Journal of Computer and System Sciences 80 (2014) 457–467

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

On the role of complementation in implicit language equations and relations

Adrian Ionescu [a,*], Ernst L. Leiss [b]

[a] Department of Mathematics and Computer Science, Wagner College, Staten Island, NY 10301, United States
[b] Department of Computer Science, University of Houston, Houston, TX 77204, United States

ARTICLE INFO

Article history:
Received 2 July 2001
Accepted 5 September 2013
Available online 11 September 2013

Keywords:
Languages
Equations
Relations
Implicit
Complementation

ABSTRACT

We solve systems of boolean implicit equations and relations with union and complementation. If the languages are regular, we determine whether a system of implicit boolean equations or relations has solutions. If there is a solution, we provide an effective construction of a (regular) solution. We give representations for maximal and minimal solutions. Moreover, we also solve the problem of uniqueness of solutions as well as whether infinitely many solutions exist.

1. Introduction

Language equations have a long history, starting with the most basic equation $X = L \cdot X \cup M$ whose solution is L^*M (unique if L does not contain the empty word λ) and which is usually used to show that for any finite automaton accepting a regular language R, there exists a regular expression α that denotes exactly R. Another application of language equations (here boolean language equations) is used in the characterization of sequential networks [1]. This hints at a fundamental problem in how language equations [4] are generally viewed – as tool to do other things rather than as an object of interest in their own right. This persists in spite of the denotational elegance of equations which makes use of the versatility of expressions, with their inherent ability of accommodating a wide variety of operators, in contrast to the more "operational" approach taken by automata. But even this slights the expressive power of language equations, as one can see from the following observation. It is well known that permitting complementation in regular expressions (which are then often called extended regular expressions) does not extend the class of languages denoted in this way; it only increases the succinctness of the notation [10]. In other words, the language denoted by an extended regular expression is still regular. This is no longer the case when studying equations, as in [7] a very simple (explicit) language equation (indeed over a one-letter alphabet!) involving concatenation and complementation was shown to have a unique non-regular solution. In other words, when considering equations permitting complementation does increase their denotational power, in contrast to expressions where it does not [8]. This then is the underlying motivation for the work reported below – we show how to handle complementation in the context of implicit equations. Note that the question of explicit equations has been addressed in [3,6].

In the remainder of this section we first amplify our comments above about the contributions of language equations, then we briefly sketch our notation. In Section 2, we study comprehensively implicit equations where the operators are all boolean operators, paying special emphasis to determining existence and uniqueness of solutions as well as to the

* Corresponding author.
E-mail addresses: ionescu@wagner.edu (A. Ionescu), coscel@cs.uh.edu (E.L. Leiss).

0022-0000/$ – see front matter
http://dx.doi.org/10.1016/j.jcss.2013.09.001

complexity of the problems solved. In Section 3, we extend these results to containment, in addition to equality. In Section 4 we work out several examples; Section 5 gives the conclusion.

Language relations are defined by using constant languages and a set of operations. There are five areas in which important contributions have been made [9]:

1. The complete solution of explicit equations (each variable having a single equation) with union and left-concatenation.
2. A comprehensive coverage of complementation for explicit equations with union and left-concatenation.
3. The complete solution of explicit equations over a one-letter alphabet with union, concatenation, and star.
4. Certain explicit equations over a one-letter alphabet with concatenation and complementation having only non-context-free solutions.
5. The complete solution of implicit equations and relations with union and left-concatenation.

In order to define languages we start with an alphabet A. The choice of the alphabet is quite important. For example, if $A = \{a\}$ all resulting languages $L \subseteq A^*$ are commutative. We also choose a set of constant languages (CONST). In general, any subset of the set of all languages over A can be chosen for CONST. However, for practical reasons the well known classes of languages are usually considered: regular languages (REG), context-free languages, context-sensitive languages, recursive languages, recursive enumerable languages. We will mainly be interested in REG because of their "nice" closure properties [5].

The basic language operations (OP) are the following ones: union ($\cup$), intersection ($\cap$), complementation ($^-$), concatenation ($\cdot$), star (*). In our case, we will mainly be interested in union and complementation (and therefore intersection). As noted in [9], complementation is a difficult operation. To mention just one reason, note that allowing complementation in regular expressions will result in nonelementary complexity [10]. For other results involving maximal and minimal solutions for language equations see [2].

We define the set $\mathrm{EX}_A(\mathrm{CONST};\mathrm{OP};X_1,\dots,X_n)$ of expressions in the constant languages CONST and in the variables $\{X_1,\dots,X_n\}$ with operations in OP. For $\alpha \in \mathrm{EX}_A(\mathrm{CONST};\mathrm{OP};X_1,\dots,X_n)$ and $L \in \mathrm{CONST}$ one defines explicit equations $X_i = \alpha(X_1,\dots,X_n)$ and implicit equations $L = \alpha(X_1,\dots,X_n)$.

For example, the original language equation $X = L \cdot X \cup M$ is an explicit equation $X = \alpha$ with α an element of $\mathrm{EX}_A(\mathrm{CONST};\mathrm{OP};X)$ where CONST is the class of all languages over the alphabet A and OP is the set consisting of the operators union and concatenation. Similarly, the equation over a one-letter alphabet whose solution is non-regular (and therefore non-context free) is $X = \beta$ with β given by $a \cdot \overline{\overline{\overline{X} \cdot \overline{X}} \cdot \overline{\overline{X} \cdot \overline{X}}} \cdot \overline{\overline{\overline{X} \cdot \overline{X}} \cdot \overline{\overline{X} \cdot \overline{X}}}$. Thus, β is an expression of $\mathrm{EX}_A(\mathrm{CONST};\mathrm{OP};X)$ where CONST is the set of all languages over the alphabet $\{a\}$ and OP consists of the operators concatenation and complementation. Finally, here is an example of an implicit equation: $ab^* = \gamma$ where $\gamma = a \cdot X \cup b \cdot Y$ is an expression in $\mathrm{EX}_A(\mathrm{CONST};\mathrm{OP};X)$ where the alphabet A is arbitrary, CONST is the class of all regular languages over A, and OP consists of union and concatenation. (This last equation has the unique solution $X = b^*$ and $Y = \emptyset$.)

2. Implicit equations with union and complementation

Let α be an expression in $\mathrm{EX}_A(\mathrm{CONST};\mathrm{OP};X_1,\dots,X_n)$, and assume that OP consists of union and complementation and that CONST=REG. Let us consider the following system of implicit boolean equations:

$$L_i = \alpha_i(X_1,\dots,X_n), \quad i = 1,2,\dots,m. \tag{1}$$

Note that, without loss of generality, one can assume that the α_i are boolean functions *only* in the variables $\{X_1,\dots,X_n\}$. Indeed, if constant languages appear in α_i, each such language L can be replaced by the extra variable Y and another equation $L = Y$ is added to the system. For example, $a^* = X \cup a$ can be replaced by the equivalent system $a^* = X \cup Y$, $a = Y$. The following two propositions are immediate:

Proposition 1. *If system (1) of implicit boolean equations has a solution $(S_1,\dots,S_n)$, then for all $w \in A^*$ and $i = 1,2,\dots,m$*

$$L_i \cap \{w\} = \alpha_i\big(S_1 \cap \{w\},\dots,S_n \cap \{w\}\big) \cap \{w\} \tag{2}$$

is also true.

Conversely, one also obtains the following result

Proposition 2. *If for all $w \in A^*$, system (2) has a solution given by*

$$(S_{w,1},\dots,S_{w,n}),$$

then system (1) has a solution $(S_1,\dots,S_n)$ given by

$$S_i = \bigcup_{w \in A^*} S_{w,i}.$$