μ-calculus over data words

Thomas Colcombet, Amaldev Manuel

LIAFA, Université Paris-Diderot

Data words and languages

A data word $w = (a_1, d_1) \ldots (a_n, d_n)$, $a_i \in \Sigma$, $d_i \in \mathcal{D}$ where,

- Σ is a finite alphabet.
- $\mathcal{D}$ is an infinite domain, eg. $\mathbb{N}$

Abstractions

- Distributed systems : Σ as program actions, $\mathcal{D}$ as process identifiers
- Security protocols : Σ as messages, $\mathcal{D}$ as nonces
- XML documents : Σ as attribute names, $\mathcal{D}$ as values

A **data language** is a set of data words. Eg. Set of all data words where all data values are distinct.

We want **automata and logics** for these structures.

Automata and logics for data words

There is a whole menagerie of them!

Automata (Not exhaustive)
General set-up — Finite state automata + Memory structures

- Registers (Kamisky-Francez, Demri-Lazic)
- Pushdown (Kamisky et. al)
- Counters (M. Ramanujam, Demri et. al)
- Hash table (Schwentick-Bjorklund)
- Pebble (Neven-Schwentick-Vianu, Tan, . . .)
- . . .

None closed under Boolean operations.

Logics (Not exhaustive)
General set-up — Logic for words + logical/non-logical concoctions

- LTL + Registers (Demri-Lazic, . . .)
- FO^2 (Bojanczyk-Muscholl-Schwentick-Segoufin-David, . . .)
- Mu-calculus + Registers (Jurdzinski et. al.)
- XPath (Figuera, Segoufin et. al.)
- Guarded-MSO (Colcombet-Ley-Puppis)
- . . .

A thematic description

To recognize a data language one needs **unbounded** memory – Consider the set of all data words where all data values are distinct.

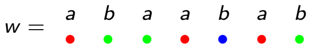
Two options

- Develop suitable notions of finiteness – Eg. graph measures like tree-width, Data monoids, Sets with atoms . . .
- Consider data words in their full generality i.e. as graphs eg. Walking automata, this work.

We consider data words as graphs.

Data words as graphs

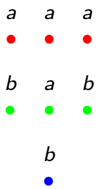
A data language $L \subseteq (\Sigma \times \mathcal{D})^*$ is invariant under permutations of $\mathcal{D}$.



String projection

$$str(w) = a \quad b \quad a \quad a \quad b \quad a \quad b$$

Classes



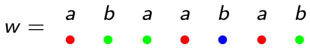
class projections

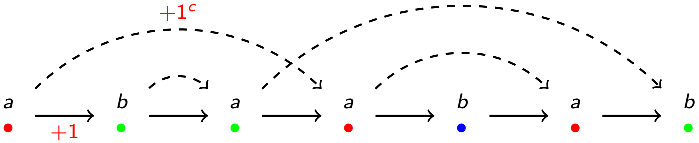
$$a \quad a \quad a$$

$$b \quad a \quad b$$

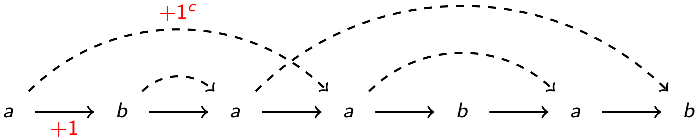
$$b$$

Data word as a graph





The Graph $([n], \Sigma, +1, +1^c)$

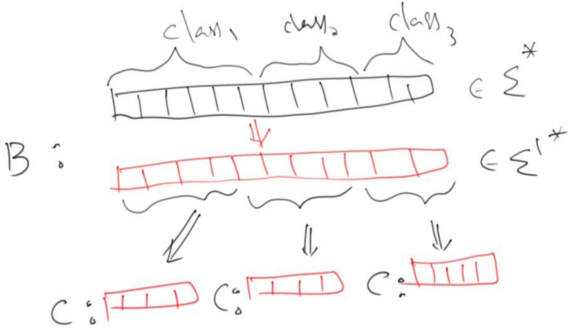


Data automata

A **Data automaton** A is a pair (B, C) where $B : \Sigma \to \Sigma'$ is a letter-to-letter finite state transducer and C is a finite state automaton over Σ'.

Run of a aata automata

- B runs over the string projection of the data word and outputs a word over the alphabet Σ'.
- C run over each of the class projections of the output of B.



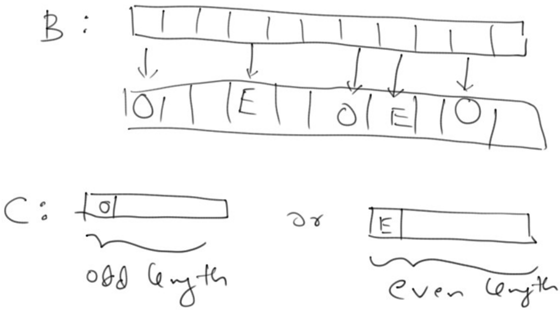
The automaton A accepts if both B and C have successful runs.
Equivalent to EMSO^2 **on data words & Emptiness decidable! – BMSSD05**

Data automata – examples

Example (All classes have even length)

B - copies the input to the output.
C - accepts when the input word has even length.

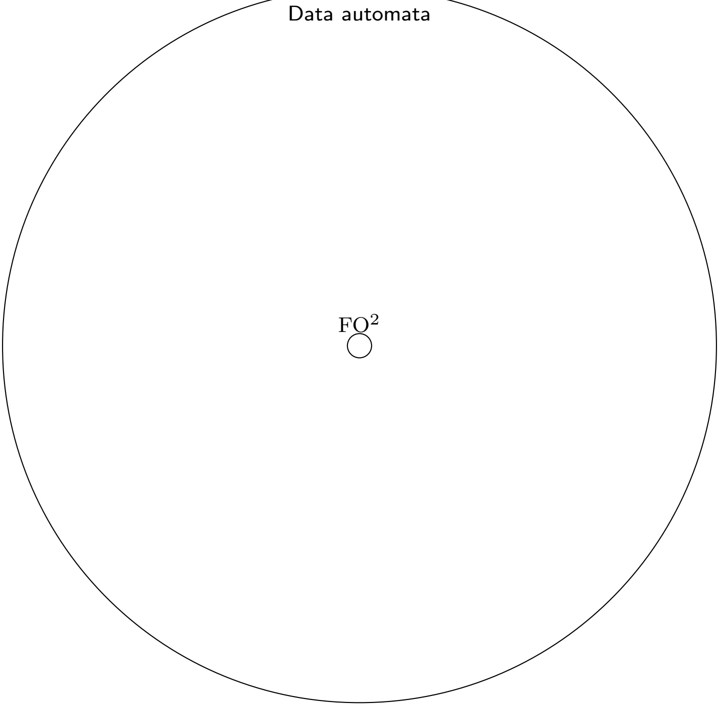
Example (Even number of classes of even length)

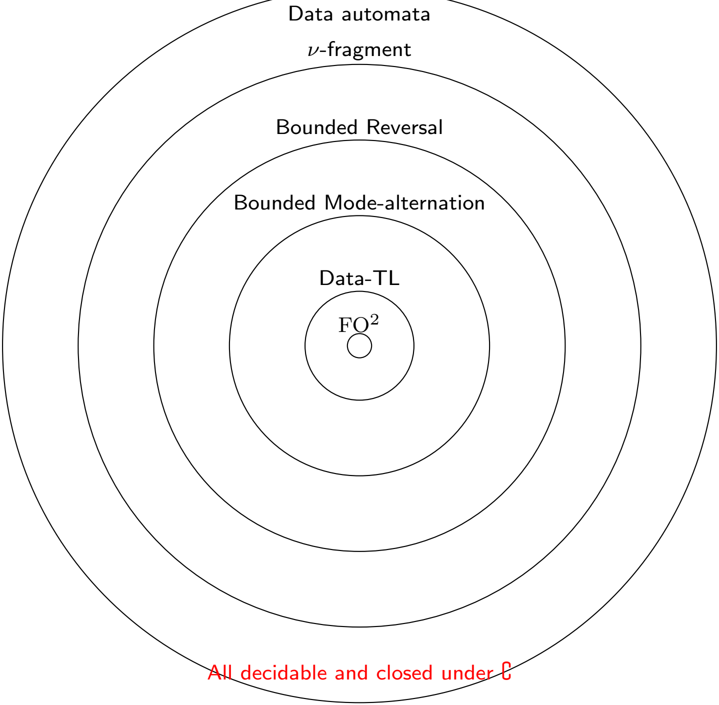


B - guesses the first positions of the class and labels them by E or O and verifies there are even number of E.
C - Verifies that the guess of B is correct and the class is of even length if and only if it is labelled E.

Our contribution

- The class of data languages (both for finite and ω-data words) defined by Data Automata is explored using μ-calculus.
- Various classes (in particular complementable ones) are defined and classified upto equivalence.
- Hierarchy theorems for some classes.


Data automata
FO^2



Data automata
ν-fragment
Bounded Reversal
Bounded Mode-alternation
Data-TL
FO^2
All decidable and closed under ∁

Bounded Mode-alternation

Take an **unambiguous** (at most one successful run) letter-to-letter transducer A.

- Global transduction : When A reads the string projection and outputs.
- Class transduction : When a copy of A runs on each class and outputs.

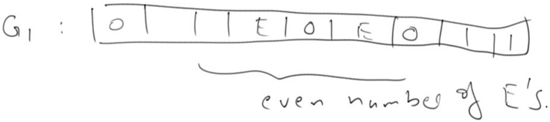
BMA $=$ Cascade of transducers of unambiguous class and global transducers.

BMA examples

Example (All classes have even length)

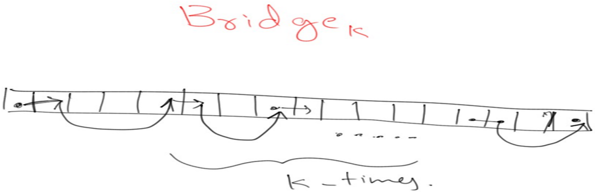
B - copies the input to the output.
C - accepts when the input word has even length.

Example (Even number of classes of even length)



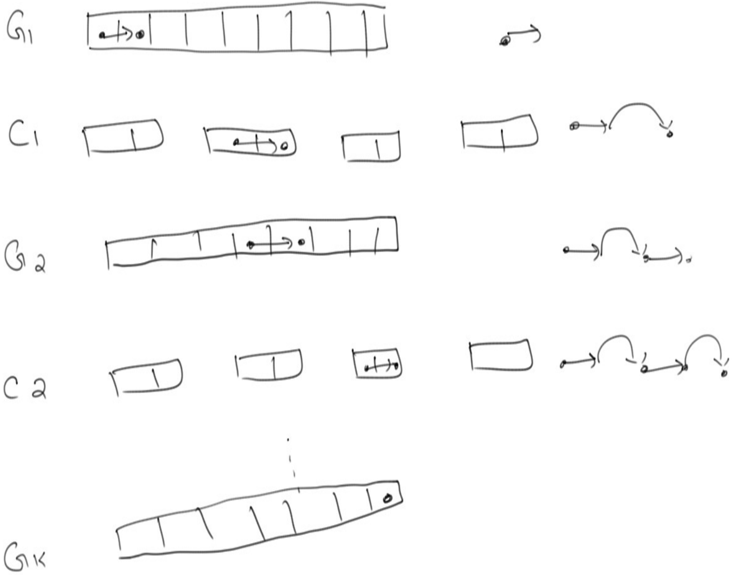
C - Guesses the parity of the class and labels the first position by E or O.
G - Verifies that there are even number of E.

Bridge$_k$



$Bridge_k$

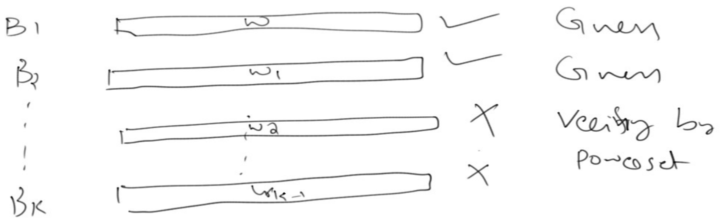
Bridge_k is accepted by a cascade of height $2k$.



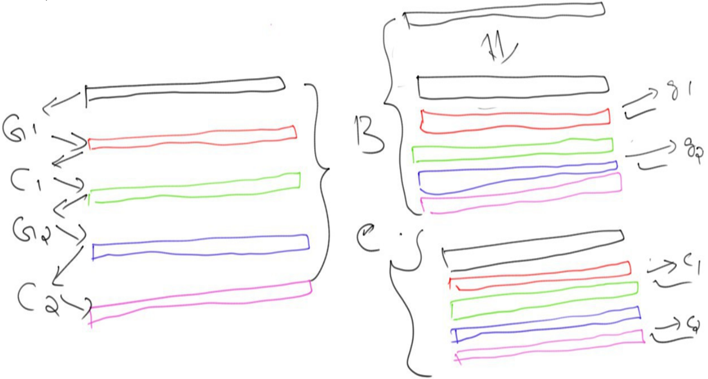
BMA is closed under complement.

$$A = (B_1, B_2, \ldots B_k)$$

Assume $w \notin L(A)$



BMA $\subsetneq$ Data automata.



- B guesses all labellings by all transductions and verifies the global transductions are correct.
- C verifies the class transductions are correct.

Hence BMA $\subsetneq$ Data automata since DA are not closed under complementation.

What is a logic corresponding to BMA ?

μ-calculus on data words

$$w = ([n], \Sigma, +1, +1^c)$$

Algebra of functions from $2^{[n]} \to 2^{[n]}$ with extremal fix-points.

Basic functions

$$\begin{array}{lcl} [\![T]\!]_w & = & \text{"constant function which maps to all positions"} \\ [\![p]\!]_w & = & \text{"All positions in } w \text{ where } p \text{ holds."} \\ [\![x]\!]_w & = & id \end{array}$$

Basic operations

$$\begin{array}{lcl} [\![\varphi_1(x) \vee \varphi_2(x)]\!]_w & = & [\![\varphi_1(x)]\!]_w \cup [\![\varphi_2(x)]\!]_w \\ [\![\varphi_1(x) \wedge \varphi_2(x)]\!]_w & = & [\![\varphi_1(x)]\!]_w \cap [\![\varphi_2(x)]\!]_w \\ [\![\neg\varphi_1(x)]\!]_w & = & [n] \setminus [\![\varphi(x)]\!]_w \\ [\![\mathtt{X}^g \varphi(x)]\!]_w & = & [\![\varphi(x)]\!]_w - 1 \\ [\![\mathtt{Y}^g \varphi(x)]\!]_w & = & [\![\varphi(x)]\!]_w + 1 \\ [\![\mathtt{X}^c \varphi(x)]\!]_w & = & [\![\varphi(x)]\!]_w - 1^c \\ [\![\mathtt{Y}^c \varphi(x)]\!]_w & = & [\![\varphi(x)]\!]_w + 1^c \end{array}$$

Extremal fix-points, when the function is monotone

$$\begin{array}{lcl} [\![\mu x.\varphi(x)]\!]_w & = & LFP([\![\varphi(x)]\!]_w) \\ [\![\nu x.\varphi(x)]\!]_w & = & GFP([\![\varphi(x)]\!]_w) \end{array}$$

Function composition

Let

$$\varphi(x) \equiv f \text{ and } \psi(y) \equiv g$$

Then,

$$\varphi(x/\psi(y)) \equiv f \circ g$$

In general for a class of formulas S, Comp(S) is the set closed under substitutions.

Comp(S) is the set of all finite iterations of functions in S

Bounded Mode-Alternation

$$M^g = \{\mathtt{X}^g, \mathtt{Y}^g\} \quad M^c = \{\mathtt{X}^c, \mathtt{Y}^c\}$$

global functions
Formulas(M^g) :- All formulas which uses only M^g.

class functions
Formulas(M^c) :- All formulas which uses only M^c.

BMA – –"finite composition of class functions and global functions"

$$BMA = \text{Comp}\left(Formulas\left(M^g\right) \cup Formulas\left(M^c\right)\right)$$

Examples

Example (All classes have even length)

$$\begin{aligned} \text{first} &:= \neg \mathtt{Y}^g \text{true} \\ \text{last} &:= \neg \mathtt{X}^g \text{true} \\ \text{cfirst} &:= \neg \mathtt{Y}^c \text{true} \\ \text{clast} &:= \neg \mathtt{X}^c \text{true} \end{aligned}$$

'First position of a class of even length" $\varphi := (\text{cfirst} \wedge \theta x.\, (\mathtt{X}^c \text{clast} \vee \mathtt{X}^c \mathtt{X}^c x))$
'All classes are of even length" $:= G\,(\text{cfirst} \rightarrow \varphi)$

Example

Example (Even number of classes of even length)

"Last position where p holds" $\mathrm{last}(p) := \quad p \wedge \neg F p$
"First position where p holds" $\mathrm{first}(p) := \quad p \wedge \neg P p$

"Even occurrences of p from the end"

$$\psi(p) := \theta x.((p \wedge \mathtt{X}^g(\neg p\,\mathcal{U}\,\mathrm{last}(p))) \vee (p \wedge \mathtt{X}^g(\neg p\,\mathcal{U}\,(p \wedge \mathtt{X}^g(\neg p\; U\, x)))))$$

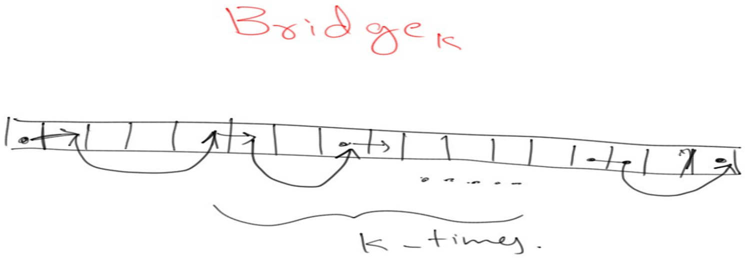
Even number of even classes $:= F(\mathrm{first}(\varphi) \wedge \psi(\varphi))$

Comp of class and global functions is equivalent to Cascades of class and global transducers.
Composition height = Cascade height
Formulas and Cascades form a hierarchy under the height of composition.
Is the hierarchy strict ?

Bridge

$$Bridge_0 \equiv \text{last}$$

$$Bridge_k \equiv \text{first} \wedge \left(Bridge_{k-1} \vee \mathtt{X}^g \mathtt{X}^c Bridge_{k-1}\right)$$



$$Bridge_\infty \equiv \theta x. (\text{first} \wedge (\text{last} \vee \mathtt{X}^g \mathtt{X}^c x))$$

Heirarchy Theorem

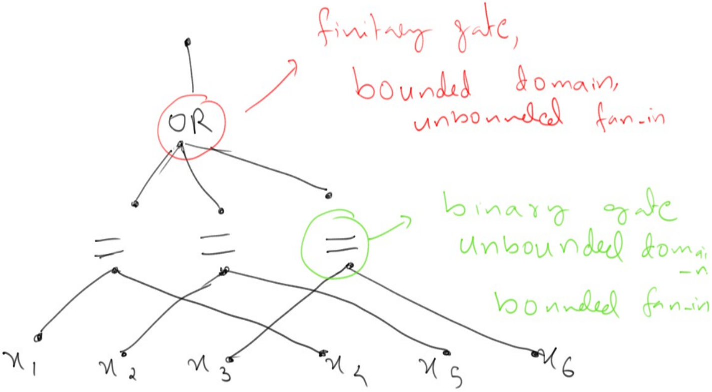
Theorem

$Bridge_{2^{\mathcal{O}(k)}}$ is not accepted by any cascade of height k.

Corrollary

$Bridge_{\infty}$ is not in BMA.

Combinatorial circuits



- circuits taking sequences of integers as input, defining functions of the form $f : \mathbb{N}^* \to \{0, 1\}$,
- made up of gates of the form $g : E^k \to F$ where $E, F \subseteq \mathbb{N}$ such that either,
  - finitary gates E and F are finite, or
  - binary gates $k \leq 2$.

The circuit lower-bound

Theorem

There does not exist a circuit of depth k which takes as input $x_1, \ldots, x_{2^k+1}, N$ and checks if

$$\left(x_1 + x_2 + \ldots + x_{2^k+1}\right) \mod N = 0$$

reduction

- Given $x_1, \ldots, x_{2^k+1}, N$ construct a word w such that $\sum_{i=1}^{2^k+1} x_i = 0 \mod N$ if and only if w has a bridge.
- Given a cascade $\mathcal{A}$ construct a circuit $\mathcal{C}$ of the same height such that $\mathcal{C}$ simulates $\mathcal{A}$ on w.

- Binary gates will correspnd to class transductions.
- Finitary gates will correspnd to global transductions.

Question

Remarks

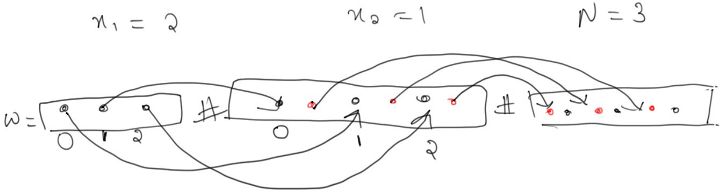
- BMA can be sequentialized, i.e. BMA is the cascade of left-to-right deterministic and right-to-left deterministic class and global transducers.
- If you drop the convervse modalities you have only left-to-right deterministic transducers.

Question

Cascade of word automata is equivalent to block product of finite semigroups. Is there an analogue for BMA ?

Thank you for your attention.

reduction



$$pos_{i+1} = (pos_i + x_i) \bmod N$$

$$w \in Bridge_{2^k} \iff \sum_i x_i \bmod N = 0$$

reduction