Promise Algebra: An Algebraic Model of Non-Deterministic Computations

Eugenia Ternovska

Simon Fraser University, Canada

July 19, 2024

Non-deterministic Algorithms

Robert W. Floyd in 1967:

"Nondeterministic algorithms resemble conventional algorithms
... except that:
(1) One may use a multiple-valued function $Choice(X)$
(2) All points of termination are labelled as successes or failures."

"All the time life is a fork. If you are straight up with yourself you don't have to decide which road to take. Your karma will look after that."

– George Harrison

Are non-deterministic algorithms strictly more efficient than deterministic ones?

In 1975, Ladner proved that if P$\neq$NP then there are infinitely many complexity classes between them

All examples of such intermediate problems are very artificial

A “clean” class of problems within NP was suggested in a seminal work of Feder and Vardi **[Feder,Vardi'93,98]**

Their goal: find a large subclass of NP which exhibits a dichotomy

They studied Uniform Constraint Satisfaction Problem (CSP)

CSP is identified with the **Homomorphism Problem:**

Given: two relational structures $\mathfrak{A}$ and $\mathfrak{B}$ Question: is there a homomorphism $\mathbf{h} : \mathfrak{A} \to \mathfrak{B}$?

$\mathfrak{B}$ is called a template

Non-Uniform CSP: the template $\mathfrak{B}$ is fixed

[Feder,Vardi'93] conjectured a dichotomy:

Non-Uniform CSP is either in P-time or NP-complete

[Bulatov:2017, Zhuk:2017] closed the conjecture positively

The CSP development relied on the techniques of Universal Algebra

Setting: Computational Decision Problems

3-Colourability, s-t-Reachability, Size Four, EVEN, ...

Queries $\mathfrak{A} \models \varphi$ ask:

- Is graph $\mathfrak{A}$ 3-colourable?
- Is the size of the domain of $\mathfrak{A}$ EVEN?

Data complexity [Vardi:1982]:

query φ is fixed and structures $\mathfrak{A}$ vary

Goals

Develop an **algebraic** language for reasoning about non-deterministic computations **in a "deterministic" way**

in particular, for reasoning about

the **set of certificates** of a computational decision problem,

as a mathematical object

We will see a mechanism for constructing such an algebra

In particular,

- how to tame the non-determinism of classical connectives
- how to view the algebra as a logic, a query language
- how to quantify over certificates, algebraically
- how to reason about the existence of a certificate
- how to capture complexity classes with an algebra

- Start from FO(LFP), the logic used in [Immerman-Vardi] theorem
- Inspired by two-variable fragments [Vardi:1995],
  partition variables of atomic symbols into inputs and outputs
- Produce **algebra of (functional) binary relations**
  on finite strings of structures over the same relational vocabulary

$$t ::= \quad \mathrm{id} \mid \underbrace{q(\bar{X}, Y)}_{\text{unary CQ}} \mid \quad \overbrace{\sim t \mid t\,;t \mid t \sqcup t \mid t^{\uparrow}}^{\text{analogous to } \neg, \wedge, \vee, *} \quad \mid \textsf{BG}(P \neq Q) \mid (P = Q)$$

$$\tau \ := \ \tau_{\mathrm{EDB}} \uplus \tau_{\mathrm{reg}}$$

atomic binary relations (CQs) specify a transition system $\mathbf{Tr}[\![\cdot]\!]$

States of $\mathbf{Tr}[\![\cdot]\!]$ are relational τ-structures

Operations, Intuitively

$$t ::= \varepsilon a \mid \overbrace{\mathrm{id} \mid {\sim}t \mid t \,;\, t \mid t \sqcup t \mid t^{\uparrow} \mid \textsf{BG}(P \neq Q) \mid (P = Q)}^{\text{function-preserving}}$$

Unary Negation (Anti-Domain): $\sim t$ – there is no outgoing t-transition

Composition: $t \,;\, g$ – function composition (execute sequentially)

Preferential Union: $t \sqcup g$ – perform t if it's defined, o.w. perform g

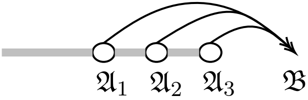
Maximum Iterate: $t^{\uparrow}$ – output the longest transition of t^*

Back Globally: $\textsf{BG}(P_{now} \neq Q)$ – compare the "content" of "register" P now with "registers" Q before, must be different

Equality Check : $(P = Q)$ – compare the "content" of "registers" P and Q

$\tau \ := \ \tau_{\mathrm{EDB}} \uplus \tau_{\mathrm{reg}}$

Maximum Iterate vs the Kleene Star



Maximum Iterate:

$t_0^{\uparrow} := \sim t, \quad t_{n+1}^{\uparrow} := t_n^{\uparrow} \,;\, t,$
$t^{\uparrow} := \bigcup_{n \in \mathbb{N}} t_n^{\uparrow}.$

deterministic operator

(a partial function).

The Kleene Star:

$t_0^* := \mathrm{id}, \quad t_{n+1}^* := t \,;\, t_n^*,$
$t^* := \bigcup_{n \in \mathbb{N}} t_n^*.$

non-deterministic operator

(not a function).

Choice Functions

A unary CQ returns a **set**

A history-dependent **Choice** function **picks one element**

E.g.:

$$Reach'(y) \quad :- \quad \underbrace{Reach(x), \mathbf{E}(x,y)}_{\text{CQ}}$$

use free Choice function variable ε (at most one per expression)

$$t ::= \mathrm{id} \mid \underbrace{q[\varepsilon](\underline{\bar{X}}, Y)}_{\text{CQ with Choice}} \mid {\sim}t \mid t\,;t \mid t \sqcup t \mid t^{\uparrow} \mid \text{BG}(P \neq Q) \mid (P = Q)$$

Notation:

$$\varepsilon\,\{\; \textit{Reach}'(y) \leftarrow \textit{Reach}(x), E(x,y) \;\}$$

Choice Functions Give Semantic to Atomic Transductions

$\mathbf{U}$ is the set of all τ-structures over the same finite domain
$(\tau \;:=\; \tau_{\text{EDB}} \uplus \tau_{\text{reg}})$

$\mathcal{M}$ is the set of atomic action symbols (macros) that refer to CQs

$$h : \mathcal{M} \to \underbrace{(\mathbf{U}^+ \rightharpoonup \mathbf{U}^+)}_{\text{partial function}}$$

h : returns **functional binary relation**

pick one possible transition from $\textbf{Tr}[\![a]\!] \subseteq \mathbf{U} \times \mathbf{U}$

E.g. $(v\mathfrak{A}, v\mathfrak{A}\mathfrak{B}) \in h(\varepsilon a)$ if $(\mathfrak{A}, \mathfrak{B}) \in \textbf{Tr}[\![a]\!]$ was selected

Extend h to All Terms

$$\bar{h} : Terms \to (\mathbf{U}^+ \rightharpoonup \mathbf{U}^+)$$

1. $\bar{h}(\varepsilon a) := h(\varepsilon a)$
2. $\bar{h}(\mathrm{id}) := \{(v, v)\}$
3. $\bar{h}({\sim}t) := \{(v, v) \mid \neg\big(\ \exists h' \exists w\ ((v, w) \in \bar{h}'(t)\big)\}$
4. $\bar{h}(t\,;g) := \{(v, w) \mid \exists\ u\ \big((v, u) \in \bar{h}(t)\ \wedge\ (u, w) \in \bar{h}(g)\big)\}$.
5. $\bar{h}(t \sqcup g) := \begin{cases} \bar{h}(t) & \text{if } \bar{h}(t) \neq \varnothing, \\ \bar{h}(g) & \text{if } \bar{h}(t) = \varnothing. \end{cases}$
6. $\bar{h}(t^{\uparrow}) := \{(v, w) \mid (v, v) \in \bar{h}({\sim}t) \wedge v = w$
   $\vee\ \exists u\ \big(v \sqsubseteq u \sqsubseteq w\ \wedge\ (v, u) \in \bar{h}(t)\ \wedge\ (u, w) \in \bar{h}(t^{\uparrow})\big)\}$.
7. $\bar{h}(P = Q) := \{(v, v) \mid Q^{v(last)} = P^{v(last)}\}$.
8. $\bar{h}(\ \textsf{BG}(P \neq Q)) := \{(v, v) \mid\ \neg\exists w\ \big(w \sqsubseteq v\ \wedge\ P^{v(last)} = Q^{w(last)}\big)\}$.

Origins of the Constructs

Epsilon Operator   [Hilbert, Bernays: 1939]

Soviet logicians in the 70's and 80's, and the study is still ongoing

[Arvind and Biswas'87], [Gire and Hoang'98], [Blass and Gurevich'00], [Otto'00], [Richerby and Dawar'03]

Unary Negation (Anti-Domain): $\sim t$

[Groenendijk and Stokhof: 1991]

[Hollenberg, Visser: 1999]

...

Maximum Iterate: $t^{\uparrow}$, Preferential Union: $t \sqcup g$

[Jackson, Stokes: 2011]

[McLean: 2017], ...

The Algebra is Equivalent to a Linear-Time Dynamic Logic

via a standard embedding:

$$t ::= \varepsilon a \mid \mathrm{id} \mid {\sim}t \mid t \,;\, t \mid t \sqcup t \mid t^{\uparrow} \mid \textsf{BG}(P \neq Q) \mid (P = Q) \mid \varphi?$$
$$\varphi ::= \top \mid \neg\varphi \mid \varphi \wedge \varphi \mid |t\rangle\varphi$$

$$\varphi? := {\sim}{\sim}\,\varphi \qquad = \mathrm{Dom}(\varphi) \qquad\qquad \text{(test action)}$$
$$|\alpha\rangle\ \varphi := \mathrm{Dom}(\alpha\,;\varphi)$$

Satisfaction relation: $v \models \varphi(h/\varepsilon)$ iff $(v, v) \in \bar{h}(\varphi)$

Programming constructs are **definable**

$$\mathbf{if}\ \varphi\ \mathbf{then}\ \alpha\ \mathbf{else}\ \beta \ := \ (\varphi?\,;\alpha) \sqcup \beta$$
$$\mathbf{while}\ \varphi\ \mathbf{do}\ \alpha := (\varphi?\,;\alpha)^{\uparrow}\,;({\sim}\varphi?)$$
$$\mathbf{repeat}\ \alpha\ \mathbf{until}\ \varphi := \alpha\,;(({\sim}\varphi?)\,;\alpha)^{\uparrow}\,;\varphi?$$

Implicit Quantification over ε

Recall: $\bar{h}(\sim t) := \{(v,v) \mid \neg\big(\exists h' \exists w \; ((v,w) \in \bar{h}'(t)\big)\}$

$\sim\sim t$ (domain) $-$ implicitly, $\exists\varepsilon$

there is a Choice function witnessing a successful execution of t

$\sim t$ (anti-domain) $-$ implicitly, $\forall\varepsilon$

there is no Choice function witnessing a successful execution of t

These "quantifiers" can alternate

This allows us to formalize problems at all levels of the PTH

Main Computational Task

Problem: **Main Task (Decision Version)**
Given: Relational τ- structure $\mathfrak{A}$ and term t
Question:

$$\exists h \ \mathfrak{A} \models |t\rangle\top(h/\varepsilon) \ ? \tag{1}$$

e.g., $\mathfrak{A}$ is a graph, t describes 3-Colourability, and h is a witness

A computational problem specified by t is an isomorphism-closed class $\mathcal{P}_t$ of structures $\mathfrak{A}$ such that (1) holds

(i.e., there is a successful execution of t on input $\mathfrak{A}$)

One-player Game: Arena: transition system $\mathbf{Tr}[\![\cdot]\!]$,

Given t and $\mathfrak{A}$, is there a winning strategy h from $\mathfrak{A}$?

Problem: **Size Four** α_4 (Counting)

Given: A structure $\mathfrak{A}$ with an empty vocabulary.
Question: Is $|dom(\mathfrak{A})|$ equal to 4?

$$\alpha_4 := (GuessNewP)^4\,;\sim GuessNewP$$

$$\begin{array}{lcl} \varepsilon GuessP & := & \varepsilon\left\{\ P(x) \leftarrow \quad \right\} \\ \varepsilon CopyPQ & := & \varepsilon\left\{\ Q(x) \leftarrow P(x)\ \right\} \end{array}$$

$$GuessNewP := \big(GuessP\,;\ \textsf{BG}(P \neq Q)\big)\ ;\ CopyPQ$$

$\mathfrak{A} \models_T |\alpha_4\rangle\top$ (i.e., there is an h) iff the input domain is of size 4

Problem: **s-t Connectivity** $\alpha(E, S, T)$ (Reachability)

Given: Binary edge relation E, two constants s and t represented as singleton-set relations S and T.
Question: Is t reachable from s by following the edges?

$$\alpha_{\mathsf{ST}}(E,S,T) := \quad M_{base\_case}\,;\mathbf{repeat}\ \Big(M_{ind\_case};\ {}_{\mathsf{BG}}(\mathit{Reach}' \neq \mathit{Reach})\Big)\,;\mathit{Copy}\ \ \mathbf{until}\ \mathit{Reach} = T.$$

$$\varepsilon M_{base\_case} \quad := \quad \varepsilon\,\{\ \mathit{Reach}(x) \longleftarrow S(x)\ \}\,,$$
$$\varepsilon M_{ind\_case} := \varepsilon\,\{\ \mathit{Reach}'(y) \longleftarrow \mathit{Reach}(x), E(x,y)\ \}\,,$$
$$\varepsilon \mathit{Copy} := \quad \varepsilon\,\{\ \mathit{Reach}(x) \longleftarrow \mathit{Reach}'(x)\ \}\,.$$

the answer to $\mathfrak{A} \models |\alpha_{\mathsf{ST}}\rangle\top$ is true

iff t is reachable from s by following the edges of the input graph

Complexity of Query Evaluation

Restricted fragment: $\sim$ applies to atomic expressions or equalities only. All Choice functions are of polynomial length $length(h) \in O(n^k)$ where $n = |\mathfrak{A}|$

Theorem: *The data complexity of checking $\mathfrak{A} \models |\alpha\rangle\top$, for α in the restricted fragment, is in NP*

Proof: Guess h. Check atomic actions (CQs) and the fixed term in poly-time using rules of Structural Operational Semantics

Thus, we return "yes" in poly-time if the witness h proves that the answer to $\mathfrak{A} \models |\alpha\rangle\top$ is "yes"; or "no" in polynomial time otherwise. $\square$

Theorem: *For every NP-recognizable class $\mathcal{K}$ of structures, there is a sentence in the restricted fragment, whose models are exactly $\mathcal{K}$*

Proof: Design term α_{TM}, focus on query

$$\mathfrak{A} \models |\alpha_{\mathrm{TM}}\rangle \top$$

Start by guessing an order:

$$\alpha_{\mathrm{TM}}(\mathfrak{A}) := \mathrm{ORDER}\ ;\ \mathrm{START}\ ;\ \textbf{repeat}\ \mathrm{STEP}\ \textbf{until}\ \mathrm{END}.$$

$\square$

Note: the structures in class $\mathcal{K}$ are not ordered

Corollary: The restricted fragment of the logic captures NP

Summary

- algebra/logic on strings of relational structures
- operations are function-preserving
- can specify reachability, cardinality and "mixed propagations" examples
  e.g., EVEN is not in Datalog, not in MSO but is in our logic
- a fragment of the logic captures exactly NP
- in general, problems at any level af the PTH can be specified (if Choice functions are of polynomial length)
- We believe it's the first algebraic approach to capturing complexity classes

Open Problems & Current Research

1. Under what conditions on the algebraic terms, a **naive** winning strategy h for $\mathfrak{A} \models |t\rangle\top$ exists?

Open Problems & Current Research

1. Under what conditions on the algebraic terms, a **naive** winning strategy h for $\mathfrak{A} \models |t\rangle\top$ exists?
   (at each step, make any possible choice, and you will succeed)

Open Problems & Current Research

1. Under what conditions on the algebraic terms, a **naive** winning strategy h for $\mathfrak{A} \models |t\rangle\top$ exists?
   (at each step, make any possible choice, and you will succeed)

   "All the time life is a fork. If you are straight up with yourself you don't have to decide which road to take. Your karma will look after that."

   – George Harrison

Open Problems & Current Research

1. Under what conditions on the algebraic terms, a **naive** winning strategy h for $\mathfrak{A} \models |t\rangle\top$ exists?
   (at each step, make any possible choice, and you will succeed)

   "All the time life is a fork. If you are straight up with yourself you don't have to decide which road to take. Your karma will look after that."

   – George Harrison

2. Connections to other logics/algebras, & automata
3. Proof system, formal proofs vs Choice functions as certificates
4. Does Interpolation theorem hold? (e.g., for a fragment)

Thank you!