Theoretical Computer Science 610 (2016) 37–58

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Ultrafilters on words for a fragment of logic ☆

Mai Gehrke [a], Andreas Krebs [b], Jean-Éric Pin [a]

[a] LIAFA, CNRS and Univ. Paris-Diderot, Case 7014, 75205 Paris Cedex 13, France
[b] Wilhelm-Schickard-Institut für Informatik, Universität Tübingen, Germany

ARTICLE INFO

Article history:
Received 8 December 2014
Received in revised form 3 August 2015
Accepted 9 August 2015
Available online 14 August 2015

Keywords:
Ultrafilters
Formal languages
Profinite equations
Regular languages
Stone duality

ABSTRACT

We give a method for specifying ultrafilter equations and identify their projections on the set of profinite words. Let $\mathcal{B}$ be the set of languages captured by first-order sentences using unary predicates for each letter, arbitrary uniform unary numerical predicates and a predicate for the length of a word. We illustrate our methods by giving ultrafilter equations characterising $\mathcal{B}$ and then projecting these to obtain profinite equations characterising $\mathcal{B} \cap \mathrm{Reg}$. This suffices to establish the decidability of the membership problem for $\mathcal{B} \cap \mathrm{Reg}$.

This paper is the third step of a programme aiming at widening the notion of recognisability using methods from topological duality theory. In two earlier papers, Gehrke, Grigorieff, and Pin proved the following results:

Result 1. *(See [4].) Any Boolean algebra of regular languages can be defined by a set of equations of the form $u \leftrightarrow v$, where u and v are profinite words.*

Result 2. *(See [5].) Any Boolean algebra of languages can be defined by a set of equations of the form $u \leftrightarrow v$, where u and v are ultrafilters on the set of words.*

These two results can be summarised by saying that Boolean algebras of languages can be defined by *ultrafilter equations* and by *profinite equations* in the regular case. When a Boolean algebra is closed under quotients, we use the notation $u = v$ instead of $u \leftrightarrow v$, for a reason that will be fully explained in Section 1.3.

Restricted instances of Result 1 have been obtained and applied very successfully long before the result was stated and proved in full generality. It is in particular a powerful tool for characterising classes of regular languages or for determining the expressive power of various fragments of logic, see the book of Almeida [2] or the survey [9] for more information.

Result 2 however is still awaiting convincing applications and even an idea of how to apply it in a concrete situation. The main problem in putting it into practise is to cope with ultrafilters, a difficulty nicely illustrated by Jan van Mill, who cooked up the nickname *three headed monster* for the set of ultrafilters on $\mathbb{N}$. Facing this obstacle, the authors thought of using Results 1 and 2 simultaneously to obtain a new proof of the equality

☆ Work supported by the project ANR 2010 BLAN 0202 02 FREC.

E-mail addresses: mgehrke@liafa.univ-paris-diderot.fr (M. Gehrke), krebs@informatik.uni-tuebingen.de (A. Krebs), Jean-Eric.Pin@liafa.univ-paris-diderot.fr (J.-É. Pin).

http://dx.doi.org/10.1016/j.tcs.2015.08.007
0304-3975/

$$\mathbf{FO}[\mathcal{N}] \cap \mathrm{Reg} = [\![\, (x^{\omega-1}y)^{\omega+1} = (x^{\omega-1}y)^{\omega} \text{ for } x, y \text{ words of the same length}]\!]. \tag{1}$$

where $[\![E]\!]$ denotes the class of languages defined by a set E of equations. This formula gives the profinite equations characterising the regular languages in $\mathbf{FO}[\mathcal{N}]$, the class of languages defined by sentences of first order logic using arbitrary numerical predicates and the usual letter predicates. This result follows from the work of Barrington, Straubing and Thérien [3] and Straubing [10] and is strongly related to circuit complexity. Indeed its proof makes use of the equality between $\mathbf{FO}[\mathcal{N}]$ and AC^0, the class of languages accepted by unbounded fan-in, polynomial size, constant-depth Boolean circuits [11, Theorem IX.2.1, p. 161]. See also [7] for similar results and problems.

However, before attacking this problem in earnest we have to tackle the following questions: how does one get hold of an ultrafilter equation given the non-constructibility of each one of them (save the trivial ones given by pairs of words)? In particular, how does one generalise the powerful use in the regular setting of the ω-power? And how does one project such ultrafilter equations to the regular fragment? In answering these questions and facing these challenges, we have chosen to consider a smaller and simpler logic fragment first. Our choice was dictated by two parameters: we wanted to be able to handle the corresponding ultrafilters and we wished to obtain a reasonably understandable list of profinite equations. Finally, we opted for $\mathbf{FO}[\mathcal{N}_0, \mathcal{N}_1^u]$, the restriction of $\mathbf{FO}[\mathcal{N}]$ to constant numerical predicates and to uniform unary numerical predicates. Here we obtain the following result (Theorem 5.16)

$$\mathbf{FO}[\mathcal{N}_0, \mathcal{N}_1^u] \cap \mathrm{Reg} = [\![(x^{\omega-1}s)(x^{\omega-1}t) = (x^{\omega-1}t)(x^{\omega-1}s), \; (x^{\omega-1}s)^2 = (x^{\omega-1}s) \text{ for } x, s, t \text{ words of the same length}]\!], \tag{2}$$

which shows in particular that membership in $\mathbf{FO}[\mathcal{N}_0, \mathcal{N}_1^u]$ is decidable for regular languages.

Although this result is of interest in itself, we claim that our *proof method* is more important than the result. Indeed, this case study demonstrates for the first time the workability of the ultrafilter approach.

This method can be summarised as follows. First we find a set of ultrafilter equations characterising $\mathbf{FO}[\mathcal{N}_0, \mathcal{N}_1^u]$ (Theorems 3.2, 3.3, and 4.7). Projecting these ultrafilter equations onto profinite words, we obtain profinite equations characterising $\mathbf{FO}[\mathcal{N}_0, \mathcal{N}_1^u] \cap \mathrm{Reg}$ (Theorem 5.2). Finally we show that the simpler class (2) generates the full family of projections of our ultrafilter equations to obtain Theorem 5.16.

In the conference version of this paper [6], we had only proved the validity in $\mathcal{B}$ of the equations given in Section 3. Here we also prove their completeness in Section 4. As a consequence, we get a new completeness result for $\mathcal{B} \cap \mathrm{Reg}$ obtained by projection in Section 5.1. This leads to a new proof of decidability of membership in $\mathcal{B} \cap \mathrm{Reg}$ in Section 5.2. The completeness result expressed by equation (2) above is then obtained from the completeness result in Section 5.1 by rewriting in Section 5.3. In [6], the completeness part of (2) was proved by traditional automata theoretic means.

1. Stone duality and equations

In this paper, given a subset S of a set E, we denote by S^c the complement of S in E.

1.1. Filters and ultrafilters

Let X be a set. A *Boolean algebra of subsets* of X is a subset of $\mathcal{P}(X)$ containing the empty set and closed under finite intersections, finite unions and complement. Let $\mathcal{B}$ be a Boolean algebra of subsets of X. An *ultrafilter* of $\mathcal{B}$ is a nonempty subset γ of $\mathcal{B}$ such that:

(1) the empty set does not belong to γ,
(2) if $K \in \gamma$ and $K \subseteq L$, then $L \in \gamma$ (closure under extension),[1]
(3) if $K, L \in \gamma$, then $K \cap L \in \gamma$ (closure under intersection),
(4) for every $L \in \mathcal{B}$, either $L \in \gamma$ or $L^c \in \gamma$ (ultrafilter condition).

Nonempty subsets of $\mathcal{B}$ satisfying just conditions (2) and (3) above are called *filters*, while filters also satisfying (1) are said to be *proper*. A subset $\mathcal{S}$ of $\mathcal{B}$ is a *filter subbasis* if it has the *finite intersection property*: every finite intersection of elements of $\mathcal{S}$ is nonempty. In this case, the set of all supersets of finite intersections of elements of $\mathcal{S}$ is a proper filter, called the *filter generated by* $\mathcal{S}$.

A nonempty subset $\mathcal{S}$ of $\mathcal{B}$ is a *filter basis* if it does not contain the empty set and if, for every $K, L \in \mathcal{S}$, there exists $M \in \mathcal{S}$ such that $M \subseteq K \cap L$. In this case, the filter generated by $\mathcal{S}$ is the set of all supersets of elements of $\mathcal{S}$. Note that if $\mathcal{S}$ and $\mathcal{T}$ are filter basis, then $\mathcal{S} \cup \mathcal{T}$ is a filter subbasis if and only if the intersection of any member of $\mathcal{S}$ with any member of $\mathcal{T}$ is nonempty.

[1] In other words, γ is an *upset*.