate symbols $(P_1, \ldots, P_n)$ with arities $k_1, \ldots, k_n$.

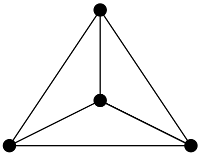
Syntax of First-Order Logic on a signature $(P_1, \ldots, P_n)$:

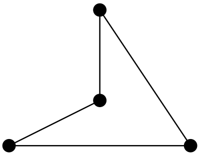
$$\varphi, \psi := P_i(x_1, \ldots, x_{k_i}) \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \neg\varphi \mid \exists x.\varphi \mid \forall x.\varphi$$

Semantics of φ: *Set of Structures* $(X, R_1, \ldots, R_n)$.

Example: For graphs, Signature (E).

- *Cliques*: $\varphi = \forall x.\forall y.E(x, y)$
- *Graphs with no central hub*: $\psi = \neg\exists x.\forall y.E(x, y)$


Model of φ


Model of ψ

FO on words, the usual way

Words on alphabet $A = \{a, b, c, \dots\}$: signature $(\leq, a, b, c, \dots)$

- $x \leq y$ means position x is before position y.
- $a(x)$ means position x is labelled by the letter a

Examples of formulas:

- $\exists x. a(x)$: words containing a. Language $A^* a A^*$.
- $\exists x, y. (x \leq y \wedge a(x) \wedge b(y))$. Language $A^* a A^* b A^*$.

FO on words, the usual way

Words on alphabet $A = \{a, b, c, \dots\}$: signature $(\leq, a, b, c, \dots)$

- $x \leq y$ means position x is before position y.
- $a(x)$ means position x is labelled by the letter a

Examples of formulas:

- $\exists x. a(x)$: words containing a. Language $A^* a A^*$.
- $\exists x, y. (x \leq y \wedge a(x) \wedge b(y))$. Language $A^* a A^* b A^*$.

Remark: We can remove negations from any formula.

FO on words, the usual way

Words on alphabet $A = \{a, b, c, \dots\}$: signature $(\leq, a, b, c, \dots)$

- $x \leq y$ means position x is before position y.
- $a(x)$ means position x is labelled by the letter a

Examples of formulas:

- $\exists x. a(x)$: words containing a. Language A^*aA^*.
- $\exists x, y. (x \leq y \wedge a(x) \wedge b(y))$. Language $A^*aA^*bA^*$.

Remark: We can remove negations from any formula.

Theorem (Schützenberger, McNaughton, Papert)

A language $L \subseteq A^*$ is definable in FO if and only if L is a regular language that "cannot count modulo".

FO on words, the usual way

Words on alphabet $A = \{a, b, c, \dots\}$: signature $(\leq, a, b, c, \dots)$

- $x \leq y$ means position x is before position y.
- $a(x)$ means position x is labelled by the letter a

Examples of formulas:

- $\exists x. a(x)$: words containing a. Language $A^* a A^*$.
- $\exists x, y.(x \leq y \wedge a(x) \wedge b(y))$. Language $A^* a A^* b A^*$.

Remark: We can remove negations from any formula.

Theorem (Schützenberger, McNaughton, Papert)

A language $L \subseteq A^*$ is definable in FO if and only if L is a regular language that "cannot count modulo".

L "cannot count modulo": there is $n \in \mathbb{N}$ such that

$$uv^n w \in L \Leftrightarrow uv^{n+1} w \in L.$$

FO on words, the usual way

Words on alphabet $A = \{a, b, c, \dots\}$: signature $(\leq, a, b, c, \dots)$

- $x \leq y$ means position x is before position y.
- $a(x)$ means position x is labelled by the letter a

Examples of formulas:

- $\exists x. a(x)$: words containing a. Language $A^* a A^*$.
- $\exists x, y.(x \leq y \wedge a(x) \wedge b(y))$. Language $A^* a A^* b A^*$.

Remark: We can remove negations from any formula.

Theorem (Schützenberger, McNaughton, Papert)

A language $L \subseteq A^*$ is definable in FO if and only if L is a regular language that "cannot count modulo".

L "cannot count modulo": there is $n \in \mathbb{N}$ such that

$$uv^n w \in L \Leftrightarrow uv^{n+1} w \in L.$$

Characterization: "Counter-free" automata

FO on words, the "unconstrained" way

Classically, restrict structures $(X, \leq, a, b, \dots)$ to those where

- $\leq$ is a total order
- $a, b, \dots$ form a partition of X.

FO on words, the "unconstrained" way

Classically, restrict structures $(X, \leq, a, b, \dots)$ to those where

- $\leq$ is a total order
- ~~$a, b, \dots$ form a partition of X.~~

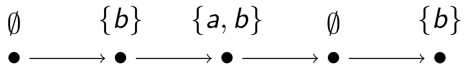
Let us drop the second constraint: $a, b, \dots$ independent.

FO on words, the "unconstrained" way

Classically, restrict structures $(X, \leq, a, b, \dots)$ to those where

- $\leq$ is a total order
- ~~$a, b, \dots$ form a partition of X.~~

Let us drop the second constraint: $a, b, \dots$ independent.
$\rightarrow$ Words on alphabet $\mathcal{P}(\{a, b, \dots\})$:



We will note $\Sigma = \{a, b, \dots\}$, and $A = 2^{\Sigma}$ the alphabet.

- Useful e.g. in verification: independent signals can be true or false simultaneously.

FO on words, the "unconstrained" way

Classically, restrict structures $(X, \leq, a, b, \dots)$ to those where

- $\leq$ is a total order
- ~~$a, b, \dots$ form a partition of X.~~

Let us drop the second constraint: $a, b, \dots$ independent.
$\rightarrow$ Words on alphabet $\mathcal{P}(\{a, b, \dots\})$:

$$\overset{\emptyset}{\bullet} \longrightarrow \overset{\{b\}}{\bullet} \longrightarrow \overset{\{a,b\}}{\bullet} \longrightarrow \overset{\emptyset}{\bullet} \longrightarrow \overset{\{b\}}{\bullet}$$

We will note $\Sigma = \{a, b, \dots\}$, and $A = 2^{\Sigma}$ the alphabet.

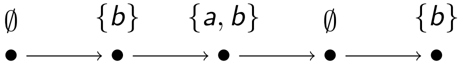
- Useful e.g. in verification: independent signals can be true or false simultaneously.
- FO languages on alphabet A are the same in both versions.

FO on words, the "unconstrained" way

Classically, restrict structures $(X, \leq, a, b, \dots)$ to those where

- $\leq$ is a total order
- ~~$a, b, \dots$ form a partition of X.~~

Let us drop the second constraint: $a, b, \dots$ independent.
$\rightarrow$ Words on alphabet $\mathcal{P}(\{a, b, \dots\})$:



We will note $\Sigma = \{a, b, \dots\}$, and $A = 2^{\Sigma}$ the alphabet.

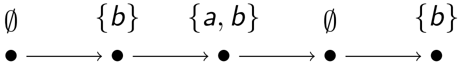
- Useful e.g. in verification: independent signals can be true or false simultaneously.
- FO languages on alphabet A are the same in both versions.
- We can no longer remove negations from all formulas.

FO on words, the "unconstrained" way

Classically, restrict structures $(X, \leq, a, b, \dots)$ to those where

- $\leq$ is a total order
- ~~$a, b, \dots$ form a partition of X.~~

Let us drop the second constraint: $a, b, \dots$ independent.
$\rightarrow$ Words on alphabet $\mathcal{P}(\{a, b, \dots\})$:



We will note $\Sigma = \{a, b, \dots\}$, and $A = 2^{\Sigma}$ the alphabet.

- Useful e.g. in verification: independent signals can be true or false simultaneously.
- FO languages on alphabet A are the same in both versions.
- We can no longer remove negations from all formulas.
- **This work:** study the negation-free fragment.

The FO^+ logic

FO^+ Logic: a ranges over Σ, no $\neg$

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

The FO^+ logic

FO^+ Logic: a ranges over Σ, no $\neg$

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

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

$$\exists x, y.(x \leq y) \wedge a(x) \wedge b(y) \;\rightsquigarrow\; A^*\{a\}A^*\{b\}A^* \cup A^*\{a, b\}A^*$$

The FO^+ logic

FO^+ Logic: a ranges over Σ, no $\neg$

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

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

$$\exists x, y.(x \leq y) \wedge a(x) \wedge b(y) \;\rightsquigarrow\; A^*\{a\}A^*\{b\}A^* \cup A^*\{a, b\}A^*$$

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

The FO^+ logic

FO^+ Logic: a ranges over Σ, no $\neg$

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

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

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

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

More generally: FO^+ can only define monotone languages:

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

The FO^+ logic

FO^+ Logic: a ranges over Σ, no $\neg$

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

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

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

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

More generally: FO^+ can only define monotone languages:

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

Motivation: abstraction of many logics not closed under $\neg$.

The FO^+ logic

FO^+ Logic: a ranges over Σ, no $\neg$

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

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

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

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

More generally: FO^+ can only define monotone languages:

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

Motivation: abstraction of many logics not closed under $\neg$.

Question [Colcombet]: FO & monotone $\overset{?}{\Rightarrow}$ 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 $A = \{\emptyset, a, b, c, \binom{a}{b}, \binom{b}{c}, \binom{c}{a}, \begin{pmatrix} a \\ b \\ c \end{pmatrix}\}$. Let $a^\uparrow = \{a, \binom{a}{b}, \binom{c}{a}\}$.

A counter-example language

Our first result

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

Alphabet $A = \{\emptyset, a, b, c, \binom{a}{b}, \binom{b}{c}, \binom{c}{a}, \begin{pmatrix} a \\ b \\ c \end{pmatrix}\}$. Let $a^\uparrow = \{a, \binom{a}{b}, \binom{c}{a}\}$.

Language $\quad L = \; (a^\uparrow b^\uparrow c^\uparrow)^* \;\; \bigcup \;\; A^* \begin{pmatrix} a \\ b \\ c \end{pmatrix} A^*.$

A counter-example language

Our first result

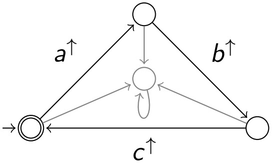
There is L monotone, FO-definable but not FO^+-definable.

Alphabet $A = \{\emptyset, a, b, c, \binom{a}{b}, \binom{b}{c}, \binom{c}{a}, \begin{pmatrix} a \\ b \\ c \end{pmatrix}\}$. Let $a^\uparrow = \{a, \binom{a}{b}, \binom{c}{a}\}$.

Language $\quad L = \ (a^\uparrow b^\uparrow c^\uparrow)^* \ \bigcup \ A^* \begin{pmatrix} a \\ b \\ c \end{pmatrix} A^*.$

Lemma: L is FO-definable.

Proof:



is counter-free. (no cycle labelled $u^{\geq 2}$)

A counter-example language

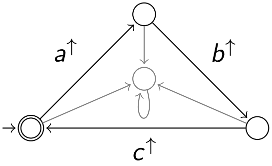
Our first result

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

Alphabet $A = \{\emptyset, a, b, c, \binom{a}{b}, \binom{b}{c}, \binom{c}{a}, \begin{pmatrix} a \\ b \\ c \end{pmatrix}\}$. Let $a^\uparrow = \{a, \binom{a}{b}, \binom{c}{a}\}$.

Language $\quad L = \; (a^\uparrow b^\uparrow c^\uparrow)^* \; \bigcup \; A^* \begin{pmatrix} a \\ b \\ c \end{pmatrix} A^*$.

Lemma: L is FO-definable.

Proof: 

is counter-free. (no cycle labelled $u^{\geq 2}$)

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:

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

Proving FO^+-undefinability

Definition (EF^+ games)

New rule:
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.

Proving 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 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:
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 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

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

Theorem (Lyndon 1959)

Let $\varphi \in \mathrm{FO}$, stable under making predicates true on more tuples. Then φ 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 $\neg$.

Background: Lyndon's theorem

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

Theorem (Lyndon 1959)

Let $\varphi \in \mathrm{FO}$, stable under making predicates true on more tuples.
Then φ 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 $\varphi \in \mathrm{FO}$, stable under making predicates true on more tuples. Then φ 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 FO^+-definability?

Theorem

Given L regular 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 L regular 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 L regular 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

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

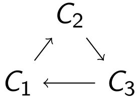
Undecidability proof sketch

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:



- $\binom{a}{b}, \binom{b}{c}, \binom{c}{a} \rightsquigarrow \binom{u_1}{u_2}$, 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:

$$\begin{array}{ccc} & C_2 & \\ \nearrow & & \searrow \\ C_1 & \longleftarrow & C_3 \end{array}$$

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

We choose

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

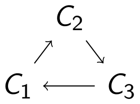
Undecidability proof sketch

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:



- $\binom{a}{b}, \binom{b}{c}, \binom{c}{a} \rightsquigarrow \binom{u_1}{u_2}$, 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 $u_1, u_2, \dots, u_n$ a long run of M, and play Duplicator in :

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

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

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).

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 $n = 5$):

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

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

Encoding the counter-example in Graphs.

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

Encoding the counter-example in Graphs.

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 !