Positive and monotone fragments of FO and LTL

Denis Kuperberg, Quentin Moreau

CNRS, LIP, ENS Lyon, Plume Team

ATLAS, 23 April 2024

First-Order Logic (FO)

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

Syntax of FO:

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

First-Order Logic (FO)

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

Syntax of FO:

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

Semantics of φ:
Structure $(X, R_1, \ldots, R_n)$ is accepted or rejected.

First-Order Logic (FO)

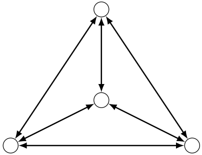
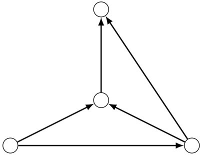
Signature: Predicate symbols $(P_1, \ldots, P_n)$ with arities $k_1, \ldots, k_n$.

Syntax of FO:

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

Semantics of φ:
Structure $(X, R_1, \ldots, R_n)$ is accepted or rejected.

Example: For directed graphs, signature = one binary predicate E.

Graph class	Cliques	No node points to everyone
Formula	$\varphi = \forall x.\forall y.E(x, y)$	$\psi = \neg\exists x.\forall y.E(x, y)$
Example graph	 Model of φ	 Model of ψ

Positive versus Monotone

Goal: Understand the role of negation in FO, any signature.

Positive versus Monotone

Goal: Understand the role of negation in FO, any signature.

Positive formula: no $\neg$

Positive versus Monotone

Goal: Understand the role of negation in FO, any signature.

Positive formula: no $\neg$

Monotone class of structures: closed under adding tuples to relations.

For graph classes: monotone = closed under adding edges.

Example: graphs containing a triangle.

Positive versus Monotone

Goal: Understand the role of negation in FO, any signature.

Positive formula: no $\neg$

Monotone class of structures: closed under adding tuples to relations.

For graph classes: monotone = closed under adding edges.

Example: graphs containing a triangle.

Monotone formula: defines a monotone class of structures.

Positive versus Monotone

Goal: Understand the role of negation in FO, any signature.

Positive formula: no $\neg$

Monotone class of structures: closed under adding tuples to relations.

For graph classes: monotone = closed under adding edges.

Example: graphs containing a triangle.

Monotone formula: defines a monotone class of structures.

Fact: φ positive $\Rightarrow$ φ monotone.

Positive versus Monotone

Goal: Understand the role of negation in FO, any signature.

Positive formula: no $\neg$

Monotone class of structures: closed under adding tuples to relations.

For graph classes: monotone = closed under adding edges.

Example: graphs containing a triangle.

Monotone formula: defines a monotone class of structures.

Fact: φ positive $\Rightarrow$ φ monotone.

What about the converse ?

Positive versus Monotone

Goal: Understand the role of negation in FO, any signature.

Positive formula: no $\neg$

Monotone class of structures: closed under adding tuples to relations.

For graph classes: monotone = closed under adding edges.

Example: graphs containing a triangle.

Monotone formula: defines a monotone class of structures.

Fact: φ positive $\Rightarrow$ φ monotone.

What about the converse ?

Motivation: Logics with fixed points.
Fixed points can only be applied to monotone φ.
Hard to recognize $\rightarrow$ replace by positive φ, syntactic condition.

Lyndon's theorem

Theorem (Lyndon 1959)

If φ is monotone then φ is equivalent to a positive formula.

On graph classes: FO-definable+monotone $\Rightarrow$ FO-definable without $\neg$.

Lyndon's theorem

Theorem (Lyndon 1959)

If φ is monotone then φ is equivalent to a positive formula.

On graph classes: FO-definable+monotone $\Rightarrow$ FO-definable without $\neg$.

⚠ Only true if we accept **infinite** structures.

Lyndon's theorem

Theorem (Lyndon 1959)

If φ is monotone then φ is equivalent to a positive formula.

On graph classes: FO-definable+monotone $\Rightarrow$ FO-definable without $\neg$.

⚠ Only true if we accept **infinite** structures.

What happens if we consider only **finite** structures ?

This was open for 28 years...

Lyndon's theorem

Theorem (Lyndon 1959)

If φ is monotone then φ is equivalent to a positive formula.

On graph classes: FO-definable+monotone $\Rightarrow$ FO-definable without $\neg$.

Only true if we accept **infinite** structures.

What happens if we consider only **finite** structures ?

This was open for 28 years...

Theorem: Lyndon's theorem fails on **finite** structures:

- [Ajtai, Gurevich 1987]
  lattices, probas, number theory, complexity, topology, very hard
- [Stolboushkin 1995]
  EF games on grid-like structures, involved

Lyndon's theorem

Theorem (Lyndon 1959)

If φ is monotone then φ is equivalent to a positive formula.

On graph classes: FO-definable+monotone $\Rightarrow$ FO-definable without $\neg$.

⚠ Only true if we accept **infinite** structures.

What happens if we consider only **finite** structures ?

This was open for 28 years. . .

Theorem: Lyndon's theorem fails on **finite** structures:

- [Ajtai, Gurevich 1987]
  lattices, probas, number theory, complexity, topology, very hard
- [Stolboushkin 1995]
  EF games on grid-like structures, involved
- [K. 2021,2023]
  EF games on words, elementary

FO on words, the usual way

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

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

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

FO on words, the usual way

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

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

- $x \leq y$ 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 $A^* a A^* b A^*$.
- $\neg a(x) \equiv \bigvee_{\beta \neq a} \beta(x)$.

FO on words, the usual way

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

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

- $x \leq y$ 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 $A^*aA^*bA^*$.
- $\neg a(x) \equiv \bigvee_{\beta \neq a} \beta(x)$.

Theorem

First-order languages form a strict subclass of regular languages.

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

Background: FO-definable languages

FO-definable languages are well-understood.

Background: FO-definable languages

FO-definable languages are well-understood.

Theorem (Schützenberger, McNaughton, Papert)

A language $L \subseteq A^*$ is FO-definable iff it is definable by:
Star-free expression $\Leftrightarrow$ LTL $\Leftrightarrow$ counter-free automaton $\Leftrightarrow$ ...

Background: FO-definable languages

FO-definable languages are well-understood.

Theorem (Schützenberger, McNaughton, Papert)

A language $L \subseteq A^*$ is FO-definable iff it is definable by:
Star-free expression $\Leftrightarrow$ LTL $\Leftrightarrow$ counter-free automaton $\Leftrightarrow$ ...

Intuition: FO languages are "Aperiodic": cannot count modulo
L aperiodic: There is $n \in \mathbb{N}$ such that $\forall u, v, w \in A^*$:

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

Background: FO-definable languages

FO-definable languages are well-understood.

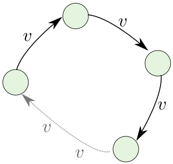
Theorem (Schützenberger, McNaughton, Papert)

A language $L \subseteq A^*$ is FO-definable iff it is definable by:
Star-free expression $\Leftrightarrow$ LTL $\Leftrightarrow$ counter-free automaton $\Leftrightarrow$ ...

Intuition: FO languages are "Aperiodic": cannot count modulo
L aperiodic: There is $n \in \mathbb{N}$ such that $\forall u, v, w \in A^*$:

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

$\Leftrightarrow$ Counter-free automaton: No cycle of the form:



Background: FO-definable languages

FO-definable languages are well-understood.

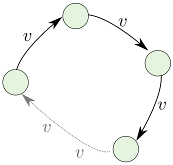
Theorem (Schützenberger, McNaughton, Papert)

A language $L \subseteq A^*$ is FO-definable iff it is definable by:
Star-free expression $\Leftrightarrow$ LTL $\Leftrightarrow$ counter-free automaton $\Leftrightarrow \dots$

Intuition: FO languages are "Aperiodic": cannot count modulo
L aperiodic: There is $n \in \mathbb{N}$ such that $\forall u, v, w \in A^*$:

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

$\Leftrightarrow$ Counter-free automaton: No cycle of the form:



Corollary: FO-definability is decidable for regular languages.

FO on words, the "unconstrained" way

For now, a word is a structure $(X, \leq, a, b, \ldots)$ where

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

FO on words, the "unconstrained" way

For now, a word is a structure $(X, \leq, a, b, \dots)$ 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

For now, a word is a structure $(X, \leq, a, b, \dots)$ 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 $A = \mathcal{P}(\Sigma)$ the alphabet.

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

FO on words, the “unconstrained” way

For now, a word is a structure $(X, \leq, a, b, \dots)$ 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 $A = \mathcal{P}(\Sigma)$ the alphabet.

- Useful e.g. in verification (LTL,…):
  independent signals can be true or false simultaneously.
- FO languages on alphabet A are the same (Preds=Σ or A).

FO on words, the “unconstrained” way

For now, a word is a structure $(X, \leq, a, b, \ldots)$ where

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

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

$$\begin{array}{ccccccccc} \emptyset & & \{b\} & & \{a,b\} & & \emptyset & & \{b\} \\ \bullet & \longrightarrow & \bullet & \longrightarrow & \bullet & \longrightarrow & \bullet & \longrightarrow & \bullet \end{array}$$

We will note $\Sigma = \{a, b, \ldots\}$, and $A = \mathcal{P}(\Sigma)$ the alphabet.

- Useful e.g. in verification (LTL,...):
  independent signals can be true or false simultaneously.
- FO languages on alphabet A are the same (Preds$=\Sigma$ or A).
- We no longer have $\neg a(x) \equiv \bigvee_{\beta \neq a} \beta(x)$.

FO on words, the "unconstrained" way

For now, a word is a structure $(X, \leq, a, b, \dots)$ 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 $\mathcal{P}(\{a, b, \dots\})$:

$$\begin{array}{ccccccccc} \emptyset & & \{b\} & & \{a,b\} & & \emptyset & & \{b\} \\ \bullet & \longrightarrow & \bullet & \longrightarrow & \bullet & \longrightarrow & \bullet & \longrightarrow & \bullet \end{array}$$

We will note $\Sigma = \{a, b, \dots\}$, and $A = \mathcal{P}(\Sigma)$ the alphabet.

- Useful e.g. in verification (LTL,...):
  independent signals can be true or false simultaneously.
- FO languages on alphabet A are the same (Preds$=\Sigma$ or A).
- We no longer have $\neg a(x) \equiv \bigvee_{\beta \neq a} \beta(x)$.
  $\rightarrow$ Negation necessary for full FO.

The FO^+ logic: positive formulas

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: positive formulas

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

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

The FO^+ logic: positive formulas

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

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

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

The FO^+ logic: positive formulas

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

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

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: positive formulas

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

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

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: positive formulas

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

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

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

Theorem [K. 2021,2023]

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

A counter-example language

Theorem [K. 2021,2023]

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

Theorem [K. 2021,2023]

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

A counter-example language

Theorem [K. 2021,2023]

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

A counter-example language

Theorem [K. 2021,2023]

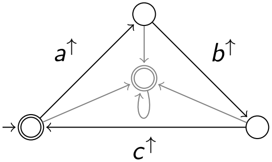
There is L monotone, FO-definable but not FO^+-definable.

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

Lemma: L is FO-definable.

Proof:



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

A counter-example language

Theorem [K. 2021,2023]

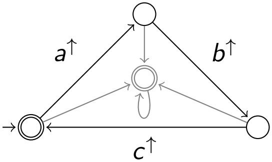
There is L monotone, FO-definable but not FO$^+$-definable.

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

Lemma: L is FO-definable.

Proof:



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

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

Can we decide membership?

Theorem
Given L regular on an ordered alphabet, it is decidable whether

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

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

Can we decide membership?

Theorem

Given L regular on an ordered alphabet, it is decidable whether

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

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

Theorem [K. 2021, 2023]

FO^+-definability is undecidable for regular languages.

Can we decide membership?

Theorem

Given L regular on an ordered alphabet, it is decidable whether

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

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

Theorem [K. 2021, 2023]

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].

Corollaries: lifting the counter-example

Monotone-$\mathrm{FO} \neq \mathrm{FO}^+$, and FO^+ membership undecidable in the following settings:

- Finite graphs, edge predicate [K. 2023]
- Finite structures, arbitrary predicates [K. 2021,2023]

- Words indexed by linear order, finiteness predicate
- Cost functions on finite words, boundedness predicate

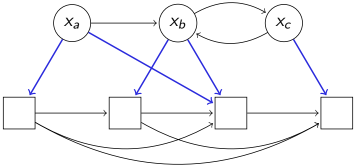
Corollaries: lifting the counter-example

Monotone-$\mathrm{FO} \neq \mathrm{FO}^+$, and FO^+ membership undecidable in the following settings:

- Finite graphs, edge predicate [K. 2023] New
- Finite structures, arbitrary predicates [K. 2021,2023] simpler than [Ajtai Gurevich 1987, Stolboushkin 1995]
- Words indexed by linear order, finiteness predicate New
- Cost functions on finite words, boundedness predicate contradicts [K. 2011, 2014]

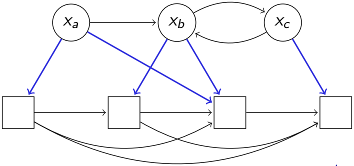
From finite words to finite graphs

Encode words into (directed) graphs, here $ab\binom{a}{b}c$:



From finite words to finite graphs

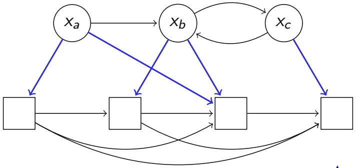
Encode words into (directed) graphs, here $ab\binom{a}{b}c$:



$\rightarrow$ formula ψ_L for graphs encoding words of $L = (a^{\uparrow}b^{\uparrow}c^{\uparrow})^* \cup (A^* \begin{pmatrix} a \\ b \\ c \end{pmatrix} A^*)$.

From finite words to finite graphs

Encode words into (directed) graphs, here $ab\binom{a}{b}c$:



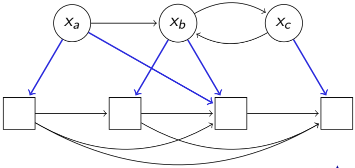
$\rightarrow$ formula ψ_L for graphs encoding words of $L = (a^{\uparrow}b^{\uparrow}c^{\uparrow})^{*} \cup (A^{*} \begin{pmatrix} a \\ b \\ c \end{pmatrix} A^{*})$.

Rule out other graphs, in a monotone way:

- ψ^{-} is a conjunction of edge requirements:

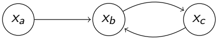
From finite words to finite graphs

Encode words into (directed) graphs, here $ab\binom{a}{b}c$:



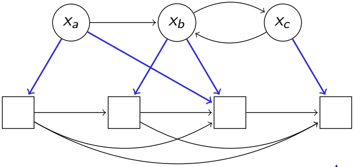
$\rightarrow$ formula ψ_L for graphs encoding words of $L = (a^{\uparrow}b^{\uparrow}c^{\uparrow})^{*} \cup (A^{*} \begin{pmatrix} a \\ b \\ c \end{pmatrix} A^{*})$.

Rule out other graphs, in a monotone way:

- ψ^{-} is a conjunction of edge requirements:
  - $\exists$ 

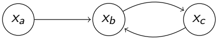
From finite words to finite graphs

Encode words into (directed) graphs, here $ab\binom{a}{b}c$:



$\rightarrow$ formula ψ_L for graphs encoding words of $L = (a^{\uparrow}b^{\uparrow}c^{\uparrow})^{*} \cup (A^{*} \begin{pmatrix} a \\ b \\ c \end{pmatrix} A^{*})$.

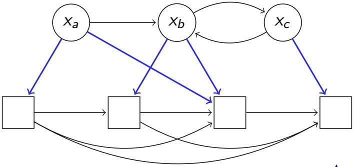
Rule out other graphs, in a monotone way:

- ψ^{-} is a conjunction of edge requirements:
  - $\exists$ 

  - $\forall^{\square} x, y.(x \rightarrow y) \vee (y \rightarrow x)$

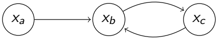
From finite words to finite graphs

Encode words into (directed) graphs, here $ab\binom{a}{b}c$:



$\rightarrow$ formula ψ_L for graphs encoding words of $L = (a^{\uparrow}b^{\uparrow}c^{\uparrow})^* \cup (A^* \begin{pmatrix} a \\ b \\ c \end{pmatrix} A^*)$.

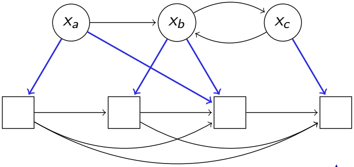
Rule out other graphs, in a monotone way:

- ψ^- is a conjunction of edge requirements:
  - $\exists$ 

  - $\forall^{\square} x, y.(x \rightarrow y) \vee (y \rightarrow x)$
- ψ^+ is a disjunction of excess edges:

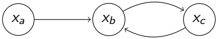
From finite words to finite graphs

Encode words into (directed) graphs, here $ab\binom{a}{b}c$:



$\rightarrow$ formula ψ_L for graphs encoding words of $L = (a^{\uparrow} b^{\uparrow} c^{\uparrow})^* \cup (A^* \begin{pmatrix} a \\ b \\ c \end{pmatrix} A^*)$.

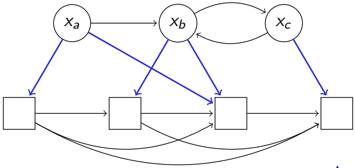
Rule out other graphs, in a monotone way:

- ψ^- is a conjunction of edge requirements:
  - $\exists$ 

  - $\forall^{\square} x, y.(x \rightarrow y) \vee (y \rightarrow x)$
- ψ^+ is a disjunction of excess edges:
  - ,

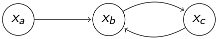
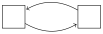
From finite words to finite graphs

Encode words into (directed) graphs, here $ab\binom{a}{b}c$:



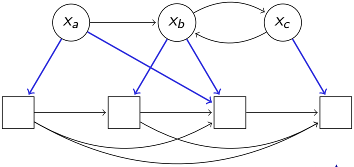
$\rightarrow$ formula ψ_L for graphs encoding words of $L = (a^{\uparrow}b^{\uparrow}c^{\uparrow})^* \cup (A^* \begin{pmatrix} a \\ b \\ c \end{pmatrix} A^*)$.

Rule out other graphs, in a monotone way:

- ψ^- is a conjunction of edge requirements:
  - $\exists$ 
  - $\forall^{\square} x, y.(x \rightarrow y) \vee (y \rightarrow x)$
- ψ^+ is a disjunction of excess edges:
  - ,
  - ,...

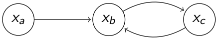
From finite words to finite graphs

Encode words into (directed) graphs, here $ab\binom{a}{b}c$:

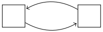


$\rightarrow$ formula ψ_L for graphs encoding words of $L = (a^\uparrow b^\uparrow c^\uparrow)^* \cup (A^* \begin{pmatrix} a \\ b \\ c \end{pmatrix} A^*)$.

Rule out other graphs, in a monotone way:

- ψ^- is a conjunction of edge requirements:
  - $\exists$ 

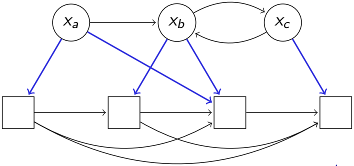
  - $\forall^\square x, y.(x \to y) \vee (y \to x)$
- ψ^+ is a disjunction of excess edges:
  - 

,
  -  ,...

Final Formula: $\exists x_a, x_b, x_c.(\psi^- \wedge (\psi_L \vee \psi^+))$

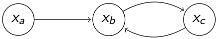
From finite words to finite graphs

Encode words into (directed) graphs, here $ab\binom{a}{b}c$:

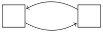


$\rightarrow$ formula ψ_L for graphs encoding words of $L = (a^{\uparrow}b^{\uparrow}c^{\uparrow})^* \cup (A^* \begin{pmatrix} a \\ b \\ c \end{pmatrix} A^*)$.

Rule out other graphs, in a monotone way:

- ψ^- is a conjunction of edge requirements:
  - $\exists$ 

  - $\forall^{\square} x, y.(x \rightarrow y) \vee (y \rightarrow x)$
- ψ^+ is a disjunction of excess edges:
  - 

,
  - 
,...

Final Formula: $\exists x_a, x_b, x_c.(\psi^- \wedge (\psi_L \vee \psi^+))$

Left as exercise: Same with undirected graphs.

Back to words: Link with LTL

LTL syntax:

$$\varphi, \psi ::= \bot \mid \top \mid a \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \mathrm{X}\varphi \mid \varphi \mathrm{U} \psi \mid \varphi \mathrm{R} \psi \mid \neg \varphi.$$

UTL syntax:

$$\varphi, \psi ::= \bot \mid \top \mid a \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \mathrm{X}\varphi \mid \mathrm{Y}\phi \mid \mathrm{P}\varphi \mid \mathrm{F}\varphi \mid \mathrm{H}\varphi \mid \mathrm{G}\varphi \mid \neg \varphi.$$

Back to words: Link with LTL

LTL syntax:

$$\varphi, \psi ::= \bot \mid \top \mid a \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \mathrm{X}\varphi \mid \varphi \mathrm{U} \psi \mid \varphi \mathrm{R} \psi \mid \neg \varphi.$$

UTL syntax:

$$\varphi, \psi ::= \bot \mid \top \mid a \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \mathrm{X}\varphi \mid \mathrm{Y}\phi \mid \mathrm{P}\varphi \mid \mathrm{F}\varphi \mid \mathrm{H}\varphi \mid \mathrm{G}\varphi \mid \neg \varphi.$$

Theorem

- $\mathrm{FO} = \mathrm{LTL} = \mathrm{FO}_3$ *[Kamp 1968]*
- $\mathrm{FO}_2[S, <] = \mathrm{UTL}$ *[Etessami, Vardi, Wilke 1997]*
- $\mathrm{FO}_2[<] = \mathrm{UTL}[P, F, G, H]$ *[Etessami, Vardi, Wilke 1997]*

Back to words: Link with LTL

LTL syntax:

$$\varphi, \psi ::= \bot \mid \top \mid a \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \mathrm{X}\varphi \mid \varphi \mathrm{U} \psi \mid \varphi \mathrm{R} \psi \mid \neg\varphi.$$

UTL syntax:

$$\varphi, \psi ::= \bot \mid \top \mid a \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \mathrm{X}\varphi \mid \mathrm{Y}\phi \mid \mathrm{P}\varphi \mid \mathrm{F}\varphi \mid \mathrm{H}\varphi \mid \mathrm{G}\varphi \mid \neg\varphi.$$

Theorem

- $\mathrm{FO}^+ = \mathrm{LTL}^+ = \mathrm{FO}_3^+$ *[K.,Moreau]*
- $\mathrm{FO}_2^+[S, <] = \mathrm{UTL}^+$ *[K.,Moreau]*
- $\mathrm{FO}_2^+[<] = \mathrm{UTL}^+[P, F, G, H]$ *[K.,Moreau]*

Refining the counter-example language

What is needed to obtain $\mathrm{FO}^+ \neq \mathrm{FO} \cap$ Monotone ?

Refining the counter-example language

What is needed to obtain $\mathrm{FO}^+ \neq \mathrm{FO} \cap$ Monotone ?

Theorem (K.,Moreau)

There is a counter-example language definable in

- FO *with one unary predicate (instead of* 3*)*
- $\mathrm{FO}[\mathit{between}]$ *: bet(a,x,y) means* $\exists z$ *between* x *and* y *s.t.* $a(z)$*.*

Refining the counter-example language

What is needed to obtain $\mathrm{FO}^+ \neq \mathrm{FO} \cap$ Monotone ?

Theorem (K.,Moreau)

There is a counter-example language definable in

- FO *with one unary predicate (instead of* 3*)*
- $\mathrm{FO}[\mathit{between}]$ *: bet(a,x,y) means* $\exists z$ *between* x *and* y *s.t.* $a(z)$*.*

Theorem (K.,Moreau)

There is no counter-example language definable in $\mathrm{FO}_2[<]$*.*
I.e. $\mathrm{FO}_2[<] \cap$ *Monotone* $\subset \mathrm{FO}^+$*.*

Further work

Open problems::

- $\mathrm{FO}_2 \cap \text{Monotone} \stackrel{?}{=} \mathrm{FO}_2^+$
- For which fragments $F \subset \mathrm{FO}$: $\quad F \cap \text{Monotone} = F^+$
- Other kind of counterexamples ?

Further work

Open problems::

- $\mathrm{FO}_2 \cap$ Monotone $\stackrel{?}{=}$ FO_2^+
- For which fragments $F \subset \mathrm{FO}$: $F \cap$ Monotone $=$ F^+
- Other kind of counterexamples ?

Thanks for your attention !