Algebra Meets Logic: The Case of Regular Languages (With Applications to Circuit Complexity)

Denis Thérien

McGill University

Introduction

The following are equivalent:

- Description by regular languages
  - Closure of finite subsets of A^* under $\cup, \cdot, *$
  - Also closed under complement

- Recognition by finite automata
  - $\mathcal{A} = (Q, A, \delta, q_0, F)$
  - $L(\mathcal{A}) = \{w \in A^* : \delta(q_0, w) \in F\}$

Algebra

Monoid: Set M with binary associative operation and identity element.

$L \subseteq A^*$ is *recognized* by the monoid M iff

- there is a morphism $\varphi : A^* \to M$, and
- there is set $F \subseteq M$ such that $L = \varphi^{-1}(F)$.

NB: For each language L there is a canonical monoid $M(L)$ that recognizes it.

Theorem: *L is regular iff $M(L)$ is finite.*

Logic

The model:

- variables stand for positions in words,
- $Q_a x$ is true in word w iff the xth position of w contains letter a,
- numerical predicates have their usual meaning, and
- if Θ is a sentence, $L(\Theta) = \{w \in A^* : w \models \theta\}$.

Theorem: *(Büchi)* L *is regular iff* $L = L(\Theta)$ *for some* Θ *in* $ESOM[+1]$.

Complexity

Turing machines are the traditional model for space/time complexity.

Theorem: *(Shepherdson)* L *is regular iff it can be recognized by a TM operating in constant space.*

Regular languages are also doable in linear time.

No more to say. BUT there are other models, e.g.:

- Boolean circuits
- Communication complexity

Restricting to First-Order

Theorem: *(McNaughton)* L *is star-free iff it is in* $FO[<]$*.*

- $ESOM[+1] = ESOM[<]$.
- $FO[+1] \subset FO[<]$.
- The theorem of McNaughton does not give a decision procedure to test if L is in $FO[<]$.

Theorem: *(Schützenberger)* L *is star-free iff* $M(L)$ *is group-free.*

Base Case: One Variable Only

Can do exactly boolean combinations of $\exists x\,[Q_a x]$.

Let $S = \{0, 1\}$ be the 2-element semilattice:

S	0	1
0	0	1
1	1	1

Let $\varphi : A^* \to S$, with $\varphi(a) = 1$ and $\varphi(b) = 0$.
Then, $w \models \exists x\,[Q_a x]$ iff $\varphi(w) = 1$.

Theorem: *(folklore)* $L \in FO_1[<]$ *iff* $M(L)$ *is a semilattice.*

The General Case: Quantifier Depth k

- L is boolean combination of $\psi = \exists x\,[\Theta(x)]$.
- This normalizes to $\psi = \exists x\,[Q_a x \wedge \Theta_{left}(x) \wedge \Theta_{right}(x)]$.
- Now, $w \models \psi$ iff there exists a factorization $w = uav$ where $u \models \Theta_{left}$ and $v \models \Theta_{right}$.

$L(\psi)$ is recognized by the monoid $S \square T$, where:

- S is a semilattice (to deal with the external quantifier), and
- T is the monoid constructed inductively (to deal with formulas of depth $k-1$).

Theorem: *(Krohn-Rhodes, Schützenberger)* M *is group-free iff* M *divides a block product of semilattices.*

Allowing Modular Quantifiers

$w \models \exists^{c \bmod q} x\,[\Theta(x)]$ iff the number of positions x in w that satisfy $\Theta(x)$ is congruent to $c \bmod q$.

Base case: one variable only

Theorem: *(folklore)* $L \in MOD_1[<]$ *iff* $M(L)$ *is an abelian group.*

General Case: Quantifier Depth k

L is a boolean combination of $\psi = \exists^{c \bmod q} x\,[\Theta(x)]$.

$L(\psi)$ is recognized by the monoid $G \square T$, where:

- G is an abelian group (to deal with the external quantifier), and
- T is the monoid constructed inductively (to deal with formulas of depth $k-1$).

Theorem: *(Jordan-Holder, STT)* $L \in MOD[<]$ *iff* $M(L)$ *is a solvable group.*

$L \in FO + MOD[<]$ iff $M(L) \in \mathbf{M_{sol}}$ (i.e. every group in $M(L)$ is solvable).

Boolean Circuits

AC^0: constant depth, poly size, unbounded AND and OR gates.

CC^0: constant depth, poly size, unbounded MOD$_q$ gates.

ACC^0: constant depth, poly size, unbounded AND, OR, and MOD$_q$ gates.

Known: $AC^0 \subset ACC^0$, Parity $\notin AC^0$.

Conjectured: $CC^0 \subset ACC^0$, AND $\notin CC^0$.

Semi-Obvious Results

- $FO[\textit{arbitrary}] = AC^0$,
- $MOD[\textit{arbitrary}] = CC^0$,
- $FO + MOD[\textit{arbitrary}] = ACC^0$.

Other results:

- $FO[+, *] =$ logtime uniform AC^0.

What about: $FO[+]$, $MOD[+]$, and $FO + MOD[+]$?

Saving on Variables (by Reuse)

Trivial: $FO_1[<] \subset FO[<]$, and
$MOD_1[<] \subset MOD[<]$.

Theorem: *(Immerman - Kozen)* $FO_3[<] = FO[<]$.

Theorem: *(ST)* $MOD_2[<] = MOD[<]$, *and*
$FO + MOD_3[<] = FO + MOD[<]$.

What about $FO_2[<]$ and $FO + MOD_2[<]$?

M is in the variety $\mathbf{DA}$ iff $\forall e = e^2$, $s \in M$, $MeM = MsM \Rightarrow s = s^2$.

Theorem: *(Schützenberger)* L *is a finite union of languages of the form* $L_0 a_1 L_1 \ldots a_s L_s$*, where each* L_i *is a commutative star-free language and the concatenation is unambiguous iff* $M(L)$ *is in* $\mathbf{DA}$*.*

Theorem: *(TW)* $L \in FO_2[<]$ *iff* $M(L)$ *is in* $\mathbf{DA}$*.*

ex: $A^*ac^*aA^*$ is in $FO[<]$ but not in $FO_2[<]$.

ex: $(c^*ac^*bc^*)^*$ is in $FO[<]$ but not in $FO_2[<]$.

ex: $\{b,c,d\}^*bd^*aA^*$ is in $FO_2[<]$.

$$\exists x[Q_a x \wedge \forall y[y < x \Rightarrow \neg Q_a x] \wedge \exists y[y < x \wedge Q_b y \wedge \forall x[y < x \rightarrow Q_d x \vee \exists y[y < x \wedge Q_a y]]]]$$

Theorem: *(ST)* $L \in FO_2[<]$ *iff* $M(L)$ *divides a block product of semilattices bracketed from left to right.*

The variety $\mathbf{A}$ of group-free monoids is the smallest class such that:

- every semilattice is in $\mathbf{A}$, and
- if $M \in \mathbf{A}$ and S is a semilattice, then $S \Box M$ is in $\mathbf{A}$.

The variety $\mathbf{DA}$ is the smallest class such that :

- every semilattice is in $\mathbf{DA}$, and
- if $M \in \mathbf{DA}$ and S is a semilattice then $M \Box S$ is in $\mathbf{DA}$.

Theorem: *(ST)* $L \in FO + MOD_2[<]$ *iff* $M(L)$ *divides* $M \square G$ *where* M *is in* $\mathbf{DA}$ *and* G *is a solvable group.*

Crucial step:

> Any formula in $FO + MOD_2[<]$ is equivalent to a formula in $FO + MOD_2[<]$ such that no existential or universal quantifier appears in the scope of a modular quantifier.

(i.e. we can always push the modular quantifiers inside)

$\mathbf{V} = \mathbf{M_{sol}}$ is the smallest variety such that:

- every commutative monoid is in $\mathbf{V}$, and
- if $M \in \mathbf{V}$ and C is a commutative monoid, then $C \square M$ is in $\mathbf{V}$.

$\mathbf{V} = \mathbf{DA} \square \mathbf{G_{sol}}$ is the smallest variety such that:

- every commutative monoid is in $\mathbf{V}$, and
- M is in $\mathbf{V}$ and C is a commutative monoid, then $M \square C$ is in $\mathbf{V}$.

ex:

- $A^*ac^*aA^*$ is in $FO[<]$ but not in $FO + MOD_2[<]$.

- $(c^*ac^*bc^*)^*$ is in $FO[<]$, not in $FO_2[<]$ but it is in $FO + MOD_2[<]$.

Back to Circuits

Theorem: *(Lautemann-T)* $L \in FO_2[\text{arbitrary}]$ *iff* L *can be recognized by an* AC^0 *circuit with* $O(n)$ *gates.*

Theorem: *(Lautemann-T)* $L \in FO + MOD_2[\text{arbitrary}]$ *iff* L *can be recognized by an* ACC^0 *circuit with* $O(n)$ *gates.*

Conjectured: -Any ACC^0 circuit for $A^*ac^*aA^*$ requires a superlinear number of gates.

-Any AC^0 circuit for $(c^*ac^*bc^*)^*$ requires a superlinear number of gates.

-A regular language L (with a neutral letter) can be decided by an AC^0 circuit with $O(n)$ gates iff $M(L)$ is in $\mathbf{DA}$.

Theorem: *(TT) If the product in M can be computed in ACC^0 with $O(n)$ gates then so can the product of $M \square T$, where T is a commutative monoid.*

Circuits: Counting Wires

Theorem: *(KPT) A regular language (with neutral letter)* L *can be recognized by an* AC^0 *circuit with* $O(n)$ *wires iff* $M(L)$ *is in* DA.

A regular language (with neutral letter) L can be recognized by an ACC^0-circuit with $O(n)$ wires iff $M(L)$ is orthodox and all groups in $M(L)$ are commutative.

Theorem: *(TT)* L *can be denoted by a 2-variable formula in which no modular quantifier appears in the scope of another quantifier iff* $M(L)$ *is orthodox and all groups in* $M(L)$ *are commutative.*

$A^*ac^*aA^*$ is not doable with $O(n)$ wires.

$(c^*ac^*bc^*)^*$ is not doable with $O(n)$ wires.

Key step for the lower bound: Adapt results on superconcentrators to our situation.

Key step for the upper bound: If multiplication in M can be done in constant-depth with $O(n)$ wires, then so can be the multiplication in $M \square S$ for any semilattices S.

Idea: On input $x_1, \ldots x_n$ the circuit needs to compute $OR(x_1, \ldots x_i)$ and $OR(x_i, \ldots x_n)$ for each i. This is possible with $O(n)$ wires!

Conclusion

- The algebra and the logic of regular languages are deeply intertwined.
- The results are often non-trivial and always elegant.
- The results give solid intuition on what to expect e.g. for Boolean circuits. Of course, jacking up theorems from automata land to circuit land is a big challenge...