Going Higher in the Quantifier Alternation Hierarchy of First-Order Logic on Words

Thomas Place
Joint work with Marc Zeitoun

LaBRI, Bordeaux University

September 3, 2014
First-Order Logic for Words

We consider first-order logic with only the linear order ‘<.’

\[
\begin{align*}
& a \\ & b \\ & b \\ & b \\ & c \\ & a \\ & a \\ & a \\ & c \\ & a
\end{align*}
\]
We consider first-order logic with only the linear order ‘<.’

A word is as a sequence of labeled positions that can be quantified.

- Unary predicates $a(x), b(x), c(x), \ldots$ testing the label of a position.
- One binary predicate: the linear-order $x < y$. 
We consider first-order logic with only the linear order ‘<.’

- A word is as a sequence of labeled positions that can be quantified.
- Unary predicates $a(x), b(x), c(x), \ldots$ testing the label of a position.
- One binary predicate: the linear-order $x < y$.

\[
\forall x \ a(x) \Rightarrow \exists y \ (b(y) \land (y < x))
\]
Classifying Formulas

What is a simple formula?

\[ \forall x \ a(x) \Rightarrow \exists y \ (b(y) \land (y < x)) \]

What is a complicated formula?

\[ \exists x_1 \ \forall x_2 \ \exists x_3 \ \forall x_4 \ \exists x_5 \ \forall x_6 \ \forall x_7 \ \exists x_8 \ \phi(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8) \Rightarrow \Sigma_{7} \text{ formula } (\phi \text{ quantifier-free}) \]

Complicated = High Quantifier Alternation
Classifying Formulas

What is a simple formula?

\[ \forall x \ a(x) \Rightarrow \exists y \ (b(y) \land (y < x)) \]

What is a complicated formula?

\[ \exists x_1 \ \forall x_2 \ \exists x_3 \ \forall x_4 \ \exists x_5 \ \forall x_6 \ \forall x_7 \ \exists x_8 \ \varphi(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8) \]

Complicated = High Quantifier Alternation
Classifying Formulas

What is a simple formula?

\[ \forall x \ a(x) \Rightarrow \exists y \ (b(y) \land (y < x)) \]

\[ \Rightarrow \ \Pi_2 \ \text{formula.} \]

What is a complicated formula?

\[ \exists x_1 \ \forall x_2 \ \exists x_3 \ \forall x_4 \ \exists x_5 \ \forall x_6 \ \forall x_7 \ \exists x_8 \ \varphi(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8) \]

\[ \Rightarrow \ \Sigma_7 \ \text{formula (\varphi \ \text{quantifier-free})} \]

Complicated = High Quantifier Alternation
Quantifier Alternation of a Formula (2)

**Level $i$: $\Sigma_i$**

For all $i$, a $\Sigma_i$ formula is (in prenex normal form)

\[
\exists x_1, \ldots, x_n \forall y_1, \ldots, y_n \ldots \ldots \phi(\bar{x}, \bar{y}, \ldots)
\]

$i$ blocks (starting with $\exists$)

quantifier-free
Quantifier Alternation of a Formula (2)

Level $i$: $\Sigma_i$

For all $i$, a $\Sigma_i$ formula is (in prenex normal form)

$$\exists x_1, \ldots, x_{n_1} \forall y_1, \ldots, y_{n_2} \cdots \phi(\bar{x}, \bar{y}, \ldots)$$

$i$ blocks (starting with $\exists$) quantifier-free

$\Sigma_i$ is not closed under complement $\Rightarrow$ we get two other classes:

Level $i$: $\Pi_i$

Negation of a $\Sigma_i$ formula:

$$\forall x_1, \ldots, x_{n_1} \exists y_1, \ldots, y_{n_2} \cdots \phi$$

$i$ blocks (starting with $\forall$)

Level $i$: $\mathcal{B}\Sigma_i$

Boolean combinations of $\Sigma_i$ (and $\Pi_i$) formulas.
Problem: understand what part of FO can be expressed with each level.

New Formulation: Membership Problem

Fix a level.

INPUT: A regular language \( L \).

QUESTION: Is \( L \) definable in \( F \)?

Problem can be defined and presented in different ways:

- Logic.
- Regular Expressions.
- Algebra.
Problem: understand what part of FO can be expressed with each level.

⇒ Get an algorithm:

**INPUT:** An FO property.

**QUESTION:** What is the minimal alternation required to state it?
**FO Quantifier Alternation Hierarchy**

**Problem**: understand what part of FO can be expressed with each level.

⇒ Get an algorithm:

**INPUT**: An FO property.

**QUESTION**: What is the minimal alternation required to state it?

**Well-known setting** (Automata)

Good tools. Yeah!

Regular Languages

New Formulation: Membership Problem

Fix a level.

**INPUT**: A regular language \( L \).

**QUESTION**: Is \( L \) definable in \( F \)?
**FO Quantifier Alternation Hierarchy**

New Formulation:
Membership Problem

Fix $\mathcal{F}$ a level.
**INPUT:** A regular language $L$.
**QUESTION:** Is $L$ definable in $\mathcal{F}$?
New Formulation: Membership Problem

Fix $\mathcal{F}$ a level.

**INPUT:** A regular language $L$.

**QUESTION:** Is $L$ definable in $\mathcal{F}$?

Problem can be defined and presented in different ways:

- Logic.
- Regular Expressions.
- Algebra.
FO Quantifier Alternation: Membership State of the Art

How did we make progress? ⇒ By considering a more general problem.

(Schützenberger)'65
(Membership Solved)

(Simon)'75
(Membership Solved)

(Arfi)'87
(Membership Solved)

(Pin, Weil)'95
(Membership Solved)

(P., Zeitoun)'14
(Membership Solved)
FO Quantifier Alternation: Membership State of the Art

(Simon)'75

(Schützenberger)'65

Membership Solved

How did we make progress?

⇒ By considering a more general problem.
FO Quantifier Alternation: Membership State of the Art

(Simon)'75

(Arfi)'87

(Pin, Weil)'95

(Schützenberger)'65

Membership Solved

Membership Solved
How did we make progress?

⇒ By considering a more general problem.
For a level $\mathcal{F}$ in the hierarchy, get an algorithm for the following problem:

Take a regular language $L$
Membership Problem

For a level $\mathcal{F}$ in the hierarchy, get an algorithm for the following problem:

Take a regular language $L$

Can it be defined using $\mathcal{F}$?
Given a level $\mathcal{F}$ in the hierarchy, decide the following problem:

Take two regular languages $L_1, L_2$
More General: Here comes Separation

Given a level $F$ in the hierarchy, decide the following problem:

Take two regular languages $L_1, L_2$

Can $L_1$ be separated from $L_2$ with a $F$ formula?

[Diagram of two automata $L_1$ and $L_2$ with transitions labeled by $a$ and $b$]
Given a level $\mathcal{F}$ in the hierarchy, decide the following problem:

Take two regular languages $L_1, L_2$

Can $L_1$ be separated from $L_2$ with a $\mathcal{F}$ formula?
More General: Here comes Separation

Given a level $\mathcal{F}$ in the hierarchy, decide the following problem:

Take two regular languages $L_1, L_2$

Can $L_1$ be separated from $L_2$ with a $\mathcal{F}$ formula?

Membership can be formally reduced to separation
More General: Here comes Separation

Given a level $\mathcal{F}$ in the hierarchy, decide the following problem:

Take two regular languages $L_1, L_2$

Can $L_1$ be separated from $L_2$ with a $\mathcal{F}$ formula?

Membership can be formally reduced to separation.
A more General Problem than Membership

$L_1$  

$L_2$
A more General Problem than Membership

Problem: Extract enough $\mathcal{F}$-related information from $\mathcal{A}$ to answer $\mathcal{F}$-separation for any pair of accepted languages.
A more General Problem than Membership

We can build a single automaton $\mathcal{A}$ for both

We need to extract enough $\mathcal{F}$-related information from $\mathcal{A}$ to answer $\mathcal{F}$-separation for any pair of accepted languages.

Membership: Asks whether everything about the input can be defined in $\mathcal{F}$.

Separation: Asks what information about the input can be defined in $\mathcal{F}$. 

Problem: Extract enough $\mathcal{F}$-related information from $\mathcal{A}$ to answer $\mathcal{F}$-separation for any pair of accepted languages.
A more General Problem than Membership

We can build a single automaton $A$ for both $L_1$ and $L_2$.

Main idea for $\Sigma_3$ and $B\Sigma_2$ membership:
1) Compute the $\Sigma_2$ and $\Pi_2$ information of the input.
2) Use it in the membership algorithm. (transfer result)

Membership: Asks whether everything about the input can be defined in $\mathcal{F}$.
Separation: Asks what information about the input can be defined in $\mathcal{F}$. 
By relying on $\Sigma_2$-Separation techniques, one can solve membership for $B\Sigma_2$, $\Sigma_3$ and $\Pi_3$. 
Results

We have a solution for the following problems:

1. Separation for $\Sigma_2$ and $\Pi_2$.
2. Membership for $\Sigma_3$.
3. Membership for $B\Sigma_2$.

Proof is essentially combinatorial, tools used:

- Myhill-Nerode equivalence, monoid definition of regular languages.
- Two Combinatorial tools:
  - Ehrenfeucht-Fraïssé Games.
  - Simon’s Factorization Forest Theorem.
Fix a level $\mathcal{F}$ in the hierarchy.

<table>
<thead>
<tr>
<th>Level 1</th>
<th>Solve the Membership Problem for $\mathcal{F}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 2</td>
<td>Solve the Separation Problem for $\mathcal{F}$</td>
</tr>
</tbody>
</table>

Level 2 for $\Sigma_i$ gives Level 1 for $\Sigma_{i+1}$

What is needed to get Level 2 for $\Sigma_{i+1}$
What Now?

Fix a level $\mathcal{F}$ in the hierarchy.

<table>
<thead>
<tr>
<th>Level 1</th>
<th>Solve the Membership Problem for $\mathcal{F}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 2</td>
<td>Solve the Separation Problem for $\mathcal{F}$</td>
</tr>
<tr>
<td>Above</td>
<td>Possible to define deeper problems for $\mathcal{F}$</td>
</tr>
</tbody>
</table>

In particular: needed to get the transfer result

$$\Sigma_2 \Rightarrow B\Sigma_2$$
Thank You