Weighted Context-Free Grammars over Bimonoids

George Rahonis and *Faidra Torpari*

Aristotle University of Thessaloniki, Greece

WATA 2018
Leipzig, May 22, 2018

Why bimonoids?

LogicGuard Project I,II

- http://www.risc.jku.at/projects/LogicGuard/
- http://www.risc.jku.at/projects/LogicGuard2/

network security
- specification & verification formalism
- tool for runtime network monitoring

McCarthy-Kleene logic

- four valued logic: t, f, u, e
- truth tables

or	t	f	u	e
t	t	t	t	t
f	t	f	u	e
u	t	u	u	e
e	e	e	e	e

and	t	f	u	e
t	t	f	u	e
f	f	f	f	f
u	u	f	u	e
e	e	e	e	e

- non-commutative
- in practice an "error" is not always a critical error, hence sometimes the system stops without reason
- a fuzzy setup has been arisen

Fuzzification of MK-logic

$$K = \{(t, f, u, e) \in [0,1]^4 \mid t + f + u + e = 1\}$$

$\mathbf{k_1} = (t_1, f_1, u_1, e_1)$, $\mathbf{k_2} = (t_2, f_2, u_2, e_2) \in K$

$\mathbf{k_3} = \mathbf{k_1} \sqcup \mathbf{k_2}$ *MK-disjunction*

$$\mathbf{k_3} = (t_3, f_3, u_3, e_3) \qquad \begin{aligned} t_3 &= t_1 + (f_1 + u_1)t_2 \\ f_3 &= f_1 f_2 \\ u_3 &= f_1 u_2 + u_1(f_2 + u_2) \\ e_3 &= e_1 + (f_1 + u_1)e_2 \end{aligned}$$

$\mathbf{k_4} = \mathbf{k_1} \sqcap \mathbf{k_2}$ *MK-conjunction*

$$\mathbf{k_4} = (t_4, f_4, u_4, e_4) \qquad \begin{aligned} t_4 &= t_1 t_2 \\ f_4 &= f_1 + (t_1 + u_1)f_2 \\ u_4 &= t_1 u_2 + u_1(t_2 + u_2) \\ e_4 &= e_1 + (t_1 + u_1)e_2 \end{aligned}$$

The bimonoid of the MK-fuzzy setup

- $\sqcup$ and $\sqcap$ are: non-commutative,
  do not distribute to each other
- $\mathbf{0} = (0, 1, 0, 0), \quad \mathbf{1} = (1, 0, 0, 0)$
- $(K, \sqcup, \mathbf{0}), \quad (K, \sqcap, \mathbf{1})$ monoids
- $\mathbf{k} = (t, f, u, e) \in K$

  $\mathbf{0} \sqcap \mathbf{k} = \mathbf{0}$ but $\mathbf{k} \sqcap \mathbf{0} = (0, t + f + u, 0, e)$
- $(K, \sqcup, \sqcap, \mathbf{0}, \mathbf{1})$ left multiplicative-zero bimonoid

Examples of Bimonoids

- $(M_n(S), \cdot, \odot, I_n, \mathbf{1})$
  - S: non-commutative semiring $(S, +, \cdot, 0, 1)$
  - $M_n(S)$: set of all $n \times n$ maxtrices with elements in S
  - $\cdot$ ordinary multiplication of matrices
  - $\odot$ Hadamard product
  - $\mathbf{1}$: $n \times n$ maxtrix with all elements equal to 1
- $(M_n(S), \cdot, \circledcirc, I_n, I'_n)$
  - $\circledcirc$ binary operation, where $A \circledcirc B = C$ $n \times n$ maxtrix with
    $c_{i,j} = a_{i,1}b_{n,j} + a_{i,2}b_{n-1,j} + \ldots + a_{i,n}b_{1,j}$
  - I'_n: $n \times n$ maxtrix where $i'_{1,n} = i'_{2,n-1} = \ldots = i'_{n,1} = 1$ and the rest equal to 0

$(K, +, \cdot, 0, 1)$: *left multiplicative-zero bimonoid*

Why weighted context-free grammars over bimonoids?

- Runtime verification: Context-free grammars as a specification formalism
  - Efficient monitoring of parametric context-free patterns P.O. Meredith, D. Jin, F. Chen, G. Roșu, *Autom. Softw. Eng.* 17(2010) 149–180. doi:10.1007/s10515-010-0063-y
- Software Model Checking: Context-free grammars for component interfaces
  - Interface Grammars for Modular Software Model Checking, G. Hughes, T. Bultan, in: *Proceedings of ISSTA 2007, ACM* 2007, pp. 39–49. doi:10.1145/1273463.1273471

Weighted context-free grammars over Σ and K

Definition

A *weighted context-free grammar* (wcfg for short) *over* Σ *and* K is a five-tuple $\mathcal{G} = (\Sigma, N, S, R, wt)$ where

- (Σ, N, S, R) context-free grammar with R **linearly ordered**
- $wt : R \to K$ mapping assigning *weights* to the rules

$$w \stackrel{r}{\Longrightarrow}_{\mathcal{G}} u \quad \text{iff} \quad w = w_1 A w_2, \;\; u = w_1 v w_2, \;\; r = A \to v \in R$$

We use only **leftmost** derivations (i.e, $w_1 \in \Sigma^*$)

Weighted context-free grammars over Σ and K

- **derivation of $\mathcal{G}$:** $d = r_0 \dots r_{n-1}$ s.t

  there are $w_i \in (\Sigma \cup N)^*$, $w_i \stackrel{r_i}{\Longrightarrow} w_{i+1}$

  we write $w_0 \stackrel{d}{\Longrightarrow} w_n$

  $weight(d) = wt(r_0) \dots wt(r_{n-1})$

- d derivation of $\mathcal{G}$ for w iff $S \stackrel{d}{\Longrightarrow} w$

Condition

For every $A \in N$ there is not any derivation d of $\mathcal{G}$ such that $A \stackrel{d}{\Longrightarrow} A$.

Weighted context-free grammars over Σ and K

- **series** $\|\mathcal{G}\|$ **of** $\mathcal{G}$

  $w \in \Sigma^*$, $d_1, \ldots, d_m$ all the derivations of $\mathcal{G}$ for w,

$$d_1 \leq_{lex} \ldots \leq_{lex} d_m$$

$$\|\mathcal{G}\|(w) = \sum_{1 \leq i \leq m} weight(d_i)$$

  none derivation of $\mathcal{G}$ for w: $\quad \|\mathcal{G}\|(w) = 0$

- series s **context-free** : if there is wcfg $\mathcal{G}$, $s = \|\mathcal{G}\|$
- $CF(K, \Sigma)$: the class of all context-free series over Σ and K
- $\mathcal{G} = (\Sigma, N, S, R, wt)$ **unambiguous** : if (Σ, N, S, R) unambiguous

Example of wcfg

$\mathcal{G} = (\Sigma, N, S, R, wt)$: unambiguous wcfg over $(K, \sqcup, \sqcap, \mathbf{0}, \mathbf{1})$ and Σ

- (Σ, N, S, R): generates all executions of a concrete program
- finitely many critical errors occuring in an execution
- critical errors: $r \in R$, $wt(r) = (t, f, u, e)$, $e > 0$
- $d = r_0 r_1 \dots r_{n-1}$ derivation of $\mathcal{G}$ for a execution

  at first r_k s.t $wt(r_0) \dots wt(r_k) = (t', f', u', e')$, $e' > 0$

  critical error occurs and the system should stop

Chomsky normal forms

Definition

A wcfg $\mathcal{G} = (\Sigma, N, S, R, wt)$ over Σ and K is said to be

- in *Chomsky normal form* if every rule $r \in R$ is of the form $r = A \to BC$ or $r = A \to a$ with $B, C \in N$ and $a \in \Sigma$,
- in *generalized Chomsky normal form* if every rule $r \in R$ is of the form $r = A \to BC$ or $r = A \to a$ with $B, C \in N$ and $a \in \Sigma \cup \{\varepsilon\}$.

- **chain rule**: rule of the form $A \to B$ and B is variable
- ε-**rule**: rule of the form $A \to \varepsilon$

$\mathcal{G}$ in Chomsky normal form: neither chain rules nor ε-rules

$\mathcal{G}$ in generalized Chomsky normal form: no chain rules

Results

Closure properties of context-free series

- $s_1, s_2 \in CF(K, \Sigma) \Longrightarrow s_1 + s_2 \in CF(K, \Sigma)$
- $s = \|\mathcal{G}\|$, $\mathcal{G}$ unambiguous, $k \in K \Longrightarrow sk = \|\mathcal{G}'\|$, $\mathcal{G}'$ unambiguous

Chomsky normal forms

- $\mathcal{G} = (\Sigma, N, S, R, wt)$ without chain rules and ε-rules. Then, we can effectively construct an equivalent one in Chomsky normal form.
- $\mathcal{G} = (\Sigma, N, S, R, wt)$. Then, we can effectively construct an equivalent one in generalized Chomsky normal form.

- Y alphabet, $\overline{Y} = \{\overline{y} \mid y \in Y\}$ copy

  Dyck language over Y (D_Y): the language of

  $$\mathcal{G}_Y = (Y \cup \overline{Y}, \{S\}, S, R)$$
  $$R = \{S \to yS\overline{y} \mid y \in Y\} \cup \{S \to SS,\ S \to \varepsilon\}$$

- $K[\Sigma \cup \{\varepsilon\}]$: set of all $s \in K\langle\langle\Sigma^*\rangle\rangle$ with $|\mathrm{supp(s)}| = 1$, $\mathrm{supp(s)} \subseteq \Sigma \cup \{\varepsilon\}$

- Δ alphabet, $h : \Delta \to K[\Sigma \cup \{\varepsilon\}]$

  alphabetic morphism induced by h: $h : \Delta^* \to K\langle\langle\Sigma^*\rangle\rangle$

  - $\delta_0, \ldots, \delta_{n-1} \in \Delta$, $h(\delta_i) = k_i.a_i$, $k_i \in K$, $a_i \in \Sigma \cup \{\varepsilon\}$
  - $h(\delta_0 \ldots \delta_{n-1}) = k_0 \ldots k_{n-1}.a_0 \ldots a_{n-1}$
  - $h(\varepsilon) = 1.\varepsilon$

A Chomsky-Schützenberger type result

Theorem

For every $s \in CF(K, \Sigma)$, there are a **linearly ordered** *alphabet $Y \cup \overline{Y}$, a recognizable language L over $Y \cup \overline{Y}$, and an alphabetic morphism $h : Y \cup \overline{Y} \to K[\Sigma \cup \{\varepsilon\}]$ such that $s = h(D_Y \cap L)$.*

- $h(D_Y \cap L) = \sum_{v \in D_Y \cap L} h(v)$
- sum up according to the lexicographic order on $(Y \cup \overline{Y})^*$

Weighted automata over Σ and K

- Weighted automata over K have been already studied.
  - MK-fuzzy automata and MSO logics, M. Droste, T. Kutsia, G. Rahonis, W. Schreiner, in: *Proceedings of GandALF 2017, EPTCS* 256 (2017) 106–120. doi:10.4204/EPTCS.256.8
- Linear order is imposed on states sets.

Definition

A series $s : \Sigma^* \to K$ is called *recognizable* if there is a weighted automaton $\mathcal{A}$ over Σ and K such that $s = \|\mathcal{A}\|$.

Recognizable and context-free series relation

Definition

A wcfg $\mathcal{G} = (\Sigma, N, S, R, wt)$ over Σ and K is called *right-linear* if its rules are of the form $A \to aB$, $A \to a$, or $A \to \varepsilon$ where $B \in N$ and $a \in \Sigma$.

Theorem

Let Σ be a **linearly ordered** *alphabet. Then a series $s \in K\langle\langle\Sigma^*\rangle\rangle$ is generated by a right-linear wcfg over Σ and K iff it is recognized by a weighted automaton over Σ and K.*

Open Problems (under investigation)

- Closure under scalar product $ks,\ k \in K,\ s \in CF(K, \Sigma)$
- Closure under Cauchy product

  $s, r \in K\langle\langle\Sigma^*\rangle\rangle,\ \ w = a_0 \dots a_{n-1} \in \Sigma^*,\ \ a_i \in \Sigma$

  $sr(w) = (s(\varepsilon)r(a_0 \dots a_{n-1})) + (s(a_0)r(a_1 \dots a_{n-1})) + \cdots + (s(w)r(\varepsilon))$

  $sr(w) = (s(a_0 \dots a_{n-1})r(\varepsilon)) + (s(a_0 \dots a_{n-2})r(a_{n-1})) + \cdots + s(\varepsilon)r(w))$
- Weighted pushdown automata over Σ and K

Thank you!

Ευχαριστώ!