Finitely typed semigroups

Andreas Krebs

Wilhelm Schickard Institut - Tübingen University

November 8, 2006

Overview

Circuits, Logic, Algebra

AC^0	ACC^0	TC^0	NC^1
FO[$<$,arb]	FO+MOD[$<$,arb]	FO+Maj[$<$,arb]	FO+Grp[$<$,arb]

Circuits, Logic, Algebra

AC^0	ACC^0	TC^0	NC^1
FO[<,arb]	FO+MOD[<,arb]	FO+Maj[<,arb]	FO+Grp[<,arb]
FO[<]	FO+MOD[<]	FO+Maj[<]	FO+Grp[<]
Aperiodic	$M_{solvable}$	???	Finite

Circuits, Logic, Algebra

AC^0	ACC^0	TC^0	NC^1
FO[<,arb]	FO+MOD[<,arb]	FO+Maj[<,arb]	FO+Grp[<,arb]
FO[<]	FO+MOD[<]	FO+Maj[<]	FO+Grp[<]
Aperiodic	$M_{solvable}$	???	Finite

Circuits, Logic, Algebra

FO[<]	FO+MOD[<]	FO+Maj[<,arb]	FO+Grp[<]
Aperiodic	$M_{solvable}$	???	Finite

FO[<], FO+MOD[<], FO+Grp[<] can only recognize regular languages, hence the corresponding monoids are finite.

FO+Maj[<] contains nonregular language, hence we need to look at infinite monoids.

$\mathbb{Z}$ for Majority

Example

ϕ=Maj x $Q_a(x)$
$h : \Sigma^* \rightarrow \mathbb{Z}$
$h(a) = +1$, $h(b) = -1$

$L_\phi = h^{-1}(\mathbb{Z}^+)$

$\mathbb{Z}$ for Majority

Problem

If we build the correct varieties to recognize all languages in TC^0 starting with $\mathbb{Z}$, then this variety will contain an infinite free monoid with > 2 generators. So it can recognize all languages, even all undeciable ones.

Example

Let $G = \mathbb{Z}\Box\mathbb{Z} = (\mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}, \mathbb{Z})$, we pick $x = (\mathbf{0}, 1)$ and $y = (\chi_0, 1)$. Then the map $h : \{a, b\}^* \to G$, with $a \mapsto x$ and $b \mapsto y$ is injective. Hence for any $L \subset \Sigma^*$, there is a subset $T \subset G$ with $L = h^{-1}(T)$.

$\mathbb{Z}$ for Majority

Problem

If we build the correct varieties to recognize all languages in TC^0 starting with $\mathbb{Z}$, then this variety will contain an infinite free monoid with > 2 generators. So it can recognize all languages, even all undeciable ones.

Example

Let $G = \mathbb{Z}\Box\mathbb{Z} = (\mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{Z}, \mathbb{Z})$, we pick $x = (\mathbf{0}, 1)$ and $y = (\chi_0, 1)$. Then the map $h : \{a, b\}^* \rightarrow G$, with $a \mapsto x$ and $b \mapsto y$ is injective. Hence for any $L \subset \Sigma^*$, there is a subset $T \subset G$ with $L = h^{-1}(T)$.

Finitely typed semigroups

Finitely typed semigroup

Let G be a semigroup and $\mathfrak{G}$ be a finite partition of G, then $(G, \mathfrak{G})$ is a finitely typed semigroup.

Recognize a language

$(G, \mathfrak{G})$ recognizes a language $L \subset \Sigma^*$, iff there is a semigroup morphism $h : \Sigma^* \to G$ such that $L = h^{-1}(\mathcal{G})$ where $\mathcal{G} \in \overline{\mathfrak{G}}$.

Finitely typed semigroups

Finitely typed semigroup

Let G be a semigroup and $\mathfrak{G}$ be a finite partition of G, then $(G, \mathfrak{G})$ is a finitely typed semigroup.

Recognize a language

$(G, \mathfrak{G})$ recognizes a language $L \subset \Sigma^*$, iff there is a semigroup morphism $h : \Sigma^* \to G$ such that $L = h^{-1}(\mathcal{G})$ where $\mathcal{G} \in \overline{\mathfrak{G}}$.

Morphisms

Morphism

A morphism $h : (G, \mathfrak{G}) \to (H, \mathfrak{H})$ is a semigroup morphism from $G \to H$ such that $h(\mathcal{G}) \in \overline{\mathfrak{H}} \cap h(G)$ for each $\mathcal{G} \in \mathfrak{G}$.

Surjective, Injective

h is surjective iff the $G \to H$ is surjective.
h is injective iff the $\mathfrak{G} \to \mathfrak{H}$ is injective.

Morphisms

Morphism

A morphism $h : (G, \mathfrak{G}) \to (H, \mathfrak{H})$ is a semigroup morphism from $G \to H$ such that $h(\mathcal{G}) \in \overline{\mathfrak{H}} \cap h(G)$ for each $\mathcal{G} \in \mathfrak{G}$.

Surjective, Injective

h is surjective iff the $G \to H$ is surjective.
h is injective iff the $\mathfrak{G} \to \mathfrak{H}$ is injective.

Sub, Factor, Direct Product

Sub, Factor, Divides

A is a finitely typed submonoid of B iff there is an injective morphism $A \to B$.
A is a finitely typed factor of B iff there is a surjective morphism $B \to A$.
A divides B iff there is a submonoid C of B such that A is a factor of C.

Direct Product

$(A, \mathfrak{A}) \times (B, \mathfrak{B}) = (A \times B, \{\mathcal{A} \times \mathcal{B} \mid \mathcal{A} \in \mathfrak{A}, \mathcal{B} \in \mathfrak{B}\})$

Sub, Factor, Direct Product

Sub, Factor, Divides

A is a finitely typed submonoid of B iff there is an injective morphism $A \rightarrow B$.
A is a finitely typed factor of B iff there is a surjective morphism $B \rightarrow A$.
A divides B iff there is a submonoid C of B such that A is a factor of C.

Direct Product

$(A, \mathfrak{A}) \times (B, \mathfrak{B}) = (A \times B, \{\mathcal{A} \times \mathcal{B} \mid \mathcal{A} \in \mathfrak{A}, \mathcal{B} \in \mathfrak{B}\})$

Normal Block product

$A\Box B=(B \times B \to A, B)$.

$(f, b)(f', b') = (f * b' + b * f', b \cdot b')$,

where $+$ is the pointwise addition in $B \times B \to A$ and $\cdot$ is the multiplication in B, and the action $*$ is defined by $(x * f * y)(b_1, b_2) = f(x * b_1, b_2 * y)$.

New Block product

Like in the finite case, except we only restrict to certain functions in $B \times B \to A$.
The "allowed" functions are a **finite** case distinction where the cases have to be of the following kinds:

- V1 $f(x, y) = v$ iff $x \cdot c \in \mathcal{B}$.
- V2 $f(x, y) = v$ iff $c \cdot y \in \mathcal{B}$.
- V3 $f(x, y) = v$ iff $x \cdot c \cdot y \in \mathcal{B}$.

The types of $A \boxdot B$ are $\{(f, b) \mid f(e_B, e_B) \in \mathcal{A}\}$

Recognize languages

Languages

By definition $(\Sigma^*, \{L, \Sigma^* \setminus L\})$ is a finitely typed monoid.
$L \subseteq \Sigma^*$ is recognized by $(G, \mathfrak{G})$ iff there is an injective morphism from $(\Sigma^*, \{L, \Sigma^* \setminus L\})$ to $(G, \mathfrak{G})$.

Finite finitely typed semigroups

Finite semigroups

We can embed the category of finite semigroups in the category of finitely typed semigroups, if we equip G with the discrete partition. $G \mapsto (G, G)$

Examples

$(U_1, \{0, 1\}) \cong (\mathbb{N}_0, \{0, \mathbb{N}^+\})$
$(\mathbb{Z}_2, \{0, 1\}) \cong (Z, \{even, odd\})$
$(G, \{N, G \setminus N\}) \cong (G/N, \{1, G \setminus \{1\}\})$

Finite Case

FO[<]=Aperiodic

The variety A of all aperiodic monoids is the smallest family of monoids that:

1. $U_1 \in A$
2. $X \in A$ and Y divides X then $Y \in A$
3. $X, Y \in A$ implies $X \times Y \in A$
4. $X, Y \in A$ implies $X \Box Y \in A$

Finite Case

FO+MOD[<]=$M_{solvable}$

The variety M_{solv} of all solvable monoids is the smallest family of monoids that:

1. $U_1, Z_p \in M_{solv}$
2. $X \in M_{solv}$ and Y divides X then $Y \in M_{solv}$
3. $X, Y \in M_{solv}$ implies $X \times Y \in M_{solv}$
4. $X, Y \in M_{solv}$ implies $X \Box Y \in M_{solv}$

Infinite Case

FO+Maj[<]=$T_<$

Let $T_<$ be the smallest variety of finitely typed monoids such that:

1. $(\mathbb{Z}, \{Z_0^-, Z^+\}) \in T_<$
2. $X \in T_<$ and Y divides X then $Y \in T_<$
3. $X, Y \in T_<$ implies $X \times Y \in T_<$
4. $X, Y \in T_<$ implies $X \boxdot Y \in T_<$

Important Classes

FO+Maj[<]= $T_<$

Let $T_<$ be the smallest variety of finitely typed monoids such that:

1. $(\mathbb{Z}, \{Z_0^-, Z^+\}) \in T_<$
2. $X \in T_<$ and Y divides X then $Y \in T_<$
3. $X, Y \in T_<$ implies $X \times Y \in T_<$
4. $X, Y \in T_<$ implies $X \boxdot Y \in T_<$

FO+Maj[<, *sq*]= $T_{<,sq} = uniform TC^0$

FO+Maj[<, *arb*]= $T_{<,arb} = nonuniform TC^0$

Final goal

$$\text{FO+Maj}[<, arb] = T_{<,arb} = nonuniform\,TC^0$$

In order to seperate TC^0 from NC^1 we "only" need to show $A_5 \notin T_{<,arb}$.

Finite Case

FO

FO[<,arb] with 2 variables $\leftrightarrow$ linear circuits with boolean gates

FO[<] with 2 variables $\leftrightarrow$ weak blocked U_1

FO+MOD

FO+MOD[<,arb] with 2 variables $\leftrightarrow$ linear circuits with boolean and modulo gates

FO+MOD[<] with 2 variables $\leftrightarrow$ weak blocked U_1 and Z_p

Maj with 2 variables

Majority Quantifier

$$w \models \text{Maj } x\ \phi \text{ iff } 0 < \sum_{i=1}^{n} \begin{cases} +1 & w_{x=i} \models \phi \\ -1 & w_{x=i} \not\models \phi \end{cases}$$

The majority quantifier has certain disadvantages:

- No neutral element
- Can only count up to n

Extended Majority Quantifier

$$w \models E\ x\ \langle \phi_1, \ldots, \phi_c \rangle \text{ iff } 0 < \sum_{i=1}^{n} \sum_{j=1}^{c} \begin{cases} +1 & w_{x=i} \models \phi_j \\ -1 & w_{x=i} \not\models \phi_j \end{cases}$$

with $c = 1$ the extended majority quantifier is just the majority quantifier.

Maj with 2 variables

Majority Quantifier

$$w \models \text{Maj } x\ \phi \text{ iff } 0 < \sum_{i=1}^{n} \begin{cases} +1 & w_{x=i} \models \phi \\ -1 & w_{x=i} \not\models \phi \end{cases}$$

The majority quantifier has certain disadvantages:

- No neutral element
- Can only count up to n

Extended Majority Quantifier

$$w \models E\ x\ \langle \phi_1, \ldots, \phi_c \rangle \text{ iff } 0 < \sum_{i=1}^{n} \sum_{j=1}^{c} \begin{cases} +1 & w_{x=i} \models \phi_j \\ -1 & w_{x=i} \not\models \phi_j \end{cases}$$

with $c = 1$ the extended majority quantifier is just the majority quantifier.

Maj with 2 variables

Majority Quantifier

$w \models \text{Maj } x \; \phi$ iff $0 < \sum_{i=1}^{n} \begin{cases} +1 & w_{x=i} \models \phi \\ -1 & w_{x=i} \not\models \phi \end{cases}$

The majority quantifier has certain disadvantages:

- No neutral element
- Can only count up to n

Extended Majority Quantifier

$w \models E \; x \; \langle \phi_1, \ldots, \phi_c \rangle$ iff $0 < \sum_{i=1}^{n} \sum_{j=1}^{c} \begin{cases} +1 & w_{x=i} \models \phi_j \\ -1 & w_{x=i} \not\models \phi_j \end{cases}$

with $c = 1$ the extended majority quantifier is just the majority quantifier.

LTC^0

Linear TC^0

- threshold gates
- constant depth
- linear size (number of gates)
- linear fan-in

Circuits $<->$ Logic

$LTC^0 \leftrightarrow$FO+E[$<, arb$] with 2 variables

Proof:
"$\leftarrow$" Each innermost formula looks like $\psi(x) = Maj_y \dots$, build a gate for each value of $x = 1, \dots, n$.
"$\rightarrow$" Label the gates with a variable and a finite set of bits.

Logic < − > Algebra

What we want

FO+E[<] with 2 variables corresponds to the smallest variety closed under weak block product with $(\mathbb{Z}, \{\mathbb{Z}_0^-, \mathbb{Z}^+\})$
$\Rightarrow$ this is too weak

What we tried

FO+E[<] with 2 variables corresponds to the smallest variety closed under w. block product with $(\mathbb{Z}, \{\mathbb{Z}_0^-, \mathbb{Z}^+\}) \Box (\mathbb{Z}, \{\mathbb{Z}_0^-, \mathbb{Z}^+\})$
$\Rightarrow$ this is too strong

What works

FO+E[<] with 2 variables corresponds to a restricted version of the smallest variety closed under weak block product with $(\mathbb{Z}, \{\mathbb{Z}_0^-, \mathbb{Z}^+\}) \Box (\mathbb{Z}, \{\mathbb{Z}_0^-, \mathbb{Z}^+\})$

Logic < − > Algebra

What we want

FO+E[<] with 2 variables corresponds to the smallest variety closed under weak block product with $(\mathbb{Z}, \{\mathbb{Z}_0^-, \mathbb{Z}^+\})$
$\Rightarrow$ this is too weak

What we tried

FO+E[<] with 2 variables corresponds to the smallest variety closed under w. block product with $(\mathbb{Z}, \{\mathbb{Z}_0^-, \mathbb{Z}^+\})\Box(\mathbb{Z}, \{\mathbb{Z}_0^-, \mathbb{Z}^+\})$
$\Rightarrow$ this is too strong

What works

FO+E[<] with 2 variables corresponds to a restricted version of the smallest variety closed under weak block product with $(\mathbb{Z}, \{\mathbb{Z}_0^-, \mathbb{Z}^+\})\Box(\mathbb{Z}, \{\mathbb{Z}_0^-, \mathbb{Z}^+\})$

Logic < − > Algebra

What we want

FO+E[<] with 2 variables corresponds to the smallest variety closed under weak block product with $(\mathbb{Z}, \{\mathbb{Z}_0^-, \mathbb{Z}^+\})$
$\Rightarrow$ this is too weak

What we tried

FO+E[<] with 2 variables corresponds to the smallest variety closed under w. block product with $(\mathbb{Z}, \{\mathbb{Z}_0^-, \mathbb{Z}^+\}) \Box (\mathbb{Z}, \{\mathbb{Z}_0^-, \mathbb{Z}^+\})$
$\Rightarrow$ this is too strong

What works

FO+E[<] with 2 variables corresponds to a restricted version of the smallest variety closed under weak block product with $(\mathbb{Z}, \{\mathbb{Z}_0^-, \mathbb{Z}^+\}) \Box (\mathbb{Z}, \{\mathbb{Z}_0^-, \mathbb{Z}^+\})$

What works

$W_2(\mathbb{Z}^+)$

Let $W_2(\mathbb{Z}^+)$ be the smallest variety closed under weak block product with $(\mathbb{Z}, \{\mathbb{Z}_0^-, \mathbb{Z}^+\})\Box(\mathbb{Z}, \{\mathbb{Z}_0^-, \mathbb{Z}^+\})$, e.g. $M \in W_2(\mathbb{Z}^+)$ implies $M\Box((\mathbb{Z}, \{\mathbb{Z}_0^-, \mathbb{Z}^+\})\Box(\mathbb{Z}, \{\mathbb{Z}_0^-, \mathbb{Z}^+\})) \in W_2(\mathbb{Z}^+)$.

restricted morphisms

Informal definition:
Without the direct product all elements of W_2 look like $(((\mathbb{Z}\Box\mathbb{Z})\Box(\mathbb{Z}\Box\mathbb{Z}))\Box(\mathbb{Z}\Box\mathbb{Z})\dots)$
The second component of $(\mathbb{Z}\Box\mathbb{Z})$ is always the neutral element.

Logic < − > Algebra

$FO+E_2[<] \leftrightarrow$ restricted morphisms to $W_2(\mathbb{Z}^+)$

Proof:
"$\longleftarrow$" Build the formula top-down.
"$\longrightarrow$" Use an infinite version of the block product principle

Logic $< - >$ Algebra

FO+$E_2[<] \leftrightarrow$ restricted morphisms to $W_2(\mathbb{Z}^+)$

Proof:

"$\leftarrow$" Build the formula top-down.

"$\rightarrow$" Use an infinite version of the block product principle

Equivalences

non-uniform LTC^0	
$\mathrm{FO}+\mathrm{E}_2[<, arb]$	$\mathrm{FO}+\mathrm{E}_2[<]$
	$W_2(\mathbb{Z}^+)$

This is also possible for other sets of predicates/types/uniformities.

Equivalences

non-uniform LTC^0	FO[<]-uniform LTC^0
$\mathrm{FO}+\mathrm{E}_2[<, arb]$	$\mathrm{FO}+\mathrm{E}_2[<]$
$W_2(arb)$	$W_2(\mathbb{Z}^+)$

This is also possible for other sets of predicates/types/uniformities.

Equivalences

non-uniform LTC^0	FO[$<$]-uniform LTC^0
FO+$E_2[<, arb]$	FO+$E_2[<]$
$W_2(arb)$	$W_2(\mathbb{Z}^+)$

This is also possible for other sets of predicates/types/uniformities.

A5 not possible

We can show $A_5 \notin$FO+E$_2[<, mon - sl]$
(FO+E$_2[<, mon - sl]$ contains all solvable groups)

Proof: Let $h : \Sigma^* \to T$ be a restricted morphism and $\mathcal{T}$ a type of T such that $L = h^{-1}(\mathcal{T})$. Assume the group $T = T' \square (\mathbb{Z} \square \mathbb{Z})^c$.
1. Construct a new morphism $h' : \Sigma^* \to T$ such that $|\pi_2(h'(\Sigma))| = 1$.
2. Construct a new morphism $h'' : \Sigma^* \to T'$.

A5 not possible

We can show $A_5 \notin$FO+E$_2[<, mon - sl]$
(FO+E$_2[<, mon - sl]$ contains all solvable groups)

Proof: Let $h : \Sigma^* \to T$ be a restricted morphism and $\mathcal{T}$ a type of T such that $L = h^{-1}(\mathcal{T})$. Assume the group $T = T'\Box(\mathbb{Z}\Box\mathbb{Z})^c$.
1. Construct a new morphism $h' : \Sigma^* \to T$ such that $|\pi_2(h'(\Sigma))| = 1$.
2. Construct a new morphism $h'' : \Sigma^* \to T'$.

A5 not possible

We can show $A_5 \notin$FO+E$_2[<, mon - sl]$
(FO+E$_2[<, mon - sl]$ contains all solvable groups)

Proof: Let $h : \Sigma^* \rightarrow T$ be a restricted morphism and $\mathcal{T}$ a type of T such that $L = h^{-1}(\mathcal{T})$. Assume the group $T = T' \Box (\mathbb{Z} \Box \mathbb{Z})^c$.
1. Construct a new morphism $h' : \Sigma^* \rightarrow T$ such that $|\pi_2(h'(\Sigma))| = 1$.
2. Construct a new morphism $h'' : \Sigma^* \rightarrow T'$.

A5 not possible

We can show $A_5 \notin \mathrm{FO}{+}\mathrm{E}_2[<, mon - sl]$
($\mathrm{FO}{+}\mathrm{E}_2[<, mon - sl]$ contains all solvable groups)

Proof: Let $h : \Sigma^* \to T$ be a restricted morphism and $\mathcal{T}$ a type of T such that $L = h^{-1}(\mathcal{T})$. Assume the group $T = T'\Box(\mathbb{Z}\Box\mathbb{Z})^c$.
1. Construct a new morphism $h' : \Sigma^* \to T$ such that $|\pi_2(h'(\Sigma))| = 1$.
2. Construct a new morphism $h'' : \Sigma^* \to T'$.

A5 not possible

We can show $A_5 \notin$FO+E$_2[<, mon - sl]$
(FO+E$_2[<, mon - sl]$ contains all solvable groups)

Proof: Let $h : \Sigma^* \to T$ be a restricted morphism and $\mathcal{T}$ a type of T such that $L = h^{-1}(\mathcal{T})$. Assume the group $T = T'\Box(\mathbb{Z}\Box\mathbb{Z})^c$.
1. Construct a new morphism $h' : \Sigma^* \to T$ such that $|\pi_2(h'(\Sigma))| = 1$.
2. Construct a new morphism $h'' : \Sigma^* \to T'$.

Other Logik families

We can also use finitely typed monoids to characterize other logic classes.

For example:
$\mathrm{MOD}[<, +]$

We use infinite monoids that can compute the predicate and monoids that compute the quantifiers.

Other Logik families

We can also use finitely typed monoids to characterize other logic classes.

For example:
$\mathrm{MOD}[<, +]$

We use infinite monoids that can compute the predicate and monoids that compute the quantifiers.

Open Problems

Use more algebraic properties to show:

A_5 not in T$[<, arb]$ => $TC^0 \neq NC^1$
A_5 not in T$[<, Sq]$ => $uniformTC^0 \neq uniformNC^1$

A_5 not in T$[<]$ => $FO + Maj[<]$
A_5 not in W$_2(arb)$ => $LTC^0 = FO + Maj_2[<, arb] \neq NC^1$

Use finitely typed monoids for other logic families.

Open Problems

Use more algebraic properties to show:

A_5 not in T$[<, arb]$ => $TC^0 \neq NC^1$
A_5 not in T$[<, Sq]$ => $uniform TC^0 \neq uniform NC^1$

A_5 not in T$[<]$ => $FO + Maj[<]$
A_5 not in $W_2(arb)$ => $LTC^0 = FO + Maj_2[<, arb] \neq NC^1$

Use finitely typed monoids for other logic families.

Open Problems

Use more algebraic properties to show:

A_5 not in $\mathrm{T}[<, arb]$ => $TC^0 \neq NC^1$
A_5 not in $\mathrm{T}[<, Sq]$ => $uniform TC^0 \neq uniform NC^1$

A_5 not in $\mathrm{T}[<]$ => $FO + Maj[<]$
A_5 not in $\mathrm{W}_2(arb)$ => $LTC^0 = FO + Maj_2[<, arb] \neq NC^1$

Use finitely typed monoids for other logic families.

Open Problems

Use more algebraic properties to show:

A_5 not in T$[<, arb]$ => $TC^0 \neq NC^1$
A_5 not in T$[<, Sq]$ => $uniformTC^0 \neq uniformNC^1$

A_5 not in T$[<]$ => $FO + Maj[<]$
A_5 not in $W_2(arb)$ => $LTC^0 = FO + Maj_2[<, arb] \neq NC^1$

Use finitely typed monoids for other logic families.