The Expressive Power of Two-Variable Logic on Words

Howard Straubing, Boston College

Simons Institute for the Theory of Computing
November 7, 2016

$FO[<]$

Formulas of first-order logic interpreted in words over a fixed finite alphabet A.

'There are two positions containing a with no positions between them.' (i.e., there is a pair of consecutive a's).

$$\exists x \exists y \Big(x < y \wedge a(x) \wedge a(y) \wedge \neg \exists z (x < z \wedge z < y) \Big)$$

If the input alphabet is $\{a, b\}$, this sentence *defines* the regular language $(a+b)^* aa (a+b)^*$

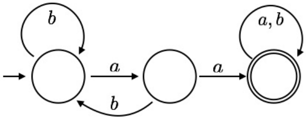
Some facts about $FO[<]$:

- *(Regularity)* Every language in $FO[<]$ is regular.
- *(Alternative characterization in temporal logic)* $L \subseteq A^*$ is in $FO[<]$ if and only if L is definable by a formula of LTL (linear propositional temporal logic). *[Kamp]*
- *(Hierarchy)* $FO[<]$ contains languages of arbitrarily large quantifier alternation depth if $|A| \geq 2$. (i.e., for all k, $\Sigma_k[<] \subsetneq FO[<]$.) *[Brzozowski-Knast]*
- *(Deciding expressibility)* There is an **algebraic decision procedure** for determining if a given regular language is definable in $FO[<]$. *[Schützenberger]*

The algebraic decision procedure.

Syntactic monoid $M(L)$ of regular language $L \subseteq A^* =$ transition semigroup of minimal DFA of L.

Example: $L = (a+b)^* aa(a+b)^*$



$$M(L) = \{1, a = aba, b = b^2 = bab, ab, ba, a^2 = 0\}.$$

L is definable in $FO[<]$ if and only if $M(L)$ contains no nontrivial groups.

Equivalently: $M(L)$ is *aperiodic*, $M(L)$ satisfies the identity $x^\omega x = x^\omega$, where m^ω denotes the idempotent power of $m \in M$. In this example, $x^3 = x^2$ for all $x \in M$.

$FO^2[<]$

- Every sentence of $FO[<]$ is equivalent to one using only three variables. *[Kamp; Immerman and Kozen]*
- $FO^2[<]$ denotes the fragment consisting of formulas using only two variables.
- Example: The language $b^*aa(a+b)^*$ is in $FO^2[<]$:

$$\exists x \Big(a(x) \wedge \exists y (y < x \wedge a(y)) \\ \wedge \forall y \big((y < x \wedge b(y) \rightarrow \forall x (x < y \rightarrow b(x)) \big) \Big)$$

- As we will see, you *cannot* define the language $(a+b)^*aa(a+b)^*$.

Some facts about $FO^2[<]$

(mostly Etessami, Vardi, Wilke, Thérien)

- *(Alternative characterization in temporal logic)* $L \subseteq A^*$ is in $FO^2[<]$ if and only if L is definable in the fragment of *LTL* with only past and future modalities.

$$\mathrm{F}(a \wedge \mathrm{P}a \wedge \neg\mathrm{P}(b \wedge \mathrm{P}a)).$$

- Similar characterizations in terms of one-pebble EF games, two-pebble EF games, 'rankers', 'turtle languages',....
- *(Position in the quantifier alternation hierarchy)* $FO^2[<] \subseteq \Sigma_2[<]$ (in fact $FO^2[<] = \Sigma_2[<] \cap \Pi_2[<]$).
- *(Deciding expressibility)* Algebraic decision procedure for definability: A regular language L is definable in $FO^2[<]$ if and only if $M(L) \in \mathbf{DA}$. (What's that?)

The monoid variety **DA** (Schützenberger)

- *(Equational characterization)* $M \in \mathbf{DA}$ if and only if M satsifies the identity

$$(xy)^\omega x(xy)^\omega = (xy)^\omega.$$

(Many other characterizations in terms of equations, ideal structure, semidirect product decompositions...)

- Example: $L = (a+b)^* aa(a+b)^*$. In $M(L)$, $(ab)^\omega = ab$, $(ab)^\omega a(ab)^\omega = 0 \neq ab$, so $M(L) \notin \mathbf{DA}$. Thus L not definable in $FO^2[<]$.

Quantifier Alternation Depth in $FO^2[<]$.

- The formula

$$\exists x\Big(a(x) \wedge \exists y(y < x \wedge a(y)) \\ \wedge \forall y((y < x \wedge b(y)) \rightarrow \forall x(x < y \rightarrow b(x)))\Big)$$

  has alternation depth 2.
- Is the quantifier alternation depth hierarchy infinite?
- Can one effectively determine the exact quantifier alternation depth of a language in $FO^2[<]$?

Is the quantifier alternation depth hierarchy infinite?

- Yes and No!
- *(Weis and Immerman)* There are languages in $FO^2[<]$ of arbitrarily large alternation depth...
- ..but for each fixed alphabet A, the alternation depth is bounded by $|A| + 1$.

Can one effectively determine the exact quantifier alternation depth of a language in $FO^2[<]$?

- Yes!
- *(Krebs and Straubing, Kufleitner and Weil)*Two different algebraic decision procedures, discovered independently.

System of equations for alternation depth

Set

$$u_1 = (x_1 x_2)^\omega, v_1 = (x_2 x_1)^\omega,$$

and for $n \geq 1$,

$$u_{n+1} = (x_1 \cdots x_{2n} x_{2n+1})^\omega u_n (x_{2n+2} x_1 \cdots x_{2n})^\omega,$$

$$v_{n+1} = (x_1 \cdots x_{2n} x_{2n+1})^\omega v_n (x_{2n+2} x_1 \cdots x_{2n})^\omega.$$

Theorem

$L \subseteq A^*$ is definable in $FO^2[<]$ with quantifier alternation depth $\leq n$ if and only if $M(L)$ is aperiodic and

$$M(L) \models u_n = v_n.$$

‘Dot-depth’

In contrast, computing quantifier alternation depth wrt $FO[<]$ is a long-open problem! A recent breakthrough *(Place, Zeitoun)* decides membership in $\Sigma_3[<]$, maybe $\Sigma_4[<]$, and the boolean closure of $\Sigma_2[<]$.

Strictness of the hierarchy follows from these equations

Recursive definition of congruence $\cong$ on A^*:

- For $w \in A^*$, $\alpha(w) \subseteq A^*$ denotes set of letters in w.
- $w \mapsto (u, a_1, a_2, v)$, where $\alpha(u) \subsetneq \alpha(ua_1) = \alpha(w)$, $\alpha(v) \subsetneq \alpha(a_2 v) = \alpha(w)$. For example, $baabcac \mapsto (baab, c, b, cac)$.
- Let $w \mapsto (u, a_1, a_2, v)$, $w' \mapsto (u', a_1', a_2', v')$. $w \cong w'$ if and only if $a_1 = a_1'$, $a_2 = a_2'$, $u \cong u'$, $v \cong v'$.
- Let $M_A = A^*/\cong$, where $|A| = n$. This is the *free idempotent monoid* on A, and satisfies the identity $x^\omega = x$.

Strictness of the hierarchy follows from these equations

- Easy to define each congruence class by a 2-variable formula with alternation depth $|A|$.
- We have

$$u_1 \cong x_1 x_2 \not\cong x_2 x_1 \cong v_1,$$

  if $A = \{x_1, x_2\}$,

$$u_2 \cong x_1 x_2 x_3 u_1 x_4 x_1 x_2 \not\cong x_1 x_2 x_3 v_1 x_4 x_1 x_2 \cong v_2$$

  if $A = \{x_1, x_2, x_3, x_4\}$ *etc.*
- So if $|A| = 2n$, a congruence class is not definable in $FO^2[<]$ with n alternations.
- Collapse of the hierarchy for fixed A can also be deduced from these equations—if M is generated by n elements then $u_k = v_k$ implies $u_n = v_n$ for $k > n$.

Adding a Successor Relation

- $FO^2[<, +1]$ allows $y = x + 1$ as an atomic formula.
- For example $(a + b)^* aa(a + b)^*$ is now definable by

$$\exists x \exists y (a(x) \wedge a(y) \wedge y = x + 1).$$

- Almost everything works more or less the same way: counterpart in temporal logic, bounded alternation depth wrt $FO[<]$, algebraic decision procedure for definability and for alternation depth, strictness of hierarchy....

Adding a Between Relation *(Krebs, Lodaya, Pandya, Straubing)*

- Roughly speaking, $FO^2[<] \subsetneq FO[<]$ because you cannot say that a position is strictly **between** two other positions.

- What happens if we add to two-variable logic a relation that says ‘there is an a between positions x and y’?

$$a(x, y) \equiv \exists z(x < z \wedge z < y \wedge a(z)).$$

- Example: $(a + b)^* aa(a + b)^*$ defined by

$$\exists x \exists y(x < y \wedge a(x) \wedge a(y) \wedge \neg b(x, y)).$$

- Example: Successor function $y = x + 1$ defined by

$$x < y \wedge \bigwedge_{a \in A} \neg a(x, y).$$

- Notation: $FO^2[<, \text{bet}]$.

Is $FO^2[<, \text{bet}]$ strictly contained in $FO[<]$?

Yes. They are separated by $L = (a(ab)^*b)^*$.

Is the quantifier alternation depth (wrt $FO[<]$) of languages in $FO^2[<, \text{bet}]$ bounded?

No, but the 'No' is qualified.

- Let $A_n = \{0, 1, \wedge_1, \vee_2, \wedge_3, \ldots, \vee_n\}$ (if n even, use $\wedge_n$ if n odd).
- $L_n \subseteq A_n^*$ is set of prefix encodings of depth n boolean circuits, together with input bits, evaluating to 1.
- For each n, $L_n \subseteq FO^2[<, \text{bet}] \backslash \Sigma_n[<]$.
- This requires an alphabet of $n + 2$ letters. If $|A| = 2$ then $FO^2[<, \text{bet}] \subseteq \Sigma_3[<]$, and we conjecture that for each fixed alphabet it is bounded as well.

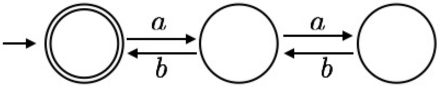
Is there an algebraic decision procedure for definability in $FO^2[<, \text{bet}]$?

Maybe. We have a necessary condition:

- M finite monoid, $m_1, m_2 \in M$. $m_1 \leq_{\mathcal{J}} m_2$ iff $m_1 \in Mm_2M$.
- If $e \in M$ idempotent ($e^2 = e$), M_e denotes submonoid generated by $\{m : e \leq_{\mathcal{J}} m\}$.
- If L is definable, then $eM_ee \in \mathbf{DA}$ for all idempotents e of M.
- This condition is also sufficient for two-letter alphabets—we conjecture that it holds for larger alphabets.

Separation of $FO^2[<, \text{bet}]$ from $FO[<]$

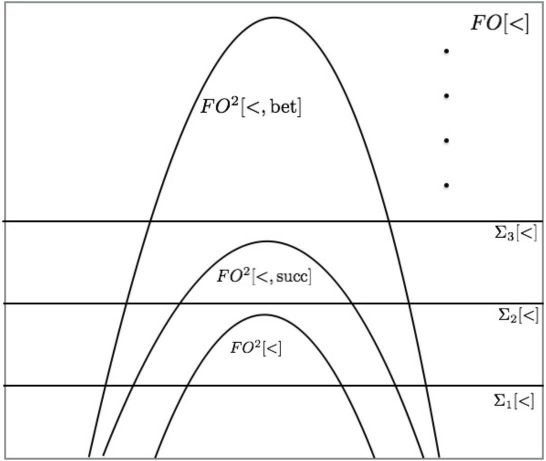
Minimal DFA of $L = (a(ab)^*b)^*$



$e = ba = (ba)^\omega$,

$x = ebe, y = eae \in e \cdot M(L)_e \cdot e$.

$(xy)^\omega$ fixes middle state, $(xy)^\omega x (xy)^\omega$ does not, so $e \cdot M(L)_e \cdot e \notin$ **DA**.


$FO[<]$
$FO^2[<, \mathrm{bet}]$
$\Sigma_3[<]$
$FO^2[<, \mathrm{succ}]$
$\Sigma_2[<]$
$FO^2[<]$
$\Sigma_1[<]$

Are there other equivalent formulations in predicate or temporal logic?

Of course!

For example, we can generalize the new relation to $(a,k)(x,y)$ to mean $x < y$ and there are at least k occurrences of a between x and y.

We call the resulting logic $FO^2[<, \mathrm{Thr}]$. We have (for languages)

$$FO^2[<, \mathrm{Thr}] = FO^2[<, \mathrm{bet}].$$

- However, note that $a(x,y)$ is *not* equivalent to a formula of $FO^2[<, \mathrm{bet}]$ with two free variables!

Are there other equivalent formulations in predicate or temporal logic?

Let $B \subseteq A$. A *simple threshold constraint* is a condition on words of the form $\#B \geq k$, meaning that the word contains at least k occurrences of letters in B.

A *threshold constraint* is a boolean combination of simple threshold constraints.

We can augment the $\{\mathrm{F}, \mathrm{P}\}$ with threshold constraints—if c is such a constraint, we interpret $(w, i) \models \mathrm{F}_c\phi$ to mean that for some $j > i$, $(w, j) \models \phi$ and $w[i+1, j-1]$ satisfies the constraint c.

..and others. For each formulation we find the computational complexity of formula satisfiability. (This version is *EXPSPACE*-complete.)

A Note on the Proofs

- Showing necessity of an equational condition is ‘easy’: Usually this can be done with an EF-game argument.
- Showing sufficiency of an equation is *hard:* Usually this entails showing that satisfaction of the equations implies a semidirect product decomposition of the monoid, and from this it is often possible to extract logical formulas.

Limitations of this approach

This algebraic method is a powerful tool for characterizing the expressive power of logics on *words* that define only *regular languages*.

Extending these methods to regular languages of trees, and to logics that can define non-regular languages, remains a major challenge!