Positive first-order logic on words and graphs

Denis Kuperberg

CNRS, LIP, ENS Lyon, Plume Team

Highlights of Logic, Games and Automata
26 July 2023
First-Order Logic (FO)

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

**Syntax** of FO:

\[
\varphi, \psi := P_i(x_1, \ldots, x_{k_i}) \mid \varphi \lor \psi \mid \varphi \land \psi \mid \neg \varphi \mid \exists x. \varphi \mid \forall x. \varphi
\]
First-Order Logic (FO)

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

**Syntax** of FO:

\[ \varphi, \psi :\begin{align*} & P_i(x_1, \ldots, x_{k_i}) \mid \varphi \lor \psi \mid \varphi \land \psi \mid \neg \varphi \mid \exists x. \varphi \mid \forall x. \varphi \end{align*} \]

**Semantics** of \(\varphi\):
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:
\[
\varphi, \psi : \equiv \ P_i(x_1, \ldots, x_{k_i}) \mid \varphi \lor \psi \mid \varphi \land \psi \mid \neg \varphi \mid \exists x. \varphi \mid \forall x. \varphi
\]

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

**Example**: For directed graphs, signature = one binary predicate \( E \).

<table>
<thead>
<tr>
<th>Graph class</th>
<th>Cliques</th>
<th>No node points to everyone</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Formula</strong></td>
<td>( \varphi = \forall x. \forall y. E(x, y) )</td>
<td>( \psi = \neg \exists x. \forall y. E(x, y) )</td>
</tr>
</tbody>
</table>

**Example graph**

Model of \( \varphi \)  
Model of \( \psi \)  

2/7
Positive versus Monotone

Positive formula: $\text{no } \neg$

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: $\varphi$ positive $\Rightarrow$ $\varphi$ monotone.

What about the converse?

Motivation: Logics with fixed points.
Fixed points can only be applied to monotone $\varphi$.

Hard to recognize $\Rightarrow$ replace by positive $\varphi$, syntactic condition.
Positive versus Monotone

**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

**Positive formula**: no ¬

**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**: \(\phi\) positive \(\Rightarrow\) \(\phi\) monotone.

What about the converse?

**Motivation**: Logics with fixed points. Fixed points can only be applied to monotone \(\phi\).

Hard to recognize \(\Rightarrow\) replace by positive \(\phi\), syntactic condition.
Positive versus Monotone

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: $\varphi$ positive $\Rightarrow$ $\varphi$ monotone.
Positive versus Monotone

**Positive formula:** no \( \lnot \)

**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:** \( \varphi \) positive \( \Rightarrow \) \( \varphi \) monotone.

What about the converse?
Positive versus Monotone

**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**: \( \varphi \) positive \( \Rightarrow \) \( \varphi \) monotone.

What about the converse ?

**Motivation**: Logics with fixed points.
Fixed points can only be applied to monotone \( \varphi \).
Hard to recognize \( \Rightarrow \) replace by positive \( \varphi \), syntactic condition.
Lyndon’s theorem

Theorem (Lyndon 1959)

If $\varphi$ is monotone then $\varphi$ is equivalent to a positive formula.

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

- Ajtai, Gurevich 1987: lattices, probas, number theory, complexity, topology, very hard
- Stolboushkin 1995: EF games on grid-like structures, involved
- This work: EF games on words, elementary
Lyndon’s theorem

Theorem (Lyndon 1959)

If $\varphi$ is monotone then $\varphi$ 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 $\varphi$ is monotone then $\varphi$ 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 \( \varphi \) is monotone then \( \varphi \) 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 $\varphi$ is monotone then $\varphi$ is equivalent to a positive formula.

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

⚠️ 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

- [This work]
  EF games on words, elementary
Positive FO on words

Finite word : structure $(X, \leq, a, b, \ldots)$ where

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

Finite word: structure \((X, \leq, a, b, \ldots)\) where

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

\[
\begin{array}{c}
\emptyset & \{b\} & \{a, b\} & \emptyset & \{b\} \\
\bullet & \rightarrow & \bullet & \rightarrow & \bullet & \rightarrow & \bullet \\
\end{array}
\]
Positive FO on words

Finite word : structure \((X, \leq, a, b, \ldots)\) where

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

\[\rightarrow\] Words on alphabet \(\mathcal{P}\{a, b, \ldots\}\):

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

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

\(\text{FO}^+\): \(\neg a\) forbidden

\(L\) Monotone: \(u\alpha v \in L\) and \(\alpha \subseteq \beta \Rightarrow u\beta v \in L\)
Our results

**Finite Model Theory:**

Lyndon’s theorem **fails** on

- Finite words
- Finite graphs
- Finite structures

**Regular Language Theory:**

- Monotone FO languages $\neq$ Positive FO languages
- Algebraic characterization
- Logical characterization
- Decidable membership
- Undecidable membership
Our results

**Finite Model Theory:**

Lyndon’s theorem fails on

- **Finite words:** \((ABC)^*\)
- Finite graphs
- Finite structures

**Regular Language Theory:**

<table>
<thead>
<tr>
<th>Monotone FO languages</th>
<th>(\neq)</th>
<th>Positive FO languages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algebraic characterization</td>
<td>Logical characterization</td>
<td></td>
</tr>
<tr>
<td>Decidable membership</td>
<td>Undecidable membership</td>
<td></td>
</tr>
</tbody>
</table>
Ongoing work

With Quentin Moreau (internship):
- Link with LTL
- 2-variable fragment

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 Quentin Moreau (internship):
- Link with LTL
- 2-variable fragment

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!