An Algebraic Characterization of NC1

Aidan Evans

Anuj Dawar

University of Cambridge

BCTCS, 14 April 2025

Outline

1. How We Recognize Languages
2. Recognition with Logic
3. NC1 via Logic
4. Recognition with Typed Monoids
5. Simplifying NC1's Logic
    - Going A Step Further
6. NC1 via Algebra
7. Conclusion

How We Recognize Languages

	Machines	**Logic**	**Algebra**
Star-Free Reg.	Counter-free DFAs	FO(<)	Aperiodic Fin. Mon.
Regular Lang.	DFAs, NFAs	MSO(<)	Finite Monoids
TC0	…	Maj(+,×)	cf. Krebs et al. (~2007)
NC1	ALogTime	cf. Barrington et al. (1990)	???
P	poly-time DTMs	FO(<,LFP)	???
NP	poly-time NTMs	ESO	???
PH	p-time c-alt ATMs	SO	???

How We Recognize Languages

	Machines	Logic	Algebra
Star-Free Reg.	Counter-free DFAs	FO(<)	Aperiodic Fin. Mon.
Regular Lang.	DFAs, NFAs	MSO(<)	Finite Monoids
TC0	…	Maj(+,×)	cf. Krebs et al. (~2007)
NC1	ALogTime	cf. Barrington et al. (1990)	???
P	poly-time DTMs	FO(<,LFP)	???
NP	poly-time NTMs	ESO	???
PH	p-time c-alt ATMs	SO	???

} Proven to exist! (NC1, P, NP, PH algebra)

How We Recognize Languages

	Machines	Logic	Algebra
Star-Free Reg.	Counter-free DFAs	FO(<)	Aperiodic Fin. Mon.
Regular Lang.	DFAs, NFAs	MSO(<)	Finite Monoids
TC0	…	Maj(+,×)	cf. Krebs et al. (~2007)
NC1	ALogTime	cf. Barrington et al. (1990)	This presentation!
P	poly-time DTMs	FO(<,LFP)	???
NP	poly-time NTMs	ESO	???
PH	p-time c-alt ATMs	SO	???

} Proven to exist!

Recognition with Logic

- $L \subseteq \Sigma^*,\ w = abaa \in \Sigma^*$
- $w = (\{1, 2, 3, 4\}, <, P_a, P_b)$ where $P_a = \{1, 3, 4\}, P_b = \{2\}$

Recognition with Logic

- $L \subseteq \Sigma^*,\ w = abaa \in \Sigma^*$
- $w = (\{1,2,3,4\}, <, P_a, P_b)$ where $P_a = \{1,3,4\}, P_b = \{2\}$

- A sentence φ using predicates P_a, P_b and the numerical relation $<$ *recognizes* L iff for every $\mathrm{u} \in \Sigma^*$, $\mathrm{u} \vDash \varphi$ iff $\mathrm{u} \in L$

Recognition with Logic

- $L \subseteq \Sigma^*,\ w = abaa \in \Sigma^*$
- $w = (\{1,2,3,4\}, <, P_a, P_b)$ where $P_a = \{1,3,4\}, P_b = \{2\}$

- A sentence φ using predicates P_a, P_b and the numerical relation $<$ *recognizes* L iff for every $\mathrm{u} \in \Sigma^*$, $\mathrm{u} \vDash \varphi$ iff $\mathrm{u} \in L$

- For example, $w \vDash \exists x \forall y (y \geq x \rightarrow P_a y)$ so w is in the language of all strings ending with ‘a’s

NC1 via Logic

- Introduce $FO(+,\times)$ with "monoid multiplication quantifiers":

NC1 via Logic

- Introduce $FO(+,\times)$ with “monoid multiplication quantifiers”:
- Monoid, $(M, \ \cdot)$: a set M and a binary operation $\cdot : M \times M \rightarrow M$ such that $\cdot$ is associative and has an identity

NC1 via Logic

- Introduce $FO(+,\times)$ with "monoid multiplication quantifiers":
- Monoid, $(M, \cdot)$: a set M and a binary operation $\cdot : M \times M \to M$ such that $\cdot$ is associative and has an identity
- Monoid Multiplication Quantifier, $\Gamma_{\gamma}^{M,B}$:

$$\Gamma_{\gamma}^{M,B} x_1 \dots x_l \big(\varphi_1(x_1, \dots, x_l), \dots, \varphi_k(x_1, \dots, x_l)\big)$$

where $M = (M,\cdot)$ is a monoid, $\mathrm{B} \subseteq M$, and $\gamma: \{0,1\}^k \to M$

NC1 via Logic

- Monoid Multiplication Quantifier, $\Gamma_{\gamma}^{M,B}$:

For a word $w = w_1 \dots w_n$,

$\varphi_i^w[a_1, \dots, a_l] = 1$, s.t. $a_j \in \{1, \dots, n\}$,

iff $w \vDash \varphi_i(x_1, \dots, x_l)$ when x_j is assigned a_j,

and 0 otherwise

NC1 via Logic

- Monoid Multiplication Quantifier, $\Gamma_{\gamma}^{M,B}$:

  Then, for $w = w_1 \ldots w_n$,

$$w \vDash \Gamma_{\gamma}^{M,B} x_1 \ldots x_l \big(\varphi_1(x_1, \ldots, x_l), \ldots, \varphi_k(x_1, \ldots, x_l)\big)$$

NC1 via Logic

- Monoid Multiplication Quantifier, $\Gamma_{\gamma}^{M,B}$:

Then, for $w = w_1 \dots w_n$,

$$w \vDash \Gamma_{\gamma}^{M,B} x_1 \dots x_l \big(\varphi_1(x_1, \dots, x_l), \dots, \varphi_k(x_1, \dots, x_l)\big)$$

iff

$$\prod_{(a_1,\dots,a_l)\in[n]^l}^{<_{Lex}} \gamma(\varphi_1^w[a_1, \dots, a_l] \circ \cdots \circ \varphi_k^w[a_1, \dots, a_l]) \in B$$

NC1 via Logic

- Monoid Multiplication Quantifier, $\Gamma_{\gamma}^{M,B}$:

For example, if $l = 1$ and $k = 1$

$$w \vDash \Gamma_{\gamma}^{M,B} x\ \varphi_1(x)$$

iff

$$\gamma(\varphi_1^w[1]) \cdot \ldots \cdot \gamma(\varphi_1^w[n]) \in B$$

NC1 via Logic

- Monoid Multiplication Quantifier, $\Gamma_{\gamma}^{M,B}$:

For example, if $l = 1$ and $k = 1$

$$w \vDash \Gamma_{\gamma}^{M,B} x \; \varphi_1(x)$$

iff

$$\gamma(\varphi_1^w[1]) \cdot \ldots \cdot \gamma(\varphi_1^w[n]) \in B$$

Say $(U_1, \cdot)$ where $U_1 = \{0,1\}$ and

$0 \cdot 0 = 0$ $\quad$ $0 \cdot 1 = 0$
$1 \cdot 0 = 0$ $\quad$ $1 \cdot 1 = 1$

NC1 via Logic

- Monoid Multiplication Quantifier, $\Gamma_{\gamma}^{M,B}$:

For example, if $l = 1$ and $k = 1$

$$w \vDash \Gamma_{\gamma}^{M,B} x\ \varphi_1(x)$$

iff

$$\gamma(\varphi_1^w[1]) \cdot \ldots \cdot \gamma(\varphi_1^w[n]) \in B$$

Say $(U_1, \cdot)$ where $U_1 = \{0,1\}$ and

$0 \cdot 0 = 0$ $\quad$ $0 \cdot 1 = 0$
$1 \cdot 0 = 0$ $\quad$ $1 \cdot 1 = 1$

$\gamma : \{0,1\} \to U_1$
s.t. $\gamma(0) = 1$ and $\gamma(1) = 0$

NC1 via Logic

- U_1 Multiplication Quantifier, $\Gamma_{\gamma}^{U_1,\{0\}}$:

  For example, if $l = 1$ and $k = 1$

$$w \vDash \Gamma_{\gamma}^{U_1,\{0\}} x \, \varphi_1(x)$$

iff

$$\gamma(\varphi_1^w[1]) \cdot \ldots \cdot \gamma(\varphi_1^w[n]) \in \{0\}$$

Say $(U_1, \cdot)$ where $U_1 = \{0,1\}$ and

$0 \cdot 0 = 0$ $\quad$ $0 \cdot 1 = 0$
$1 \cdot 0 = 0$ $\quad$ $1 \cdot 1 = 1$

$\gamma : \{0,1\} \to U_1$
s.t. $\gamma(0) = 1$ and $\gamma(1) = 0$

NC1 via Logic

- U_1 Multiplication Quantifier, $\Gamma_{\gamma}^{U_1,\{0\}}$:

  For example, if $l = 1$ and $k = 1$

$$w \vDash \Gamma_{\gamma}^{U_1,\{0\}} x\, \varphi_1(x)$$

iff

$$\gamma(\varphi_1^w[1]) \cdot \ldots \cdot \gamma(\varphi_1^w[n]) \in \{0\}$$

Same as "∃"!

Say $(U_1, \cdot)$ where $U_1 = \{0,1\}$ and

$0 \cdot 0 = 0 \quad 0 \cdot 1 = 0$
$1 \cdot 0 = 0 \quad 1 \cdot 1 = 1$

$\gamma : \{0,1\} \to U_1$
s.t. $\gamma(0) = 1$ and $\gamma(1) = 0$

NC1 via Logic

- NC1 is equal to the languages recognized by FO$(+,\times)$ with multiplication quantifiers for finite monoids
  - Or simply multiplication quantifiers for a finite non-solvable group, e.g., S_5
  - Result of Barrington, Immerman, and Straubing (1990)

NC1 via Logic

- NC1 is equal to the languages recognized by FO$(+,\times)$ with multiplication quantifiers for finite monoids
  - Or simply multiplication quantifiers for a finite non-solvable group, e.g., S_5
  - Result of Barrington, Immerman, and Straubing (1990)

- Their proof requires that we have multiplication quantifiers binding multiple variables
- Can this be done with only unary quantifiers? (i.e., $l = 1$)
  - First asked in Lautemann et al. (2001)

NC1 via Logic

- NC1 is equal to the languages recognized by FO$(+,\times)$ with multiplication quantifiers for finite monoids
  - Or simply multiplication quantifiers for a finite non-solvable group, e.g., S_5
  - Result of Barrington, Immerman, and Straubing (1990)

- Their proof requires that we have multiplication quantifiers binding multiple variables
- Can this be done with only unary quantifiers? (i.e., $l = 1$) Yes!
  - First asked in Lautemann et al. (2001)

Recognition via Typed Monoids

- A *typed monoid* is a tuple (M, G, E) where
  - M is a monoid
  - $G \subseteq \wp(M)$ is finite and closed under union, intersection, and complementation
  - $E \subseteq M$ and is finite

Recognition via Typed Monoids

- A *typed monoid* is a tuple (M, G, E) where
  - M is a monoid
  - $G \subseteq \wp(M)$ is finite and closed under union, intersection, and complementation
  - $E \subseteq M$ and is finite
- Say we have a typed monoid $T = (M, G, E)$ and a language $L \subseteq \Sigma^*$.

Recognition via Typed Monoids

- A *typed monoid* is a tuple (M, G, E) where
  - M is a monoid
  - $G \subseteq \wp(M)$ is finite and closed under union, intersection, and complementation
  - $E \subseteq M$ and is finite
- Say we have a typed monoid $T = (M, G, E)$ and a language $L \subseteq \Sigma^*$.
- We say that T *recognizes* L if there exists a homomorphism $h: \Sigma^* \to M$, where $h(\Sigma) \subseteq E$, and an element $A \in G$ such that $L = h^{-1}(A)$,
  - (N.B., $h^{-1}(A) = \{w \in \Sigma^* \mid h(w) \in A\}$)

Recognition via Typed Monoids

- Krebs et al. (2007) gave a characterization of TC0 in terms of typed monoids
- Essentially:
    - Given the quantifiers of a “nice” logic characterizing TC0
    - Construct a class of typed monoids by taking a base set of typed monoids relating to these quantifiers and closing it under certain operations

Recognition via Typed Monoids

- Krebs et al. (2007) gave a characterization of TC0 in terms of typed monoids
- Essentially:
  - Given the quantifiers of a “nice” logic characterizing TC0
  - Construct a class of typed monoids by taking a base set of typed monoids relating to these quantifiers and closing it under certain operations
- The catch: the “nice” logic has to contain only unary first-order quantifiers*
- Our logical characterization of NC1 contains non-unary quantifiers

Simplifying NC1's Logic

- Consider $l = 2$ and $w = w_1 \ldots w_n$

$$w \vDash \Gamma_{\gamma}^{M,B} xy\big(\varphi_1(x, y), \ldots, \varphi_k(x, y)\big)$$

Simplifying NC1's Logic

- Consider $l = 2$ and $w = w_1 \dots w_n$

$$w \vDash \Gamma_{\gamma}^{M,B} xy\big(\varphi_1(x,y), \dots, \varphi_k(x,y)\big)$$

- Let $m_{i,j} = \gamma(\varphi_1^w[i,j] \circ \cdots \circ \varphi_k^w[i,j])$

Simplifying NC1’s Logic

- Consider $l = 2$ and $w = w_1 \ldots w_n$

$$w \vDash \Gamma_{\gamma}^{M,B} xy\big(\varphi_1(x,y), \ldots, \varphi_k(x,y)\big)$$

- Let $m_{i,j} = \gamma(\varphi_1^w[i,j] \circ \cdots \circ \varphi_k^w[i,j])$

$$m_{1,1} \ldots m_{1,n} m_{2,1} \ldots m_{2,n} \ldots m_{n,1} \ldots m_{n,n} \in B$$

Simplifying NC1's Logic

- Consider $l = 2$ and $w = w_1 \dots w_n$

$$w \vDash \Gamma_{\gamma}^{M,B} xy\big(\varphi_1(x,y), \dots, \varphi_k(x,y)\big)$$

- Let $m_{i,j} = \gamma(\varphi_1^w[i,j] \circ \cdots \circ \varphi_k^w[i,j])$

$$\underbrace{m_{1,1} \dots m_{1,n}}_{b_1} \underbrace{m_{2,1} \dots m_{2,n}}_{b_2} \dots \underbrace{m_{n,1} \dots m_{n,n}}_{b_n} \in B$$

Simplifying NC1's Logic

$$m = b_1 \ldots b_n$$

Simplifying NC1's Logic

$$m = b_1 \ldots b_n$$

Say $M = \{m_1, \ldots, m_c\}$

Simplifying NC1's Logic

$$m = b_1 \ldots b_n$$

Say $M = \{m_1, \ldots, m_c\}$

$$\Phi = \Gamma_\sigma^{M,B} x\big(\psi_1(x), \ldots, \psi_c(x)\big)$$

Simplifying NC1's Logic

$$m = b_1 \ldots b_n$$

Say $M = \{m_1, \ldots, m_c\}$

$$\Phi = \Gamma_{\sigma}^{M,B} x\big(\psi_1(x), \ldots, \psi_c(x)\big)$$

$$\psi_i(x) = \Gamma_{\sigma}^{M,\{m_i\}} y(\lambda_1(x,y), \ldots, \lambda_c(x,y))$$

Simplifying NC1's Logic

$$m = b_1 \dots b_n$$

Say $M = \{m_1, \dots, m_c\}$

$$\Phi = \Gamma_\sigma^{M,B} x\big(\psi_1(x), \dots, \psi_c(x)\big)$$

$$\psi_i(x) = \Gamma_\sigma^{M,\{m_i\}} y(\lambda_1(x,y), \dots, \lambda_c(x,y))$$

All together,

$$w \vDash \Phi \text{ iff } w \vDash \Gamma_\gamma^{M,B} xy\big(\varphi_1(x,y), \dots, \varphi_k(x,y)\big)$$

Simplifying NC1's Logic

- Therefore… finite non-unary multiplication quantifiers can be defined ("axiomatized") using simply unary ones

Simplifying NC1's Logic

- Therefore… finite non-unary multiplication quantifiers can be defined ("axiomatized") using simply unary ones
- Permitting NC1 to be characterized by the languages expressible in FO($+$,$\times$) with unary multiplication quantifiers for S_5. Call these unary Γ^{S_5} quantifiers.

Simplifying NC1's Logic

- Therefore… finite non-unary multiplication quantifiers can be defined ("axiomatized") using simply unary ones

- Permitting NC1 to be characterized by the languages expressible in FO(+,×) with unary multiplication quantifiers for S_5. Call these unary Γ^{S_5} quantifiers.

- To apply the translation theorem of Krebs et al., we have one more small step:
  - Introduce a unary quantifier Sq where $w \vDash Sq\ x\ \varphi(x)$ iff $|\{a \in [|w|] \mid w, x \mapsto a \vDash \varphi(x)\}| = q^2$ for some $q \in \mathbb{N}$
  - Introduce a unary majority quantifier Maj
  - Replace $+, \times$ with just $<$
  - These steps together do not change the expressive power

A Small Detour: One Step Further

- We can, moreover, improve this to quantifiers which aren’t lexicographic!

A Small Detour: One Step Further

- We can, moreover, improve this to quantifiers which aren't lexicographic!
- Using the work of Bojanczyk et al. (2019, "String-to-string interpretations…"), every FO[<]-definable linear order has a lexicographic nature to it

A Small Detour: One Step Further

- We can, moreover, improve this to quantifiers which aren’t lexicographic!
- Using the work of Bojanczyk et al. (2019, “String-to-string interpretations…”), every FO[<]-definable linear order has a lexicographic nature to it
- Once extracted, we can repeat the earlier techniques to decompose any finite multiplication quantifier using any FO[<]-definable linear order into a sentence using only unary finite multiplication quantifiers

NC1 via Algebra

- Step 1: We have our logical characterization: FO($<$) with Sq, Maj, and unary Γ^{S_5} quantifiers.
    - N.B., only unary quantifiers, and $<$ is the only numerical predicate.

NC1 via Algebra

- Step 1: We have our logical characterization: FO(<) with Sq, Maj, and unary Γ^{S_5} quantifiers.
  - N.B., only unary quantifiers, and $<$ is the only numerical predicate.
- Step 2: Find a typed monoid capturing the semantics of each quantifier

∀ and ∃	$(U_1, \wp(U_1), U_1)$
Maj	$(\mathbb{Z}, \{\emptyset, \mathbb{Z}^+, \mathbb{Z} - \mathbb{Z}^+, \mathbb{Z}\}, \pm 1)$
Sq	$(\mathbb{N}, \{\emptyset, \mathbb{S}, \mathbb{N} - \mathbb{S}, \mathbb{N}\}, \{0,1\})$
All unary Γ^{S_5}	$(S_5, \wp(S_5), S_5)$

NC1 via Algebra

- Step 3: Closing

$$\{(U_1, \wp(U_1), U_1), (\mathbb{Z}, \{\emptyset, \mathbb{Z}^+, \mathbb{Z} - \mathbb{Z}^+, \mathbb{Z}\}, \pm 1), (\mathbb{N}, \{\emptyset, \mathbb{S}, \mathbb{N} - \mathbb{S}, \mathbb{N}\}, \{0,1\}), (S_5, \wp(S_5), S_5)\}$$

under the “ordered strong block product”. Call this class of typed monoids N.

NC1 via Algebra

- Step 3: Closing

$$\{(U_1, \wp(U_1), U_1), (\mathbb{Z}, \{\emptyset, \mathbb{Z}^+, \mathbb{Z} - \mathbb{Z}^+, \mathbb{Z}\}, \pm 1), (\mathbb{N}, \{\emptyset, \mathbb{S}, \mathbb{N} - \mathbb{S}, \mathbb{N}\}, \{0,1\}), (S_5, \wp(S_5), S_5)\}$$

under the “ordered strong block product”. Call this class of typed monoids N.

Finally, we get **a language L is in NC1 iff L is recognized by a typed monoid in N**.

Conclusion

In progress:

- Constructing algebraic characterizations of classes beyond NC1

Conclusion

In progress:

- Constructing algebraic characterizations of classes beyond NC1
- Exploring implications of quantifier definability with second-order quantifiers

Conclusion

In progress:

- Constructing algebraic characterizations of classes beyond NC1
- Exploring implications of quantifier definability with second-order quantifiers

Outline:

1. How We Recognize Languages
2. Recognition with Logic
3. NC1 via Logic
4. Recognition with Typed Monoids
5. Simplifying NC1’s Logic
6. NC1 via Algebra
7. Conclusion

Email: ate26@cam.ac.uk

Thank You!