Revue française d'automatique informatique recherche opérationnelle. Mathématique

Derick Wood

Bicolored digraph grammar systems

Revue française d'automatique informatique recherche opérationnelle. Mathématique, tome 7, nº R1 (1973), p. 45-52

<http://www.numdam.org/item?id=M2AN_1973__7_1_45_0>

R.A.I.R.O.
(7e année, avril 1973, R-1, p. 45-50)

BICOLORED DIGRAPH GRAMMAR SYSTEMS (1)

by Derick WOOD (2)

Abstract. — *A new class of generating systems (the bdg systems), is introduced and they are shown to be equivalent to the class of programmed grammars. They have independent interest since their definition gives closer ties between language theory and graph theory. This in turn gives rise to a number of interesting hybrid graph — grammar open problems.*

INTRODUCTION

There has been much interest in recent years in various extensions and modifications to the notion of grammar or generating system as studied by Chomsky (1959). There have been changes in the manner in which rules are applied, for example, matrix grammars and scattered context grammars (Salomaa, 1972) and changes in the rules themselves, for example, indexed grammars and programmed grammars (Salomaa, 1972). In this note we introduce a hybrid generating system based on bicoloured digraphs and phrase structure grammars, a *bicolored-digraph-grammar system* or *bdg system.*

BASIC NOTATION

A *phrase structure grammar* (PSG)G, is a quadruple (N, T, S, P) where N is a finite nonempty set of *nonterminals*, T is a finite nonempty set of *terminals*, S in N, is the *sentence* symbol and $P \subseteq V^* \times V^*$ is a finite set of *rules*, where $V = N \cup T$ and members of P are usually denoted $x \rightarrow y$, x, y in V^*.

(1) Supported under National Research Council of Canada, Grant No. A-7700.
(2) Department of Applied Mathematics, McMaster University, Hamilton, Ontario, Canada.

In the usual way if x, y in V^*, then $x \underset{G}{\Rightarrow} y$ is a *derivation* if $x = uwv$, $y = uzv$ and $w \to z$ in P, similarly $x \underset{G}{\Rightarrow}^+ y$ if there exists a sequence of derivations from x to give y. $x \underset{G}{\Rightarrow}^* y$ if $x = y$ or $x \underset{G}{\Rightarrow}^+ y$. If x in V^* and $w \to z$ in P then $w \to z$ is *applicable* to x if there exists u, v in V^* such that $x = uwv$. A rule $w \to z$ is a *w-rule*.

The *language* generated by a grammar G, denoted $L(G)$ is

$$\{ x : S \underset{G}{\Rightarrow}^* x, x \text{ in } T^* \} .$$

Further details of language therory can be found in Salomaa (1972). We now introduce the notion of a bicolored digraph.

A *bicolored digraph* Γ is a couple (U, E) where U is a finite nonempty set of *points* and $E \subseteq U \times \{ \underset{\sim}{r}, \underset{\sim}{g} \} \times U$ is a finite set of *directed edges*.

An edge connects two points together, is colored red or green and has a direction associated with it, indicated by the order of appearance of the points in the triple defining the edge. Thus $(u, \underset{\sim}{r}, v)$ is a red edge from u to v. As in a one-way street we can walk from u to v along the edge $(u, \underset{\sim}{r}, v)$ but not vice versa. Further details of digraph theory can be found in Harary, Norman, and Cartwright (1965).

BICOLORED-DIGRAPH-GRAMMAR SYSTEMS

We are now in a position to define bdg systems.

Definition.

A *bicolored-digraph-grammar system (bdg system)* θ is a quadruple (Γ, G, Σ, h) where $\Gamma = (U, E)$ is *a bicolored digraph*, $G = (N, T, S, P)$ is *a PSG*, $\Sigma \subseteq U$ is a nonempty set of *entry points* and $h : U \to (2^P - \phi)$ is a map, associating sets of rules of G with points in Γ. We require that each rule of G is associated with at least one point of Γ and that each S-rule is associated with at least one entry point.

A bdg system comprises a digraph, a grammar and a means of generation of words from V^*. The digraph Γ, is used as a means of regulating the application of the rules of G. We now define the yield operation.

Definition.

Given a bdg system $\theta = (\Gamma, G, \Sigma, h)$ and couples (x, u) and (y, v) in $V^* \times U$ we have the derivation $(x, u) \underset{\theta}{\Rightarrow} (y, v)$ if having chosen a rule $w \to z$ in $h(u)$ either $x = x_1 w x_2$, $y = x_1 z x_2$, this is the leftmost occurrence of w in x and

$(u, \underset{\sim}{g}, v)$ in E or $x = y$, $w \rightarrow z$ is not applicable to x and $(u, \underset{\sim}{r}, v)$ in E. We define $(x, u) \underset{\theta}{\Rightarrow}^{+} (y, v)$ iff there exists a sequence $(x_0, u_0), \ldots, (x_n, u_n)$ such that $x_0 = x$, $u_0 = u$, $x_n = y$, $u_n = v$ and $(x_i, u_i) \underset{\theta}{\Rightarrow} (x_{i+1}, u_{i+1})$, for all i, $0 \leqslant i < n$. Then $(x, u) \underset{\theta}{\Rightarrow}^{*} (y, v)$ iff $(x, u) \underset{\theta}{\Rightarrow}^{+} (y, v)$ or $(x, u) = (y, v)$. The subscript θ is dropped if there is no danger of confusion.

In a bdg system when the chosen rule can be applied (i.e. a derivation takes place in the underlying grammar), then a walk along a green edge (for success) is carried out to reach the next point on the digraph and hence, a new set of rules that can be applied. If, however, the chosen rule at the present point is not applicable then a walk along a red edge (for failure) is carried out. Note that a derivation can only be carried out if there is a red or green edge leading out of the present point.

Having defined the yield operation we can now associate a language with a bdg system.

Definition.

Given a bdg system $\theta = (\Gamma, G, \Sigma, h)$ then the language generated by θ is $L(\theta) = \{ x : x \text{ in } T^*, (S, u) \underset{\theta}{\Rightarrow}^{*} (x, v), u \text{ in } \Sigma \}$. A language L is a *bdg language* iff there exists a bdg system θ such that $L = L(\theta)$.

REMARKS

(*i*) Note that the definition of $L(\theta)$ implies the existence of at least one-S-rule in $h(u)$ for some u in Σ.

(*ii*) The restriction on the map h associated with a bdg system $\theta = (\Gamma, G, \Sigma, h)$ that each rule of G is associated with at least one point in U can be dropped without gaining any generative power. This can easily be seen, as such a rule can be removed from P without affecting $L(\theta)$. Therefore all rules that are not associated with points in U can be dropped from P, to give P' and G'. Then $\theta' = (\Gamma, G', \Sigma, h)$ is the new bdg system fulfilling the restriction.

(*iii*) The only initially useful points in Σ are those that have S-rules associated with them, all other rules associated with the points in Σ are initially useless.

We now show that each point need only be associated with one rule.

Definition.

A bdg system $\theta = (\Gamma, G, \Sigma, h)$ is a *canonical bdg system* if for all u in U, $h(u) = \{ w \rightarrow z \}$, for some $w \rightarrow z$ in P. Two bdg systems θ_1 and θ_2 are *equivalent*, $\theta_1 \equiv \theta_2$, iff $L(\theta_1) = L(\theta_2)$.

Theorem 1

Given a bdg system $\theta = (\Gamma, G, \Sigma, h)$ then there exists an equivalent canonical bdg system θ'.

Proof : By construction. Let $\theta' = (\Gamma', G, \Sigma', h')$ where

(*i*) $$\Gamma' = (U', E'),\ U' = \{ [u, w \to z] : w \to z \text{ in } h(u) \}$$

and $E' = \{ ([u, w \to z], C, [v, x \to y]) : [u, w \to z], [v, x \to y] \text{ in } U', \text{and } (u, C, v)$ in E where C in $\{ \underset{\sim}{r}, \underset{\sim}{g} \} \}$,

(*ii*) $$\Sigma' = \{ [u, S \to x] : u \text{ in } \Sigma \text{ and } S \to x \text{ in } h(u) \}$$

and

(*iii*) $$h' : U' \to P \text{ is defined by } h([u, w \to z]) = w \to z.$$

We need to show that θ' is in canonical form and that $\theta \equiv \theta'$. Trivially, by the definition of h', θ' is in canonical form, therefore it remains to show that $\theta \equiv \theta'$.

Claim 1. $L(\theta') \subseteq L(\theta)$.

Consider an arbitrary derivation $(S, u') \underset{\theta'}{\Rightarrow}^{+} (y, v')$, then there is a sequence $(x_0, u'_0), \ldots, (x_n, u'_n)$ such that $x_0 = S$,

$$u'_0 = u',\ x_n = y,\ u'_n = v' \text{ and } (x_i, u'_i) \underset{\theta'}{\Rightarrow} (x_{i+1}, u'_{i+1}),$$

for all i, $0 \leqslant i < n$. Letting $u'_i = [u_i, w_i \to z_i]$ then $h'(u'_i) = w_i \to z_i$, $0 \leqslant i \leqslant n$, then by the definition of a bdg system $(x_i, u'_i) \underset{\theta'}{\Rightarrow} (x_{i+1}, u'_{i+1})$ either by application of the rule $w_i \to z_i$ or $x_i = x_{i+1}$ and $w_i \to z_i$ is not applicable to x_i, for all i, $0 \leqslant i < n$. This sequence is mimicked in θ by the sequence $(x_0, u_0), \ldots, (x_n, u_n)$ since $w_i \to z_i$ can be chosen from $h(u_i)$, $0 \leqslant i < n$ and by the construction $u'_i = [u_i, w_i \to z_i]$. Further as u'_0 in Σ' then u_0 is in Σ. Therefore, as $(S, u') \underset{\theta'}{\Rightarrow}^{+} (y, v')$ gives rise to the corresponding derivation $(S, u) \underset{\theta}{\Rightarrow}^{+} (y, v)$, where $u = u_0$ and $v = u_n$, we have $L(\theta') \subseteq L(\theta)$.

Claim 2. $L(\theta) \subseteq L(\theta')$.

Again taking an arbitrary derivation $(S, u) \underset{\theta}{\Rightarrow}^{+} (y, v)$ a derivation in θ' can be constructed which will carry out the grammatical derivation $S \underset{G}{\overset{+}{\Rightarrow}} y$.

Therefore $\theta' \equiv \theta$.

Let us look at some examples of bdg systems.

EXAMPLE 1.

Denote an entry point by an entry arrow, all edges are colored $\underset{\sim}{g}$.

θ :

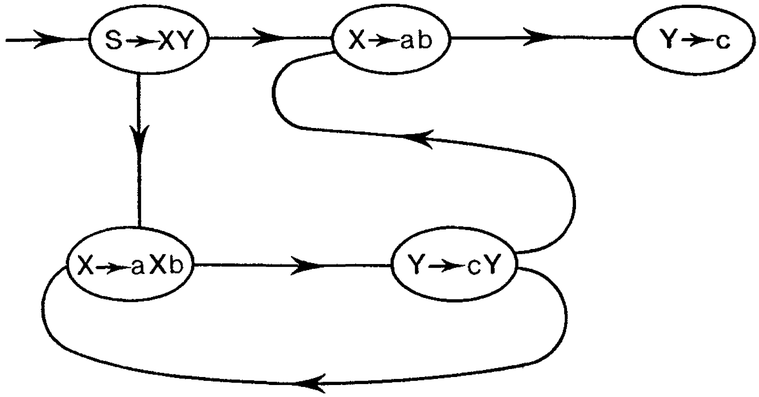

Figure 1

Then $L(\theta) = \{ ww : w \text{ in} \{ a, b \}^{+} \}$, which is context-sensitive whilst the underlying grammar G is almost right linear.

EXAMPLE 2.

θ :

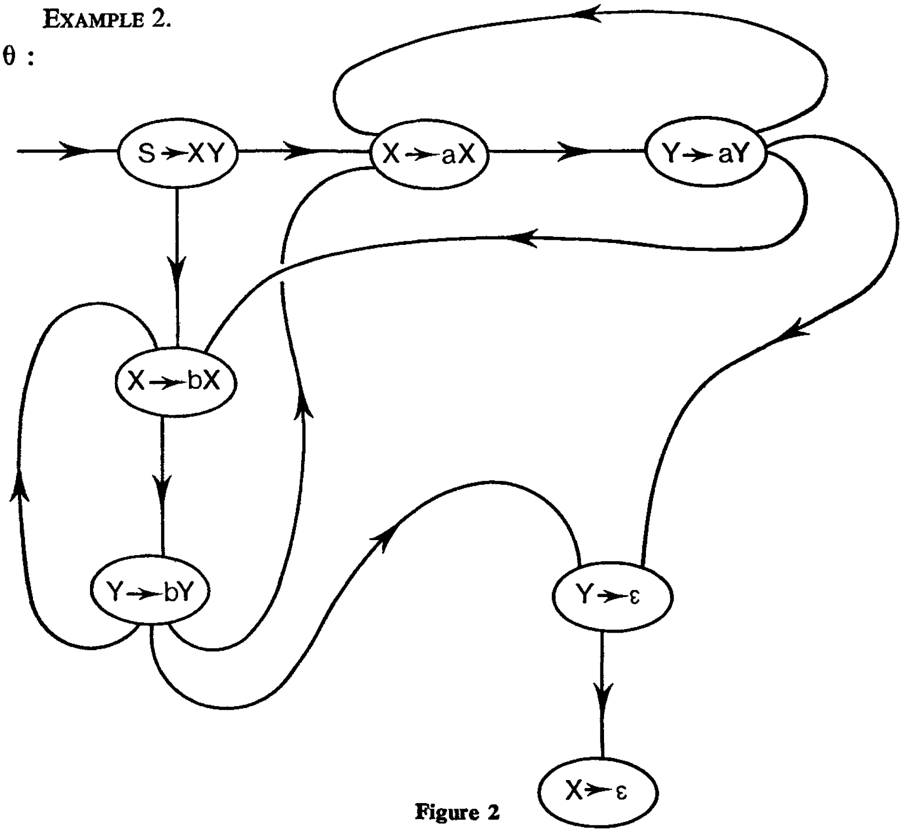

Figure 2

Then $L(\theta) = \{ a^n b^n c^n : n \geqslant 1 \}$, again a context sensitive language whilst the underlying grammar is context-free, in fact, with the introduction of one more nonterminal and two more points the underlying grammar can be made almost right linear.

PROGRAMMED GRAMMARS

We take the definition of a programmed grammar from Rosenkrantz (1969).

A *programmed grammar* (pg) G, is a quintuple (N, T, L, S, P) where N, T, and S are as in a *PSG*, L is a finite set of *labels* such that with each label l in L there is associated a unique rule $(l, w, z, S(l), F(l))$ and P is a finite set of *rules*, where a rule is written $l\ w \rightarrow z\ S(l)\ F(l)$. $w \rightarrow z$ is an ordinary phrase structure rule, l is the label, and $S(l), F(l) \subseteq L$ are the *success* and *failure* fields respectively.

In applying the rule $(l, w, z, S(l), F(l))$ to a word x in V^*, x is first scanned to see if it contains the subword w. If it does the leftmost appearance of w is replaced by z and the next rule to be applied to the resulting word is selected from $S(l)$. If x does not contain w, then no change is made and the next rule to be applied is selected from $F(l)$. If at any point in the derivation the next rule label must be chosen from the empty set, the derivation comes to a halt. The reader can either consult Rosenkrantz (1969) or develop for himself the necessary formalism for the yield relation and the language generated by a pg.

We now have the following.

Theorem 2

Every programmed grammar is a canonical bdg system and vice versa.

Proof: (*i*) Given a programmed grammar $G = (N, T, L, S, P)$ construct the canonical bdg system $\theta = (\Gamma, G', \Sigma, h)$ where $\Gamma = (U, E)$, $U = L$, and $E = \{ (l_1, C, l_2) : (l_1, w, z, S(l_1), F(l_1))$ in P and $C = \underset{\sim}{g}$ if l_2 in $S(l_1)$ and $C = \underset{\sim}{r}$ otherwise $\}$, $G' = (N, T, S, P')$ where $P' = \{ w \rightarrow z : (l, w, z, S(l), F(l))$ in $P \}$, $\Sigma = \{ l : (l, S, z, S(l), F(l))$ in $P \}$ and h is defined by: for all l in L, $h(l) = \{ w \rightarrow z \}$, where $(l, w, z, S(l), F(l))$ in P.

It should be clear that $L(G) = L(\theta)$, as green corresponds to success and red corresponds to failure.

(*ii*) Similarly given a canonical bdg a notational construction can be carried out to given an ' equivalent ' programmed grammar.

This gives :

Corollary 3

A language is a bdg language iff it is a programmed language.

Rosenkrantz (1969) also introduces the notion of an *unconditional transfer pg*, a pg in which the success and failure fields of each rule are the same. A little thought shows that this is identical to the notion of a *canonical digraph-grammar system*, a bdg system in which an edge can be used as an exit from a point whether the chosen rule is or is not applicable (*i.e.* is independent of its color). Thus we obtain :

Theorem 4

A language is a dg language iff it is an utp language.

CONCLUDING REMARKS

bdg systems have been shown to be programmed grammars in disguise. However, bdg systems do provide a new way to examine programmed grammars, as they emphasize the hybrid nature of these animals, displaying as they do the graph theoretic foundation for their definition. This enables us to state some open problems which are a result of looking at bdg systems from this standpoint.

Problem 1

In the definition of a derivation in a bdg system, the notion of *a priori*, choosing one rule from a set of rules associated with a particular point before attempting to apply it, is the fundamental reason that Theorem 1 is true. Let $\underset{\theta}{\Rightarrow}$ be denoted by $\underset{\theta}{\Rightarrow}^{ch}$ (*ch* for chosen) and define a new yield operation $\underset{\theta}{\Rightarrow}$ by :

$$(x, u) \underset{\theta}{\Rightarrow} (y, v)$$

if either $x = x_1 w x_2$, $y = x_1 z x_2$, $w \rightarrow z$ is in $h(u)$, this is the leftmost occurrence of w in x and $(u, \underset{\sim}{g}, v)$ in E or $x = y$, no rule in $h(u)$ is applicable to x and $(u, \underset{\sim}{r}, v)$ in E.

Letting

$$L(\theta) = \{ x : x \text{ in } T\ , (S, u) \underset{\theta}{\Rightarrow}^{*} (x, v), u \text{ in } \Sigma \}$$

and

$$L_{ch}(\theta) = \{ x : x \text{ in } T^{*}, (S, u) \underset{\theta}{\Rightarrow}^{ch*} (x, v), u \text{ in } \Sigma \},$$

is it true that $\{ L : L(\theta)$, for some bdg system $\theta \} = \{ L : L = L_{ch}(\theta)$, for some bdg system $\theta \}$?

Problem 2

Given a class of underlying grammars $\mathcal{G}$, say the class of context-free grammars, is there a bdg language L which is inherently non-planar with respect to $\mathcal{G}$? i.e. For all $\theta = (\Gamma, G, \Sigma, h)$ such that G in $\mathcal{G}$ and $L = L(\theta)$ then θ is a non-planar graph.

In another paper, in preparation, further graph-grammar systems are investigated.

REFERENCES

CHOMSKY N., *On certain formal properties of grammars*, IC 2, 1959, 2, pp. 137-167.

HARARY F., NORMAN R. Z., and CARTWRIGHT D., *Structural models*, 1965, John Wiley and Son.

ROSENKRANTZ D. J., *Programmed grammars and classes of formal languages*, JACM 16, 1969, 1, pp. 107-131.

SALOMAA A., *Formal languages*, to be published by Academic Press, 1973.