Theoretical Computer Science 618 (2016) 52–71

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Input-driven languages are linear conjunctive ☆

Alexander Okhotin [1]

Department of Mathematics and Statistics, University of Turku, Turku FI-20014, Finland

ARTICLE INFO

Article history:
Received 7 May 2015
Received in revised form 5 January 2016
Accepted 7 January 2016
Available online 12 January 2016
Communicated by D. Perrin

Keywords:
Context-free grammars
Conjunctive grammars
Cellular automata
Trellis automata
Pushdown automata
Input-driven automata
Visibly pushdown automata

ABSTRACT

Linear conjunctive grammars define the same family of languages as one-way real-time cellular automata (A. Okhotin, "On the equivalence of linear conjunctive grammars to trellis automata", *RAIRO ITA*, 2004), and this family is known to be incomparable with the context-free languages (V. Terrier, "On real-time one-way cellular array", *Theoret. Comput. Sci.*, 1995). This paper demonstrates the containment of the languages accepted by input-driven pushdown automata (a.k.a. visibly pushdown automata) in the family of linear conjunctive languages, which is established by a direct simulation of an input-driven automaton by a one-way real-time cellular automaton. On the other hand, it is shown that the language families defined by the unambiguous grammars, the LR(k) grammars and the LL(k) grammars are incomparable with the linear conjunctive languages.

1. Introduction

The theory of formal grammars is centred around a simple fundamental model, which is historically known under the name of a *context-free grammar*. Whereas other "context-sensitive" models proposed by Chomsky in his early papers were quickly found to be unrelated to syntactic descriptions of any kind, the context-free model was universally accepted as the standard mathematical model of syntax, that is, the ordinary kind of formal grammars. Many other practically important subfamilies of these grammars were quickly identified, such as the unambiguous grammars, the linear grammars, the LL(k) grammars [31], the LR(k) grammars [18], the bracketed grammars [12] and their variants. These models formed the core of formal language theory.

The following years saw many attempts at investigating formal grammars beyond the ordinary grammars. At first, there was no general theory on what kind of models would be suitable for representing syntax, and some successful definitions were initially based upon rather unobvious ideas, such as inserting subtrees into parse trees [17]. An understanding of these extensions came only in the 1980s, with the idea of parsing as deduction [30] leading to a remarkable paper by Rounds [32], who finally explained various kinds of grammars as fragments of the first-order logic over positions in a string, with recursive definitions under least fixed point semantics.

In light of this understanding, formal grammars are regarded as a specialized logic for describing the syntax of languages, where properties of strings (or their *syntactic categories*) are defined inductively by combining substrings with known properties. Then, ordinary grammars (Chomsky's "context-free") use substrings as basic objects, concatenation as the only function

☆ A preliminary version of this paper, entitled "Comparing linear conjunctive languages to subfamilies of the context-free languages" was presented at the SOFSEM 2011 conference held in Nový Smokovec, Slovakia, on 22–28 January 2011, and its extended abstract appeared in the conference proceedings.

E-mail address: alexander.okhotin@utu.fi.

[1] Supported by the Academy of Finland under grants 134860 and 257857.

http://dx.doi.org/10.1016/j.tcs.2016.01.007

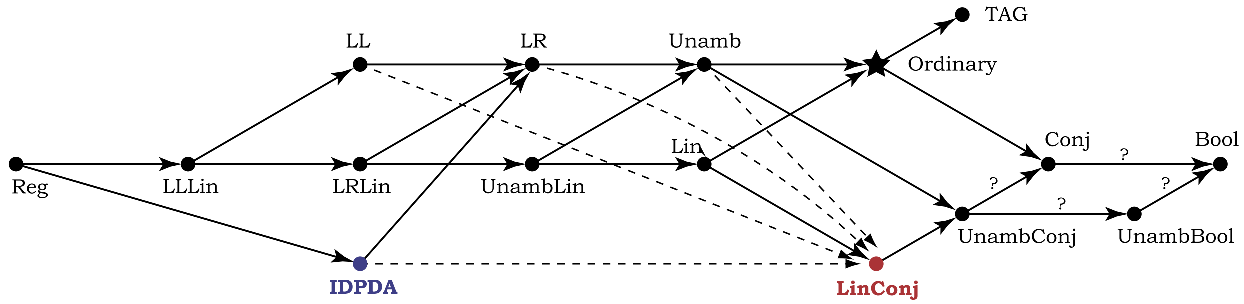

Fig. 1. The known hierarchy of formal grammars (solid lines) and possible inclusions to be investigated in this paper (dotted lines). The families depicted are those defined by finite automata (*Reg*), by (linear) LL(k) grammars (*LLLin*, *LL*), by (linear) LR(1) grammars (*LRLin*, *LR*), by (linear) unambiguous grammars (*UnambLin*, *Unamb*), by ordinary context-free grammars (*Ordinary*), by linear grammars (*Lin*), by tree-adjoining grammars (*TAG*), by input-driven pushdown automata (*IDPDA*), by (linear) conjunctive grammars (*LinConj*, *Conj*), by unambiguous conjunctive and Boolean grammars (*UnambConj*, *UnambBool*), and by Boolean grammars (*Bool*).

and disjunction as the only logical connective. Other kinds of grammars are variants of this definition, and are organized into the hierarchy depicted in Fig. 1, where each solid arrow represents containment of one family in another. These inclusions are known to be proper in all cases, except for those labelled with question marks.

The hierarchy is centred around the ordinary formal grammars (*Ordinary*). Their simplest extensions are those featuring extra logical connectives in the rules, namely, *conjunctive grammars* [22] (*Conj*) with conjunction, and *Boolean grammars* [25, 20] (*Bool*) further equipped with negation. Extensions of another kind maintain disjunction as the only logical connective, but extend the basic objects and the operations on them: the simplest of these grammars are the *tree-adjoining grammars* [17] (*TAG*), which deal with pairs of strings instead of strings, and use wrapping of one pair around another as the only function. The latter model and its further extensions received much attention in computational linguistics.

Other families presented in Fig. 1 restrict these families in one or another way. First, there are the unambiguous subclasses of these grammars (*Unamb*, *UnambConj*, *UnambBool*), which define a unique parse for every string they generate; these grammars are notable for having square-time parsing algorithms [26]. Further below, there are left-to-right deterministic variants of grammars that admit linear-time parsing: the *LR*(k) *grammars* [18] (*LR*), also known for their equivalent representation by deterministic pushdown automata [11], and the *LL*(k) *grammars* [31] (*LL*), which can be parsed by recursive descent with k symbols of lookahead. Another small subfamily of the LR(k) grammars are the *bracketed grammars* [12], which later evolved into *input-driven pushdown automata* [21] (*IDPDA*), also known under the name of *visibly pushdown automata* [2]. The hierarchy also features subclasses of some of these families, in which the concatenation is restricted to linear (*LLLin*, *LRLin*, *UnambLin*, *LinConj*). The largest of these is the family of linear conjunctive languages.[2]

This purpose of this paper is to investigate the hierarchy of formal grammars presented in Fig. 1, and to settle four previously unresolved relations marked in the figure by dotted arrows. Each of these relations might turn out to be a proper containment, although incomparability would appear more likely, because those grammar families are defined by incomparable restrictions to the underlying logic.

As shown in Fig. 1, all results in this paper are concerned with possible containment of other language families in the family of linear conjunctive languages (*LinConj*). Consider first the whole family of *conjunctive grammars* [22], which extend the ordinary grammars with an additional conjunction operation that expresses a substring satisfying multiple conditions simultaneously. These grammars have greater expressive power than the ordinary grammars, and yet inherit their main practical properties, such as subcubic-time parsing [28], and offer a new field for theoretical studies, which is elaborated in a recent survey paper [27]. Their special case, in which concatenation can be taken only with terminal strings, is known as *linear conjunctive grammars* [22].

Linear conjunctive grammars are notable for having an equivalent representation by one of the simplest types of cellular automata. These are the *one-way real-time cellular automata*, also known as *trellis automata*, studied by Dyer [9], Čulík, Gruska and Salomaa [7,8], Ibarra and Kim [13], Terrier [34], and others. These automata work in real time, making $n-1$ parallel steps on an input of length n, and the next value of each cell is determined only by its own value and the value of its right neighbour. Precise definitions of linear conjunctive grammars and trellis automata are recalled in Section 2, along with the computational equivalence result [23].

The main result of this paper is that every language recognized by an input-driven pushdown automaton (*IDPDA*) is linear conjunctive (*LinConj*), so that one of the dotted lines in Fig. 1 becomes a solid line. An *input-driven pushdown automaton* (IDPDA) is a special case of a deterministic pushdown automaton, in which the input alphabet Σ is split into three disjoint subsets, Σ_{+1}, Σ_{-1} and Σ_0, and the type of the input symbol determines the type of the operation with the stack (push, pop, or ignore, respectively). These automata develop the idea behind *bracketed grammars* [12]; they were first studied by

[2] It is interesting to note that none of these models allow context-dependent rules of any kind. This makes Chomsky's term "context-free grammar" redundant, as each grammar family in Fig. 1 could be nicknamed "context-free". For that reason, this paper refers to Chomsky's "context-free grammars" as the ordinary grammars.