Rational Expressions for Ungraded Tree Languages

Younes Guellouma[1], Hadda Cherroun[1], Mohamed Habib Maicha[1], and Djelloul Ziadi[2]

[1]Laboratoire d'informatique et de mathématiques, Université Amar Télidji Laghouat, BP 37G, Route de Ghardaïa 03000 Laghouat, Algérie,
{y.guellouma,hadda_cherroun,m.maicha}@lagh-univ.dz
[2]Laboratoire LITIS - EA 4108, Normandie Université, Rouen, France,
Djelloul.Ziadi@univ-rouen.fr

Abstract

A hedge automaton can be used to recognize a finite or an infinite ungraded trees set using string REGEX (hedges). However, the use of hedges cannot represent the parent-child relation in trees and model merely only the horizontal link between trees. In this paper, we propose a new rational expressions to represent all the set of trees recognizable by hedge automata.

Key words : Hedge automata, rational expressions, tree languages.

1 Introduction

In the last few years, tree automata start to draw attention in many fields in applied mathematics and computer science like in Xml Schemas [1, 2], cryptography [3, 4], term rewriting [5] etc.

Kleene theorem reports that there exists an equivalence between accepted (noted $Rec(\Sigma)$) and rational (noted $Rat(\Sigma)$) tree languages for a set of graded symbols Σ [6, 7, 8].

However, no work deals -in our best knowledge- with rational expressions for graded tree automata.

In this paper, we describe a new rational expressions for hedge automata which are natural. Indeed, Hedge automata are perfect data structures that present ungraded trees. String rational expressions (REGEX) are used to denote horizontal link between tree nodes. We adapt a rational tree expression to generalise these REGEX to the parent-child relation in trees.

2 Preliminaries

An alphabet Σ is a finite set of symbols. The set U_Σ of ungraded trees is the smallest set satisfying:

- $\Sigma \subset U_\Sigma$ and
- if $f \in \Sigma, t_1, \ldots, t_n \in U_\Sigma$, $n \in \mathbb{N}$ then $f(t_1 \ldots t_n) \in U_\Sigma$.

For an ungraded tree $f(t_1 \ldots t_n)$, the sequence $t_1 \ldots t_n$ is called a *hedge*. We note that a set of hedges forms a string language. An ungraded tree language L is any subset of U_Σ.

By the same way, one can define the Hedge rational expression using rational expressions on word to define hedges or horizontal word language. However, to reach more simplicity and thus subtract more naturally hedges for the HTA, we use a set of variables to define that word rational expressions.

Hedge automata is a model that recognizes a possibly infinite set of ungraded tree language. We will not address the regularity of the recognizable language of hedge automata in this paper. Nevertheless, the proposed rational expression is used to prove the assumed regularity [7, 9].

Definition 1 *A hedge automaton $\mathcal{H}$ is the tuple $\mathcal{H} = (Q, \Sigma, Q_f, \Delta)$ where :*

1. *Q is a set of states,*

2. Σ *is an ungraded alphabet,*
3. $Q_f \subseteq Q$ *is a set of final states and*
4. $\Delta \subset \Sigma \times \mathrm{Rat}(Q) \times Q$ *is a finite set of transitions.* $\mathrm{Rat}(Q)$ *is the set of REGEX over* Q.

The output function $\delta : U_\Sigma \to 2^Q$ that recognizes a tree $t = f(t_1 \ldots t_n)$ can be defined as follows.

$$\delta(t) = \{q \mid \exists (f, E, q) \in \Delta \text{ and } \forall < i \leq n, \delta(t_1) \ldots \delta(t_n) \cap [\![E]\!] \neq \emptyset \tag{2.1}$$

Example 1 *Let* $\mathcal{H} = (Q, \Sigma, Q_f, \Delta)$ *such that :*

1. $Q = \{q_1, q_2\}$,
2. $\Sigma = \{a, b\}$,
3. $Q_f = \{q_2\}$,
4. $\Delta = \{(a, 1, q_1), (a, q_1^*, q_2), (b, q_1 + q_2, q_2)\}$.

The tree $t = a(b, a(a, a))$ *is not recognized by the automaton because* b *cannot be used by the transitions set. However, the tree* $t' = b(b(a))$ *can be recognized by* $\mathcal{H}$ *using* q_1 *to replace* a *and* q_2 *to replace* $b(q_1)$*. Then,* q_2 *can replace* $b(q_2)$ *since* $q_2 \in [\![q_1 + q_2]\!]$.

Before announcing the rational expressions for ungraded trees, we define the some needed notions.

Given a symbol $c \in \Sigma$, the *c-product* is the operation $\cdot_c$ defined for a tree t in $U(\Sigma)$ and a tree language L by

$$t \cdot_c L = \begin{cases} L & \text{if } t = c, \\ \{d\} & \text{if } t = d \in \Sigma \setminus \{c\}, \\ f(t_1 \cdot_c L \ldots t_n \cdot_c L) & \text{otherwise (if } t = f(t_1 \ldots t_n)) \end{cases} \tag{2.2}$$

This c-product is extended for any two tree languages L and L' by $L \cdot_c L' = \bigcup_{t \in L} t \cdot_c L'$.

We use the *multiple concatenation* for sake of simplicity.

let $\{x_1, \ldots, x_n\}$ be any set. Let $L, L_1, \ldots, Ln$ be n ungraded languages then:

$$L._{<x_1,\ldots,x_n>} < L_1, \ldots, L_n >= ((L._{x_1} L_1)._{x_2} \ldots ._{x_n} L_n \tag{2.3}$$

3 Rational expressions for ungraded tree languages

Effectively, one can give many definitions of rational expressions for ungraded languages. For instance, the word concatenation can be done using a $._\epsilon$ concatenation using a special symbol ϵ not in Σ. However, we adopt a definition that exhibit the horizontal relation in trees.

Subsequently, we define rational expressions for ungraded trees over an ungraded alphabet Σ as follows.

Definition 2 *A* Rational expressions for ungraded alphabet *(REU)* E *over* Σ *is inductively defined as follows.*

$$E = 0, \quad E = f(\varphi(\sigma))._{<\sigma>} < E_1, \ldots, E_n >,$$
$$E = E_1 + E_2, \quad E = E_1 \cdot_c E_2, \quad E = E_1^{*_c}$$

where $E_1, \ldots, E_n$ *are REU over* (Σ)*,* $\varphi(\sigma)$ *is a REGEX over* Σ^**,* $c \in \Sigma$ *and* $n = [\varphi(\sigma)]$.

The semantics $[\![E]\!]$ *of a REU* E *is defined as follows.*

$$[\![0]\!] = \emptyset,$$
$$[\![f(\varphi(\sigma))._{<\sigma>} < E_1, \ldots, E_n >]\!] \;= f([\![\varphi(\sigma)]\!])._{<\sigma>} < [\![E_1]\!], \ldots, [\![E_n]\!] >$$
$$[\![E_1 + E_2]\!] = [\![E_1]\!] \cup [\![E_2]\!], \quad [\![E_1 \cdot_c E_2]\!] = [\![E_1]\!] \cdot_c [\![E_2]\!], \quad [\![E_1^{*_c}]\!] = ([\![E_1]\!])^{*_c}$$

In fact, the hedge notion is not distinguishable in this definition since concatenation symbols for horizontal and vertical links are the same. We extend therefore this definition to meet this requirement.

Definition 3 *A* Rational Hedge Expressions *(RHE) E over $(\Sigma, X), X \cap \Sigma = \emptyset$ is inductively defined as follows.*

$$E = 0, \quad E = f(\varphi(X)).{}_{<X>} < E_1, \ldots, E_n >,$$
$$E = E_1 + E_2, \quad E = E_1 \cdot_c E_2, \quad E = E_1^{*c}$$

where $E_1, \ldots, E_n$ are REU over (Σ, X), $\varphi(X)$ is a REGEX over X^*, $c \in \Sigma$ and $n = ||\varphi(X)||$.

Now, we can prove that the semantics of a RHE such defined is free from X.
Consequently,

Lemma 1 *Let E be a RHE over (Σ, X). Then, $[\![E]\!] \subseteq U_\Sigma$.*

Proof The proof can be done inductively on the structure of E.

1. If $E = a, a \in \Sigma$ then $[\![E]\!] \subset U_\Sigma$ since X and Σ are disjoint.
2. If $E = E_1 + E_2$ such that E_1 and E_2 are free from X then by definition we have $[\![E]\!] \subseteq U_\Sigma$.
3. If $E = E_1 \cdot_c E_2$ such that E_1 and E_2 are free from X. Then E is free from X since $c \notin X$.
4. If $E = E_1^{*c}$ such that E_1 is free from X then $[\![E]\!]$ does not contain any element from X.
5. If $E = f(\varphi(X)).{}_{<X>} < E_1, \ldots, E_n >$ such that $E_1, \ldots, E_n$ does not contain any element from X. The fact that E is free from X leads to $\forall x \in \varphi(X) = \{x_1, \ldots, x_n\}$. If $x' \notin \varphi(X)$ then $f(\varphi(X)).{}_{x'} E_i, 1 \leq i \leq n = 0$. Otherwise, every element x from $\varphi(X)$ will be replaced by an expression E_i which is free from X.

□

We can generalize the Kleene result to ungraded trees.

Theorem 1 *The class of recognizable ungraded tree languages and the class of rational ungraded tree languages are equivalent.*

This theorem can be proved by two constructions. The first one creates for a RHE E a hedge automaton $\mathcal{H}$ such that $L(\mathcal{H}) = [\![E]\!]$. The second one tries to find a RHE E that denotes the recognized language of a hedge automaton.

4 Conclusion

In this paper, we proposed a generalization of rational expression from trees to ungraded tree languages. We mentioned that the hedge automata languages coincide with that of the proposed rational expressions. This result can be used to propose generalizations of different conversions which can exist between hedge automata and rational expressions namely Thompson, derivatives and position from one hand, Arden and Mc Naughton -Yamada transformation from the other one.

References

[1] Silvano Zilio and Denis Lugiez. Xml schema, tree logic and sheaves automata. In Robert Nieuwenhuis, editor, *Rewriting Techniques and Applications*, volume 2706 of *Lecture Notes in Computer Science*, pages 246–263. Springer Berlin Heidelberg, 2003.

[2] Wouter Gelade, Tomasz Idziaszek, Wim Martens, Frank Neven, and Jan Paredaens. Simplifying XML schema: Single-type approximations of regular tree languages. *J. Comput. Syst. Sci.*, 79(6):910–936, 2013.

[3] David Monniaux. Abstracting cryptographic protocols with tree automata. In Agostino Cortesi and Gilberto Filé, editors, *Static Analysis*, volume 1694 of *Lecture Notes in Computer Science*, pages 149–163. Springer Berlin Heidelberg, 1999.

[4] Hubert Comon and Véronique Cortier. Tree automata with one memory set constraints and cryptographic protocols. *Theoretical Computer Science*, 331(1):143 – 214, 2005. Automata, Languages and Programming.

[5] Alfons Geser, Dieter Hofbauer, Johannes Waldmann, and Hans Zantema. On tree automata that certify termination of left-linear term rewriting systems. *Information and Computation*, 205(4):512 – 534, 2007. Special Issue: 16th International Conference on Rewriting Techniques and Applications.

[6] J.W. Thatcher and J.B. Wright. Generalized finite automata theory with an application to a decision problem of second-order logic. *Mathematical systems theory*, 2(1):57–81, 1968.

[7] H. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications, 2007. release October, 12th 2007.

[8] Loek Cleophas. *Tree Algorithms: Two Taxonomies and a Toolkit.* PhD thesis, Eindhoven University of Technology, april 2008.

[9] Anne Brüggemann-Klein, Makoto Murata, and Derick Wood. Regular tree and regular hedge languages over unranked alphabets: Version 1, 2001.

[10] Y. Guellouma H. Cherroun. From Tree Automata to Rational Tree Expression. *International Journal of Foundations of Computer Science*, Vol. 29, No. 6 (2018) 10451062 2018.