ELSEVIER

Discrete Mathematics 182 (1998) 29–51

DISCRETE MATHEMATICS

The obstructions of a minor-closed set of graphs defined by a context-free grammar[1]

B. Courcelle, G. Sénizergues

LaBRI, CNRS, URA 1304, Bordeaux I University, 351 Cours de la Libération, F33405 Talence, France

Received 18 August 1995; received in revised form 17 January 1997; accepted 15 May 1997

Abstract

We establish that the finite set of obstructions of a minor-closed set of graphs given by a hyperedge replacement grammar can be effectively constructed. Our proof uses an auxiliary result stating that the system of equations associated with a proper hyperedge replacement grammar has a unique solution.

1. Introduction

Several finite devices can be used to specify sets of finite graphs. In this paper we are interested in the following ones:

— forbidden minors.

— Hyperedge Replacement graph grammars.

— monadic second-order formulas.

The paradigm of a specification by forbidden minors is Kuratowski's theorem saying that a graph is planar iff it does not contain $K_{3,3}$ nor K_5 as a minor. The graphs $K_{3,3}$ and K_5 are called the *obstructions* of the class of planar graphs.

Graph *grammars* specify graphs in a generative way, by rules building graphs from smaller ones. Monadic second-order *formulas* can be used to express characteristic properties like connectivity or the existence of cycles. In certain cases, these three types of definitions are equivalent. The following theorem is the basis of the present paper (the definitions will be given later).

Theorem (Courcelle [6]). *Let L be a minor-closed set of graphs of bounded tree-width. Then the following are true:*

[1]This work has been supported by the ESPRIT Basic Research Working Group COMPUGRAPH II.
E-mail: courcell,ges@labri.u-bordeaux.fr.

PII S0012-365X(95)00132-5

1. *L has a finite set of obstructions;*
2. *L is definable by a Monadic Second-order (MS) formula (i.e. there exists a MS formula φ, such that L is the set of all graphs satisfying φ),*
3. *L is generated by a Hyperedge Replacement graph (HR) grammar.*

Our main result states that one can *compute* monadic second-order formula and the set of obstructions from a HR grammar (Theorem 5.10). The computability of a grammar from the obstructions or a MS formula was proved elsewhere [3, 5, 6].

Some comments concerning this theorem are in order. Since every set of graphs defined by a HR grammar has bounded tree-width, the hypothesis that the graphs of the considered set have bounded tree-width is necessary. Otherwise, a minor-closed set (of unbounded tree-width) always has a finite set of obstructions and is always definable by a MS formula. However, the MS formula can be constructed from the obstructions but not vice versa. Hence, the restriction to sets of graphs of bounded tree-width is essential in our results.

We can sketch the main idea of the proof, which makes an essential use of the results of Robertson and Seymour [11].

Let Γ be a HR grammar defining a minor-closed set of graphs L. One can transform it into a system of equations Σ in sets of graphs such that

— this system has a unique solution.
— L is one component of the unique solution.
— each component of the unique solution is minor-closed, hence can be specified by a finite set of obstructions, whence by a MS formula.

For any sequence of MS formulas, say $(\varphi_1, \dots, \varphi_m)$, (where m is the number of equations of Σ), one can verify whether the m-tuple of graphs they specify is a solution of Σ (i.e., is *the* solution of Σ). By enumerating blindly m-tuples of MS formulae and testing for each of them whether it defines the solution of Σ, one is sure to find one. From the MS formulae one can then construct the obstructions.

This procedure is of course highly intractable. Unfortunately, we have no hope to make it tractable and to obtain, for instance, the still unknown set of obstructions of graphs of tree-width at most 4. (Sanders knows about 70 obstructions but is not sure that the list is complete [13].)

The paper is organized as follows: Section 2 contains only definitions and can be skipped by the reader knowing graph grammars; Section 3 deals with proper HR grammars in the general case where they generate hypergraphs; Section 4 gives a construction of the minor closure of a HR set of graphs; Section 5 is devoted to the main result; Section 6 presents our conclusion.

2. Hypergraphs and HR hypergraph grammars

We review the basic definitions from [1]. As in this paper, we deal with a certain class of oriented hypergraphs. The reader knowing [3] may skip this section.

2.1. Hypergraphs

The *hypergraphs* we define have labelled hyperedges. The alphabet of hyperedge labels is a *ranked alphabet* A, i.e., an alphabet that is given with a mapping $\tau : A \to \mathbb{N}$ (the integer $\tau(a)$ is called the *type* of a). A *hypergraph over* A *of type* n is a 5-tuple $H = \langle V_H, E_H, \mathrm{lab}_H, \mathrm{vert}_H, \mathrm{src}_H \rangle$ where V_H is the finite set of vertices, E_H is the finite set of hyperedges, lab_H is a mapping $E_H \to A$ defining the *label* of a hyperedge, vert_H is a mapping $E_H \to V_H^*$, defining the (possibly empty) *sequence of vertices* of a hyperedge, and src_H is a sequence of vertices of length n. We impose the condition that the length of $\mathrm{vert}_H(e)$ is equal to $\tau(\mathrm{lab}_H(e))$ for all e in E_H. One may also have labels of type 0, labelling hyperedges with no vertex. An element of src_H is called *a source* of H. The sets E_H and V_H are assumed to be finite and disjoint. We consider two isomorphic hypergraphs as equal.

We denote by $G(A)$ the set of all isomorphism classes of hypergraphs over A, by $G(A)_n$ the subset of those of type n. A hypergraph of type n is also called an *n-hypergraph.*

A *graph* is a hypergraph all hyperedges of which are of type 2. A graph is thus directed (unless otherwise specified). We say that H has pairwise distinct sources if no vertex occurs twice in the sequence src_H.

We now define the substitution of a hypergraph for a hyperedge in a hypergraph.

2.2. Substitutions

Let $G \in G(A)$, let $e \in E_G$; let $H \in G(A)$ be a hypergraph of type $\tau(e)$. We denote by $G[H/e]$ the result of the *substitution of* H for e in G. This hypergraph can be constructed as follows:

(1) construct a hypergraph G' by deleting e from G (but keep the vertices of e);
(2) add to G' an isomorphic copy $\bar{H}$ of H, disjoint from G';
(3) fuse the vertex $\mathrm{vert}_G(e, i)$, i.e., the ith element of the sequence $\mathrm{vert}_G(e)$ (that is still a vertex of G'), with the ith source of $\bar{H}$; this is done for all $i = 1, \ldots, \tau(e)$;
(4) the sequence of sources of $G[H/e]$ is the image of the sequence of sources of G' under the identifications induced by step (3).

If $e_1, \ldots, e_l$ are pairwise distinct hyperedges of G, if $H_1, \ldots, H_l$ are hypergraphs of respective types $\tau(e_1), \ldots, \tau(e_l)$, then the substitutions in G of H_1 for $e_1, \ldots, H_l$ for e_l can be done in any order; the result is the same, and it is denoted by $G[H_1/e_1, \ldots, H_l/e_l]$.

2.3. Hypergraph operations

Let $U = \{u_1, u_2, \ldots, u_m\}$ be another finite-ranked alphabet, disjoint from A. We call U the alphabet of *unknowns*. U is given with a fixed total ordering of its elements: $u_1 < u_2 < \cdots < u_m$.

Let us call *hypergraph-operator* of type n any $(l+1)$-tuple $P=(G,e_1,e_2,\dots,e_l)$ where $G\in \boldsymbol{G}(A\cup U)_n$ and $e_1,e_2,\dots,e_l$ is an enumeration without repetition of all the U-labelled hyperedges of G. Let us suppose $label(e_j)=u_{i_j}$ and let $n_i=\tau(u_i)$.

The *hypergraph-operation*

$$\bar{P}:\boldsymbol{G}(A)_{n_{i_1}}\times\cdots\times\boldsymbol{G}(A)_{n_{i_l}}\to\boldsymbol{G}(A)_n$$

is defined by

$$\bar{P}[H_1,H_2,\dots,H_l]=G[H_1/e_1,H_2/e_2,\dots,H_l/e_l].$$

The graph-operator P will be said standard iff, $\forall j,k\in[1,l]$, $j\leqslant k\Rightarrow label(e_j)\leqslant label(e_k)$. One can check that, for every $(H_1,H_2,\dots,H_m)\in\boldsymbol{G}(A)_{n_1}\times\boldsymbol{G}(A)_{n_2}\times\cdots\times\boldsymbol{G}(A)_{n_m}$ the graph $\bar{P}[H_{i_1},H_{i_2},\dots,H_{i_l}]$ is independent of the chosen enumeration $e_1,\dots,e_l$ provided P is associated to the same graph G and is standard.

Therefore, we can define an operation

$$\bar{G}:\boldsymbol{G}(A)_{n_1}\times\cdots\times\boldsymbol{G}(A)_{n_m}\to\boldsymbol{G}(A)_n$$

by

$$\bar{G}[H_1,H_2,\dots,H_m]=\bar{P}[H_{i_1},H_{i_2},\dots,H_{i_l}],$$

where $P=(G,e_1,\dots,e_l)$ is standard.

We extend the operation $\bar{P}$ to *sets* of hypergraphs by letting

$$\hat{P}:\mathscr{P}(\boldsymbol{G}(A)_{n_{i_1}})\times\cdots\times\mathscr{P}(\boldsymbol{G}(A)_{n_{i_l}})\to\mathscr{P}(\boldsymbol{G}(A)_n)$$

be defined by

$$\hat{P}[L_1,L_2,\dots,L_l]=\{\bar{P}[H_1,H_2,\dots,H_l]\mid\forall j\in[1,l],H_j\in L_j\}.$$

Here also, the set $\hat{P}[L_{i_1},L_{i_2},\dots,L_{i_l}]$ is independent of the chosen enumeration $e_1,\dots,e_l$ provided P is associated to the same graph G and is standard. Therefore, we can define an operation

$$\hat{G}:\mathscr{P}(\boldsymbol{G}(A)_{n_1})\times\cdots\times\mathscr{P}(\boldsymbol{G}(A)_{n_m})\to\mathscr{P}(\boldsymbol{G}(A)_n)$$

by

$$\hat{G}[L_1,L_2,\dots,L_m]=\hat{P}[L_{i_1},L_{i_2},\dots,L_{i_l}],$$

where $P=(G,e_1,\dots,e_l)$ is standard.

2.4. HR grammars

A *Hyperedge Replacement* grammar is a 4-tuple $\Gamma=(A,U,Q,Z)$ where A is the finite *terminal* ranked alphabet, U is the finite *nonterminal*-ranked alphabet, Q is the finite set of *production rules*, i.e., is a finite set of pairs of the form $(u,D)\in U\times\boldsymbol{G}(A\cup U)_{\tau(u)}$ and $Z\in U$ is a non-terminal symbol called the *axiom*. A production rule (u,D) will frequently be written $u\to D$; we shall denote by $Q(u)$ the set $\{D\mid(u,D)\in Q\}$.

The *one-step derivation* relation, $\underset{Q}{\rightarrow}$ is defined by

$K \underset{Q}{\rightarrow} H$ iff there exists a hyperedge e in K, the label of which is some u in U, and a production rule (u, D) in Q, such that $H = K[D/e]$.

The *derivation* relation $\overset{*}{\underset{Q}{\rightarrow}}$ is the reflexive and transitive closure of $\underset{Q}{\rightarrow}$.
For every hypergraph $K \in \boldsymbol{G}(A \cup U)_n$, we define

$$\boldsymbol{L}(\Gamma, K) := \{H \in \boldsymbol{G}(A)_n \mid K \overset{*}{\underset{Q}{\rightarrow}} H\}.$$

For every nonterminal symbol $u \in U$ we denote by $\tilde{u}$ the unique hypergraph such that: $\tilde{u}$ has $\tau(u)$ distinct sources, exactly one hyperedge e, this e is labelled by u, $\tilde{u}$ has no internal vertex and the ith vertex of e is the ith source of $\tilde{u}$. We shall sometimes (abusively) use the symbol u in place of $\tilde{u}$ in the sequel, when no confusion is possible.

The set of hypergraphs *generated* by Γ is defined by

$$\boldsymbol{L}(\Gamma) := \boldsymbol{L}(\Gamma, \tilde{Z}).$$

A set of graphs is HR iff it is generated by some HR grammar Γ. We shall simply say a grammar in the sequel.

2.5. *Systems of equations in sets of hypergraphs*

Let $\Gamma = (A, U, Q, Z)$ be a grammar. Let us suppose that

$$U = \{u_1, \dots, u_m\} \quad \text{and} \quad Q(u_i) = \{D_{i,1}, \dots, D_{i,s_i}\} \text{ (for } 1 \leqslant i \leqslant m).$$

The *system of equations* S_Γ associated with Γ is then the set of equations

$$u_i = \sum_{j=1}^{s_i} D_{i,j} \quad (\text{for } 1 \leqslant i \leqslant m).$$

A m-tuple $(L_1, L_2, \dots, L_m) \in \mathscr{P}(\boldsymbol{G}(A)_{n_1}) \times \cdots \times \mathscr{P}(\boldsymbol{G}(A)_{n_m})$ is then a *solution* of S_Γ iff, for all i in $[1, m]$ we have:

$$L_i = \bigcup_{j=1}^{s_i} \widehat{D_{i,j}}(L_1, L_2, \dots, L_m).$$

We denote the formal sum $\sum_{j=1}^{s_i} D_{i,j}$ by t_i and the operation $\sum_{j=1}^{s_i} \widehat{D_{i,j}}$ by $\hat{t}_i$.

Theorem 2.1 (Bauderon, Courcelle [1]). *$(\boldsymbol{L}(\Gamma, u_1), \dots, \boldsymbol{L}(\Gamma, u_m))$ is the least solution of S_Γ, for componentwise set inclusion.*

Example 2.2. We let $\Gamma = (\{a\}, \{u\}, Q, u)$, where a and u have rank 2 and $Q(u)$ is shown in Fig. 1.

$\boldsymbol{L}(\Gamma, u)$ is the set of edge series-parallel multidigraphs in the sense of [14]. A typical example of such a graph is shown in Fig. 2.

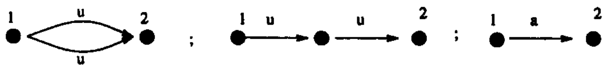

Fig. 1.

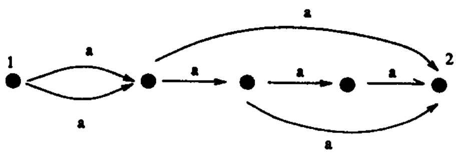

Fig. 2.

3. Proper hyperedge replacement grammars

We denote by I_G the set of internal vertices of a hypergraph G i.e., of those that are not sources. For every hypergraph G, we let $\|G\|$ be the integer $\mathrm{Card}(I_G) + \mathrm{Card}(E_G)$ and we call it the *size* of G. We say that a graph is *empty* iff $\|G\| = 0$.

There is only one empty 0-hypergraph. An empty n-hypergraph, for $n \geqslant 1$, has vertices, namely, the sources (possibly all identical).

For every hypergraph G, we let $\varepsilon(G)$ denote the empty hypergraph obtained by deleting the hyperedges and the internal vertices of G.

Lemma 3.1. *For all hypergraphs $H, G_1, \ldots, G_l$, if $e_1, \ldots, e_l$ are hyperedges of H of respective types $\tau(G_1), \ldots, \tau(G_l)$ we have*

$$\|H[G_1/e_1, \ldots, G_l/e_l]\| = \|H[\varepsilon(G_1)/e_1, \ldots, \varepsilon(G_l)/e_l]\| + \|G_1\| + \cdots + \|G_l\|.$$

Proof. Easy verification from the definitions. $\square$

A *unit hypergraph* is a graph with exactly one U-labelled hyperedge, no terminal hyperedge and no isolated internal vertex. (A vertex is *isolated* if it is not in the vertex sequence of any hyperedge). A grammar Γ is *proper* if no right-hand side of a rule is unit or empty.

Lemma 3.2. *A proper grammar generates nonempty hypergraphs.*

Proof. Let Γ be a proper grammar. Assume we have a derivation $u \xrightarrow[\Gamma]{*} G$ with G empty. Let $u' \to D$ be a rule with terminal right-hand side used in this derivation. If D has a hyperedge or an isolated internal vertex, then G has also a hyperedge or an internal vertex. Hence, D should be empty. But Γ has no rule with empty right-hand side. Contradiction. $\square$

Theorem 3.3. *Let Γ be a proper grammar with nonterminals $u_1, \dots, u_m$. The m-tuple*

$$(L(\Gamma, u_1), \dots, L(\Gamma, u_m))$$

is the unique m-tuple of sets of nonempty hypergraphs that is a solution of S_Γ.

Proof. Let $L_i = L(\Gamma, u_i)$ for $i = 1, \dots, m$. Hence $(L_i)_{1 \leqslant i \leqslant m}$ is the least solution of S_Γ in sets of hypergraphs. The sets L_i contain no empty hypergraph (Lemma 3.2). Hence, $(L_i)_{1 \leqslant i \leqslant m}$ is the least solution of S_Γ in sets of nonempty hypergraphs.

Assume $(M_i)_{1 \leqslant i \leqslant m}$ is any solution in sets of nonempty hypergraphs. We have $L_i \subseteq M_i$ for all i. If we do not have the equalities, we let G be a hypergraph of minimal size in $\bigcup\{M_j - L_j / 1 \leqslant j \leqslant m\}$, and $G \in M_i - L_i$.

Since $(M_j)_{1 \leqslant j \leqslant m}$ is a solution of S_Γ in nonempty hypergraphs, we have

$$G = D[G_1/e_1, \dots, G_l/e_l]$$

for some rule of Γ

$$u \to D[e_1, \dots, e_l]$$

with label$(e_j) = u_{i_j}$ and $G_j \in M_{i_j}$, for all $j \in [1, l]$. By Lemma 3.1 we have

$$\|G\| = \|D[\varepsilon(G_1), \dots, \varepsilon(G_l)]\| + \|G_1\| + \cdots + \|G_l\|.$$

Case 1: $\|G_j\| < \|G\|$ *for all* j. By minimality of $\|G\|$, we have $G_j \in L_{i_j}$ for each j hence $G \in L_i$ but this contradicts the choice of G.

Case 2: $\|G_j\| = \|G\|$ *for some* j. Since the hypergraphs $G_1, \dots, G_l$ are all nonempty, we must have $l = 1$ and $\|D[\varepsilon(G_1)/e_1]\| = 0$. Hence, D has no terminal hyperedge and no internal isolated vertex. Hence it is unit. But this means that Γ contains the rule $u_i \to D$ which has a unit right-hand side. Hence Γ is not proper.

In both cases we get a contradiction. It follows that $M_j - L_j = \emptyset$ for every j hence that

$$(M_1, \dots, M_m) = (L_1, \dots, L_m),$$

as was to be shown. □

Theorem 3.4. *For every grammar Γ one can construct a proper grammar Γ' with the same set of nonterminals such that, for every $u \in U$*

$$L(\Gamma', u) = \{G \in L(\Gamma, u) \mid G \textit{ is nonempty}\}.$$

A very similar theorem is proved in [9, p. 77, Corollary 1.10].

Proof. For every $u \in U$ we let

$$\mathrm{EMPTY}(u) = \{G \in L(\Gamma, u) \mid G \text{ is empty}\},$$

$$\mathrm{UNIT}(u) = \{G \mid u \xrightarrow[\Gamma]{*} G, G \text{ is a unit graph}\}.$$

These sets are finite and can be computed because the membership problem for an HR set of graphs is decidable [9, p. 75, Corollary 1.6.]

We first construct a grammar Γ'' consisting of all rules of Γ together with the following ones:

$$\text{all rules} \quad u \to H[H'[E_1/e_1, \dots, E_l/e_l]/u'], \tag{1}$$

where

— $H[u'] \in \mathrm{UNIT}(u)$, $u' \to H'$ is a rule of Γ,
— $0 \leqslant l \leqslant$ number of nonterminal hyperedges of H',
— $e_1, \dots, e_l$ are nonterminal hyperedges of H' with respective labels $w_1, \dots, w_l$, $E_i \in \mathrm{EMPTY}(w_i)$, for all $i \in [1, l]$,
— $H[H'[E_1/e_1, \dots, E_l/e_l]/u']$ is neither empty nor unit.

Remark. (1) In a rule of the form (1) fulfilling the above conditions it may happen that

— $H[u'] = u, u' = u$ (because $u \in \mathrm{UNIT}(u)$)
— $l = 0$, i.e., the rule is of the form $u \to H[H'/u']$.

(2) It may happen that H' is empty, $l = 0$, but $H[H'/u']$ is not empty, as in the example shown in Fig. 3.

(3) It may happen that H' is unit, $l = 0$, but $H[H'/u']$ is not unit, see Fig. 4.

For each new rule $u \to G$, we have $u \xrightarrow[\Gamma]{+} G$. It follows that

$$L(\Gamma, u) = L(\Gamma'', u) \quad \text{for all } u \in U.$$

We now let Γ' be obtained from Γ'' by the deletion of all rules with an empty or unit right-hand side. Hence

$$L(\Gamma', u) \subseteq L(\Gamma'', u) \quad \text{for all } u \in U.$$

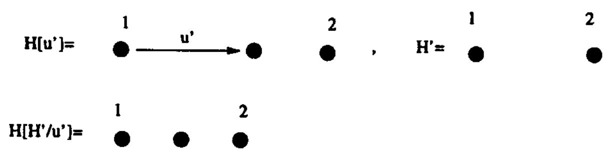

Fig. 3.

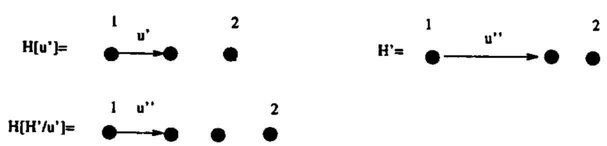

Fig. 4.

Since Γ' is proper $\boldsymbol{L}(\Gamma', u)$ is contained in the set of nonempty hypergraphs of $\boldsymbol{L}(\Gamma'', u)$. Let conversely $G \in \boldsymbol{L}(\Gamma'', u)$ be nonempty. Let

$$d: u \to G_1 \to G_2 \cdots \to G_p = G$$

be a derivation of G in Γ''.

If no rule used in this derivation has an empty or unit right-hand side, then $G \in \boldsymbol{L}(\Gamma', u)$. Otherwise we transform d into another derivation d' of G in Γ'' which does not use rules with empty or unit right-hand sides. We do the proof by induction on p (simultaneously for all $u \in U$).

Basis: $p = 1$. Then $u \to G_1$ is a rule of Γ'', $G_1 = G$ is nonempty and we take $d' = d$.

Inductive step:

Case 1: *$G_1, \dots, G_{p-1}$ are all unit graphs.* In this case $u \to G$ is a rule of Γ'', we let $d': u \to G$.

Case 2: *G_i (for some $1 < i \leqslant p - 1$) is the first nonunit hypergraph in the sequence $G_1, \dots, G_i$.* Then $G_{i-1} \in \mathrm{UNIT}(u)$ and $u \to G_i$ is a rule of Γ''. We take

$$d': u \to G_i \to G_{i+1} \to \cdots \to G_p = G.$$

(Note that, by point 3 of the above remark, we may have $G_{i-1} = H[u']$ and $G_i = H[H'[u'']/u']$ nonunit for some rule $u' \to H'[u'']$ with unit right-hand side).

Case 3: *G_1 is nonunit.* Then it is nonempty. We have $G_1 = H[e_1, \dots, e_l]$ and $G = H[G'_1/e_1, \dots, G'_l/e_l]$ where $\mathrm{label}(e_j) = u_{i_j}$ and $u_{i_j} \xrightarrow{*} G'_j$ by some derivations d_j of length at most $p - 1$. Let us assume that $G'_1, \dots, G'_{l_0}$ are empty and the others are not. By induction, we have derivations

$$d'_j: u_{i_j} \xrightarrow{*} G'_j$$

without unit or empty rules for $j = l_0 + 1, \dots, l$. We have also in Γ'' a rule of the form $u \to H'$ where

$$H' = H[G'_1/e_1, \dots, G'_{l_0}/e_{l_0}].$$

Hence, we can take for d' the derivation starting with $u \to H'$ and continued by the derivations $d'_{l_0+1}, \dots, d'_l$ so as to generate $H'[G'_{l_0+1}/e_{l_0+1}, \dots, G'_l/e_l] = G$. $\square$

Let us illustrate this construction by the following.

Example 3.5. $\Gamma = \langle A, U, Q, Z \rangle$ where $A = \{a\}$, $U = \{u, v\}$, $(\tau(u) = \tau(v) = \tau(a) = 2)$, $Z = u$ and $Q(u)$, $Q(v)$ are shown in Fig. 5

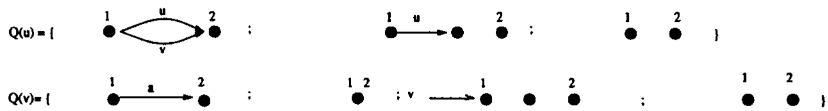

Fig. 5.

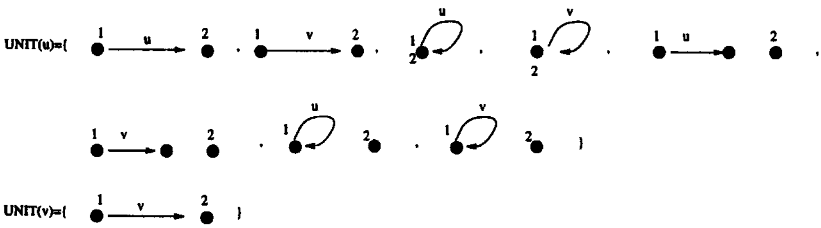

Fig. 6.

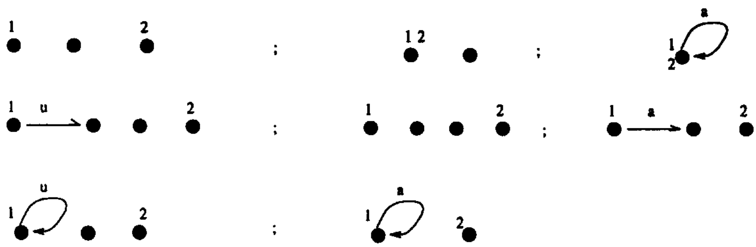

Fig. 7.

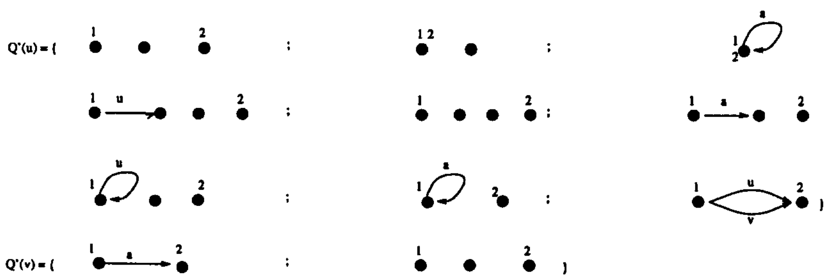

Fig. 8.

We have
EMPTY(u) = EMPTY(v) = $\{\overset{1}{\bullet}\ \overset{2}{\bullet},\ \overset{1\,2}{\bullet}\}$.
The graphs in UNIT(u), UNIT(v) are shown in Fig. 6.
Then $\Gamma'' = \langle A, U, Q'', Z\rangle$ where Q'' consists of the union of Q with the set of rules Q''_1: $Q''_1(u)$ is shown in Fig. 7 and $Q''_1(v) = \emptyset$.
Finally, Γ' is shown in Fig. 8.

4. The minor closure of a HR set of graphs

In this section we let A consist of one symbol of type 2. The elements of $\boldsymbol{G}(A)$ are thus *directed* graphs (possibly with loops and multiple edges) and sources (we abbreviate

$\boldsymbol{G}(A)$ as $\boldsymbol{G}$, $\boldsymbol{G}(A)_n$ as $\boldsymbol{G}_n$). We shall use grammars, with nonterminals of arbitrary types, to generate subsets of $\boldsymbol{G}$.

For simplicity, we give all definitions for directed graphs. The extension to undirected graphs is straightforward (see the end of the paper).

Definition (*Minor inclusion*). Let G, $H \in \boldsymbol{G}$. We write $G \trianglelefteq H$ and we say that G *is a minor* of H (or *is included* in H *as a minor*) iff

1. G and H are of same type
2. G is obtained from a subgraph G' of H by edge contractions.

Since isomorphic graphs are considered as equal, minor inclusion is a partial order (since $G \trianglelefteq H$ and $H \trianglelefteq G$ implies $G = H$). G is a *proper minor* of H if $G \trianglelefteq H$ and $G \neq H$.

Since $G \trianglelefteq H$ implies that G and H are of same type, the sources of H cannot be deleted. By edge contractions several distinct sources can get fused. In the example shown on Fig. 9, $G \trianglelefteq G' \subseteq H$.

For every set of graphs L, we let

$$\trianglelefteq(L) = \{G/G \trianglelefteq H, \text{ for some } H \in L\}$$

and we call it the *minor closure* of L. A set L is *minor closed* iff $L = \trianglelefteq(L)$. We write $G \trianglelefteq' H$ iff G and H are of same type and G can be obtained from H by edge contractions and edge deletions.

It follows that $G \trianglelefteq' H$ does not hold if, for example

$$G = \overset{1}{\bullet}\ \overset{2}{\bullet}\ \bullet\ \bullet$$

and

$$H = \overset{1}{\bullet}\ \overset{2}{\bullet}\ \bullet\ \bullet\ \bullet$$

whereas of course $G \trianglelefteq H$. For every set of graphs L we let

$$\trianglelefteq'(L) = \{G/G \trianglelefteq' H, \text{ for some } H \in L\}$$

Proposition 4.1. *For every grammar Γ, one can construct a grammar Γ' with same set U of nonterminals such that $\boldsymbol{L}(\Gamma', u) = \trianglelefteq'(\boldsymbol{L}(\Gamma, u))$ for every $u \in U$.*

We need first a lemma

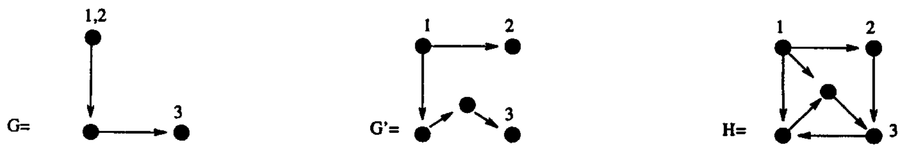

Fig. 9.

Lemma 4.2. *Let $H \in \mathbf{G}(A \cup U)$ with nonterminal hyperedges $e_1, \dots, e_l$; let $G_1, \dots, G_l \in \mathbf{G}$ of respective types $\tau(e_1), \dots, \tau(e_l)$. Then $G \trianglelefteq' H[G_1/e_1, \dots, G_l/e_l]$ iff there exist $H' \in \mathbf{G}(A \cup U)$ and $G'_1, \dots, G'_l \in \mathbf{G}$ such that*
1. *$G'_i \trianglelefteq' G_i$ for every $i = 1, \dots, l$,*
2. *H' is obtained from H by contractions and deletions of terminal edges,*
3. *$G = H'[G'_1/e_1, \dots, G'_l/e_l]$.*

Proof. Straightforward. □

Proof of Proposition 4.1. For every rule $u \to D$ of Γ, for every D' obtained from D by deletions and contractions of terminal edges, we put $u \to D'$ as a rule of Γ'. With the help of Lemma 4.2 and by induction on the length of derivation sequences, one can prove that:
1. if $u \xrightarrow[\Gamma]{*} G$, G is terminal and $G'' \trianglelefteq' G$ then $u \xrightarrow[\Gamma']{*} G'$,
2. if $u \xrightarrow[\Gamma']{*} G'$ then there exists $G \in \mathbf{L}(\Gamma, u)$ such that $G' \trianglelefteq' G$.

Hence, $\mathbf{L}(\Gamma', u) = \trianglelefteq'(\mathbf{L}(\Gamma, u))$. We omit details. □

For every set of graphs L, we define

$c(L) = \{G \mid G$ is obtained from a graph in L by deletion of internal isolated vertices$\}$.

For every $L \subseteq \mathbf{G}$ we have $L \subseteq c(L)$ and $\trianglelefteq(L) = c(\trianglelefteq'(L))$.

Proposition 4.3. *For every grammar Γ with set of nonterminals U, one can construct a grammar Γ' with set of nonterminals $U' \supseteq U$ such that:*
1. *for every $u \in U, \mathbf{L}(\Gamma', u) = c(\mathbf{L}(\Gamma, u))$ (hence $\mathbf{L}(\Gamma', u) = c(\mathbf{L}(\Gamma', u))$)*
2. *for every $u \in U' - U$, $\mathbf{L}(\Gamma', u) = c(\mathbf{L}(\Gamma', u))$*
3. *if $\mathbf{L}(\Gamma, u) = \trianglelefteq'(\mathbf{L}(\Gamma, u))$ for every $u \in U$ then $\mathbf{L}(\Gamma', u) = \trianglelefteq(\mathbf{L}(\Gamma', u))$ for every $u \in U' - U$ (hence also for every $u \in U'$).*

Proof (*sketch*). Given Γ, one first constructs a grammar Γ_1 consisting of all rules $u \to D_1$ where $D_1 \in c(D)$ for some rule $u \to D$ of Γ. Clearly, $\mathbf{L}(\Gamma_1, w) \subseteq c(\mathbf{L}(\Gamma, w))$ for all nonterminals w. We would like to have the equality but this is not always the case as shown by the following example of Γ with one nonterminal u of type 2. The right-hand sides of the rules of Γ are given in Fig. 10.

Fig. 10.

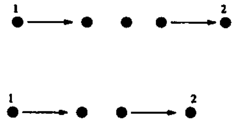

Fig. 11.

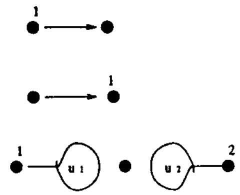

Fig. 12.

No right-hand side of a rule has an internal isolated vertex, hence $\Gamma_1 = \Gamma$. However, the graph G_1 (first line of Fig. 11) belongs to $\boldsymbol{L}(\Gamma, u)$, but the graph G_2 (second line of Fig. 11) belongs to $c(\boldsymbol{L}(\Gamma, u)) - \boldsymbol{L}(\Gamma, u)$.

The above construction is then insufficient. It works for grammars Γ satisfying the following additional condition:

> for every nonterminal u, for every $G \in \boldsymbol{L}(\Gamma, u)$, there is a derivation of G from u in Γ such that the isolated internal vertices of G come from isolated internal vertices of right-hand sides of rules of Γ.

Every grammar Γ can be transformed into an equivalent one satisfying this condition. One needs to introduce new nonterminals. For example, in the case of the above grammar Γ, we add two nonterminals u_1, u_2 of type 1 and the rules

$$u_1 \to D_1, u_2 \to D_2 \text{ and } u \to D_3,$$

where D_1, D_2, D_3 are shown in Fig. 12 (D_i is on the ith line).

The general case is routine. We omit the proof. □

Corollary 4.4. *Let Γ be a grammar. One can construct a grammar Γ' such that $\boldsymbol{L}(\Gamma') = \trianglelefteq(\boldsymbol{L}(\Gamma))$ and each set $\boldsymbol{L}(\Gamma', u)$ (where u is a nonterminal of Γ') is minor closed.*

Proof. Immediate consequence of Propositions 4.1 and 4.3. □

Example (*Example* 2.2 *continued*). The grammar Γ' is obtained by adding the two rules $u \to D$ where D is either $\overset{1}{\bullet}\ \overset{2}{\bullet}$ or $\overset{1,2}{\bullet}$. The condition of the proof of Proposition 4.3 holds and $\boldsymbol{L}(\Gamma', u) = \trianglelefteq(\boldsymbol{L}(\Gamma, u))$.

Proposition 4.5. *One cannot decide whether the set $\boldsymbol{L}(\Gamma)$ generated by a given grammar Γ is minor-closed.*

Proof (*sketch*). Let us consider the finite alphabet $X = \{a,b\}$, the free monoid $(X^*,.,\varepsilon)$ and the ordering $\preceq$ on X^*: for every words w_1, $w_2 \in X^*$, $w_1 \preceq w_2$ means that w_1 can be obtained from w_2 by deleting some letters. ($\preceq$ is called the *subword* ordering; it can be also defined as the smallest precongruence over $(X^*,.)$ which contains $\{(\varepsilon,a),(\varepsilon,b)\}$).

Let us define the following binary operations $\circ$ over $\boldsymbol{G}_2$: for every H_1, $H_2 \in \boldsymbol{G}_2$, $H_1 \circ H_2$ is the graph H obtained by gluing the second source of H_1 on the first source of H_2 and taking as ith source of H the ith source of H_i (for $i \in \{1,2\}$). Let us denote by $\boldsymbol{U}_2$ the graph in $\boldsymbol{G}_2$ consisting of only one vertex and no edge (hence with 2 sources which are identical). Then $(\boldsymbol{G}_2,\circ,\boldsymbol{U}_2)$ is a monoid.

We define a map $W : X^* \to \boldsymbol{G}_2$ as the unique monoid homomorphism from $(X^*,.,\varepsilon)$ to $(\boldsymbol{G}_2,\circ,\boldsymbol{U}_2)$ such that

$$W(a) = \overset{1}{\bullet} \rightarrow \overset{2}{\bullet}, \qquad W(b) = \overset{1}{\bullet} \leftarrow \overset{2}{\bullet}.$$

One can notice that W is also an increasing map from $(X^*,\preceq)$ to $(\boldsymbol{G}_2,\trianglelefteq)$.

In part 1 we prove that one cannot decide whether the set $\boldsymbol{L}(\Gamma)$ generated by a given context-free grammar Γ is downwards closed with respect to $\preceq$. In part 2, using the encoding W, we give a reduction of the above decision problem to the decision problem we are considering.

Part 1: Let us suppose that one could decide whether the set $\boldsymbol{L}(\Gamma)$ generated by a given context-free grammar Γ is downwards closed with respect to $\preceq$ (written $\preceq$-closed), i.e. if $u \preceq v \in \boldsymbol{L}(\Gamma)$ implies $u \in \boldsymbol{L}(\Gamma)$. The so-called "Universality Problem" (abbreviated UP) for $\boldsymbol{L}(\Gamma)$ could then be solved by the following algorithm:

— Test whether $\boldsymbol{L}(\Gamma)$ is $\preceq$-closed
- if it is not, then $\boldsymbol{L}(\Gamma) \neq X^*$,
- if it is, then
  (a) apply the algorithm described in [4] to construct a rational expression E such that $\boldsymbol{L}(E) = \boldsymbol{L}(\Gamma)$,
  (b) test whether $\boldsymbol{L}(E) = X^*$.

As the UP is undecidable for context-free grammars [10, Ch. 8], part 1 is achieved.

Part 2: Let us denote by M the subset of $\boldsymbol{G}_2$ consisting of all graphs H such that: H has two distinct sources, the sources have degree 0 or 1 and are not linked by any path, H has no cycle and no vertex of degree $\geqslant 3$. One can notice that

$$M = \trianglelefteq(W(X^*)) - W(X^*).$$

We claim that

— from every context-free grammar Γ over X^*, one can construct a HR grammar $W(\Gamma)$ such that

$$\boldsymbol{L}(W(\Gamma)) = W(\boldsymbol{L}(\Gamma)) \cup M \tag{1}$$

— for every subset $L \subseteq X^*$,

$$L \text{ is } \preceq\text{-closed} \Leftrightarrow W(L) \cup M \text{ is } \trianglelefteq\text{-closed}. \tag{2}$$

From (1) and (2), it follows that the "subword-closedness" problem for context-free grammars is reducible to the "minor-closedness" problem for HR grammars. But the first problem is undecidable (by part 1), hence the second problem is undecidable too. □

If we are given a grammar Γ with the information that "$\boldsymbol{L}(\Gamma)$ is minor-closed" we shall construct the set of obstructions $\Omega(\boldsymbol{L}(\Gamma))$ not directly from Γ but from the grammar Γ' constructed by Corollary 4.4, because we need to work with a grammar such that *each* nonterminal generates a minor-closed set.

5. Obstructions

The following result is known from Robertson and Seymour [11]:

Theorem 5.1 (Graph Minor Theorem). *For every infinite sequence of graphs $(G_i)_{i\geqslant 0}$ all of the same type n, there exists $0 \leqslant i < j$ such that $G_i \trianglelefteq G_j$.*

Discussion. The Graph Minor Theorem is proved in [11] for graphs with pairwise distinct sources. However Theorem 5.1 follows easily from the special case: consider $(G_i)_{i\geqslant 0}$, where all the graphs are of same type, there is an infinite subsequence $G_{i_1}, G_{i_2}, \ldots, G_{i_n} \ldots$ such that for some empty graph E, $\varepsilon(G_{i_j}) = E$ for all j.

The graphs of this subsequence can be considered as having all $\mathrm{card}(V_E)$ pairwise distinct sources, hence the result of [11] applies and we have $G_{i_j} \trianglelefteq G_{i_{j'}}$, for some $j < j'$, as was to be proved.

Let L be a minor-closed set of graphs, all of same type, say n. We let

$$\Omega(L) = \{G \mid G \text{ has type } n,\ G \notin L \text{ and every proper minor of } G \text{ is in } L\}. \tag{3}$$

Let M be a set of graphs, all of same type, say n. We let

$$\mathrm{FORB}(M) = \{G \mid G \text{ has type } n \text{ and no minor of } G \text{ is in } M\},$$

so that

$$L = \mathrm{FORB}(\Omega(L)). \tag{4}$$

By Theorem 4.1, $\Omega(L)$ is finite and (4) gives a finitary description of L.

Our aim is to show that one can construct effectively the set $\Omega(\trianglelefteq(\boldsymbol{L}(\Gamma_0)))$ where Γ_0 is a given grammar. We shall need the notion of tree-width

5.1. Tree-width

Definition. Let G be a graph. A *tree-decomposition* of G is a pair (T,f) consisting of an undirected tree T, and a mapping $f: V_T \to \mathscr{P}(V_G)$ such that

T1 $V_G = \bigcup\{f(i) \mid i \in V_T\}$,
T2 every edge of G has its vertices in $f(i)$ for some i
T3 if $i, j, k \in V_T$, and if j is on the unique path in T from i to k, then $f(i) \cap f(k) \subseteq f(j)$,
T4 all sources of G are in $f(i)$ for some i in V_T.

The *width* of such a decomposition is defined as

$$\mathrm{Max}\{\mathrm{card}(f(i)) \mid i \in V_T\} - 1.$$

The *tree-width* of G is the minimum width of a tree-decomposition of G. It is denoted by $\mathrm{twd}(G)$. For a 0-graph, condition T4 is always satisfied in a trivial way. Trees are of tree-width 1, series-parallel graphs are of tree-width 2 (or 1 in degenerated cases), a complete graph with n vertices is of tree-width $n-1$.

The *tree-width of a set L* of graphs (denoted by $\mathrm{twd}(L)$) is the least upper bound in $\mathbb{N} \cup \{\infty\}$ of $\{\mathrm{twd}(G) \mid G \in L\}$. The set of complete graphs and the set of square grids have infinite tree-width.

Lemma 5.2 (Courcelle [5]). *Let Γ be a grammar. Let $D[e_1, \ldots, e_l]$ be the right-hand side of a rule in Γ.*
1. *If $G_1, \ldots, G_l$ are graphs of respective types $\tau(e_1), \ldots, \tau(e_l)$ and of tree-width at most k, then the graph $D[G_1/e_1, \ldots, G_l/e_l]$ has tree-width at most k, if $k + 1 \geqslant \mathrm{Card}(V_D)$.*
2. *For every nonterminal u of Γ, the tree-width of $\boldsymbol{L}(\Gamma, u)$ is at most k if $k + 1 \geqslant max\{\mathrm{Card}(V_D) \mid (u, D)$ is a rule of $\Gamma\}$.*

We shall use Monadic Second-order logic (MS) to describe sets of graphs.

5.2. Monadic Second-order Logic

We shall use Monadic Second-order Logic (MS) to write formally hypergraph properties and to describe sets of hypergraphs. If φ is a MS formula (see below), we shall write $H \models \varphi$ in order to denote that a hypergraph H satisfies the property expressed by φ. The set of (finite) hypergraphs H such that $H \models \varphi$ is the set of hypergraphs *defined by* φ. A hypergraph $H \in \boldsymbol{G}(A)_n$ is considered as a logical structure $< \boldsymbol{D}_H, (\mathrm{edg}_{a,H})_{a \in A}, (s_{i,H})_{1 \leqslant i \leqslant n} >$ with domain $\boldsymbol{D}_H := V_H \cup \boldsymbol{E}_H$, constants $s_{i,H}$, $1 \leqslant i \leqslant n$, denoting the sources of H and $(\tau(a)+1)$-ary relations $\mathrm{edg}_{a,H}$ such that $\mathrm{edg}_{a,H}(e, x_1, \ldots, x_n)$ holds iff e is a hyperedge with label a and $(x_1, \ldots, x_n)$ is its sequence of vertices. Every closed first-order formula φ written with the symbols edg_a $(a \in A)$, $s_i (1 \leqslant i \leqslant n))$ is either true or false in the logical structure associated with $H \in \boldsymbol{G}(A)_n$. However, first-order logic is insufficient for our purposes, and we shall need monadic second-order logic, its extension using *set* variables. Formal definitions can be found in [2, 3, 5–7]. We give only as an example the following MS formula (for $A = \{a\}$, $\tau(a) = 2$):

$$\exists X \quad (\exists u \exists x \exists y(\mathrm{edg}_a(u,x,y) \wedge x \in X) \wedge$$
$$(\exists u \exists x \exists y(\mathrm{edg}_a(u,x,y) \wedge x \notin X) \wedge$$
$$(\forall u \forall x \forall y(\mathrm{edg}_a(u,x,y) \Rightarrow (x \in X \Leftrightarrow y \in X))).$$

This formula expresses the nonconnectedness of a graph H, having no isolated vertex. (The upper case variable X denotes subsets of $\boldsymbol{D}_H$; the lower-case variables u, x, y denote elements of $\boldsymbol{D}_H$).

Given a MS formula φ and an integer n, we denote the set of finite models of φ by

$$\boldsymbol{L}(\varphi,n) = \{H \in \boldsymbol{G}(A)_n | H \models \varphi\}.$$

The following fundamental result will be used in the sequel.

Theorem 5.3 (Courcelle [2]). *Given integers k,n and a MS-formula φ, one can decide whether the following is true:*,

$$\forall H \in \boldsymbol{G}_n, \mathrm{twd}(H) \leqslant k \Rightarrow (H \models \varphi).$$

This result is no longer true if we omit the constraint "twd$(H) \leqslant k$". See [2].

The three next lemmas will be used in our main construction.

Lemma 5.4 (Courcelle [5, Lemma 4.2]). *For every finite graph K, one can construct a closed MS formula θ_K defining the set of graphs that contain K as a minor.*

Let us use in the next lemma the notations of Section 2.3.

Lemma 5.5. *Let $D \in \boldsymbol{G}(A \cup U)_n$, let $P = (D, e_1, \dots, e_l)$ be an associated graph operator and let $\varphi_1, \dots, \varphi_m$ be closed MS formulas. Then one can construct a closed MS formula ψ such that, $\boldsymbol{L}(\psi,n) = \hat{D}(\boldsymbol{L}(\varphi_1,n_1), \dots, \boldsymbol{L}(\varphi_m,n_m))$.*

We use in the proof the notions of unit graph and the function $G \mapsto \varepsilon(G)$ defined in Section 3. Let us say that a MS formula φ is *rigid* iff for every $H, K \in \boldsymbol{G}(A)$, if $H \models \varphi$ and $K \models \varphi$ then $\varepsilon(H) = \varepsilon(K)$. Let us say that a graph D is *identification-free* iff, for every edge e of D, if the label of e is an element of U then, for every $1 \leqslant i < j \leqslant \tau(e)$, $\mathrm{vert}_D(e,i) \neq \mathrm{vert}_D(e,j)$.

Proof (*sketch*). Let us denote by $P = (D, e_1, \dots, e_l)$ a standard graph operator associated to D.

First special case: *D is a unit graph.* Let u_i be the unique label of D. One can construct from φ and D a formula φ_i' such that, for every graph H, $H \models \varphi_i'$ iff there exists some graph H' obtained from H by separating some sources which are identical in D, and such that $H' \models \varphi_i$. Hence

$$H \models \varphi_i' \Leftrightarrow \text{there exists } H_1, \dots, H_l \text{ such that for all } j$$
$$H_j \models \varphi_{i_j} \text{ and } H = \bar{P}[H_1, \dots, H_l].$$

Second special case: *$\varphi_1, \dots, \varphi_m$ are rigid.* D can be written in the form

$$D = E[U_1/e_1, U_2/e_2, \dots, U_l/e_l],$$

where $E \in \boldsymbol{G}(A \cup V)_n$, $V = \{v_1, \dots, v_l\}$ is a new ranked alphabet of variables, for every $j \in [1, l]$, label$(e_j) = v_j, \tau(v_j) = \tau(u_{i_j})$, U_j is a unit graph labelled by u_{i_j} and E is identification-free. Let us define now sequences of hypergraphs E_i' (for $1 \leqslant i \leqslant m$), V_j, V_j', V_j'', E_j'' (for $1 \leqslant j \leqslant l$) by

V_j is the only unit graph, labelled by v_j, such that $\varepsilon(V_j) = \varepsilon(U_j)$,
$E_i' = \varepsilon(H_i)$ for every model H_i of $\varphi_i (1 \leqslant i \leqslant m)$,
V_j' is the only unit graph, labelled by v_j, such that $\varepsilon(V_j') = \varepsilon(E_{i_j}')$,
$V_j'' = \bar{V}_j[V_1', \dots, V_l']$,
$E_j'' = \varepsilon(V_j'')$.

The required formula ψ will be of the form $\exists X_0, X_1, \dots, X_l . \psi'$ where ψ' expresses the following conditions on a graph H:
— the subgraph induced by X_0 is isomorphic to $\bar{E}[E_1'', \dots, E_j'', \dots, E_l'']$,
— the subgraph induced by X_j is isomorphic to some $\bar{V}_j''[H_1, \dots, H_l]$ where for every j, $H_j \models \varphi_{i_j}$ (this can be expressed by a MS formula by the first special case),
— the vertices in $X_0 \cap X_j$ correspond exactly to the vertices of the edge labelled v_j in $\bar{E}[V_1'', \dots, V_l'']$.

General case: For every empty graph E there exists a formula θ_E such that, for every graph G, $G \models \theta_E \Leftrightarrow \varepsilon(G) = \varepsilon(E)$. For every formula φ expressing properties of graphs in $\boldsymbol{G}(A)_n$, φ is equivalent to the disjunction $\bigvee_{E \in \mathscr{E}_n} (\varphi \wedge \theta_E)$, $\mathscr{E}_n$ denotes the set of all empty graphs of type n. Every formula $(\varphi \wedge \theta_E)$ is rigid. Hence, by the second special case, for every empty graphs $E_1, E_2, \dots, E_m$ of respective types $n_1, n_2, \dots, n_m$, one can construct a formula $\psi_{(E_1, E_2, \dots, E_m)}$ such that

$$\boldsymbol{L}(\psi_{(E_1, E_2, \dots, E_m)}) = \hat{D}(\boldsymbol{L}(\varphi_1 \wedge \theta_{E_1}, n_1), \boldsymbol{L}(\varphi_2 \wedge \theta_{E_2}, n_2), \dots, \boldsymbol{L}(\varphi_m \wedge \theta_{E_m}, n_m)).$$

Let $\psi = \bigvee_{E_1, E_2, \dots, E_m} \psi_{(E_1, E_2, \dots, E_m)}$ where $(E_1, E_2, \dots, E_m)$ describes the set of all possible m-tuples of empty graphs of types $n_1, n_2, \dots, n_m$. We have then

$$\boldsymbol{L}(\psi, n) = \hat{D}(\boldsymbol{L}(\varphi_1, n_1), \dots, \boldsymbol{L}(\varphi_m, n_m)). \qquad \square$$

Let us use in next lemma the notations of Section 2.5.

Lemma 5.6. *Let $\Gamma = (A, U, Q, Z)$ be a grammar where*

$$U = \{u_1, \dots, u_m\} \text{ and } Q(u_i) = \{D_{i,1}, \dots, D_{i,s_i}\} \quad (\text{for } 1 \leqslant i \leqslant m).$$

Let $\varphi_1, \dots, \varphi_m$ be closed MS formulas. Then one can construct closed MS formulas $\psi_1, \dots, \psi_m$ such that, for every $i \in [1, m]$,

$$\boldsymbol{L}(\psi_i, n_i) = \hat{t}_i(\boldsymbol{L}(\varphi_1, n_1), \dots, \boldsymbol{L}(\varphi_m, n_m)).$$

Proof. Let $\psi_{i,j}$ be the formula associated to $D_{i,j}$ and $(\varphi_1, \dots, \varphi_m)$ by Lemma 5.5. The formulas

$$\psi_i = \bigvee_{1 \leqslant j \leqslant s_i} \psi_{i,j}$$

fulfill the required property. □

5.3. Main construction

Let Γ_0 be a grammar. We show here how to construct $\Omega(\trianglelefteq(\boldsymbol{L}(\Gamma_0)))$. By Corollary 4.4 one can construct a grammar Γ with nonterminals $u_1, \dots, u_m$ such that

$$\boldsymbol{L}(\Gamma, u_1) = \trianglelefteq(\boldsymbol{L}(\Gamma_0)), \tag{5}$$

$$\boldsymbol{L}(\Gamma, u_i) = \trianglelefteq(\boldsymbol{L}(\Gamma, u_i)) \text{ for } i = 2, \dots, m. \tag{6}$$

By Theorem 3.4, one can construct a proper grammar Γ' with nonterminals $u_1, \dots, u_m$, such that

$$\boldsymbol{L}(\Gamma', u_i) = \boldsymbol{L}(\Gamma, u_i)) - \mathrm{EMPTY}(u_i) \text{ for } i = 1, \dots, m. \tag{7}$$

In order to give an explicit and decidable characterization of the sets of obstructions $(\Omega(\boldsymbol{L}(\Gamma, u_i)))_{1 \leqslant i \leqslant m}$, we define the following sets of graphs for every $k, n \in \mathbb{N}$:

$\Lambda(k,n) = \{H \in \boldsymbol{G}_n \mid H$ is a n-graph formed from an orientation of the $(k+1) \times (k+1)$ square grid augmented with at most n isolated vertices in such a way that the set of isolated vertices is exactly the set of sources$\}$
$\mathrm{TWD}(k,n) = \{H \in \boldsymbol{G}_n \mid \mathrm{twd}(H) \leqslant k\}$,
$\mathrm{EMPTY}(n) = \{H \in \boldsymbol{G}_n \mid H$ is empty$\}$.

An example of graph in $\Lambda(2,4)$ is shown in Fig. 13.

Lemma 5.7. *For every k, $n \in \mathbb{N}$, there exists $k' \in \mathbb{N}$, such that*

$$\mathrm{TWD}(k,n) \subseteq \mathrm{FORB}(\Lambda(k,n)) \subseteq \mathrm{TWD}(k',n). \tag{8}$$

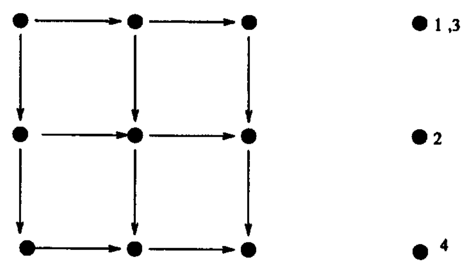

Fig. 13.

Proof. (1) The undirected $(k+1)\times(k+1)$-grid Q_{k+1} has tree-width $k+1$. It follows that every element in $\Lambda(k,n)$ has tree-width $= \max(k+1, n-1)$. Since $H \trianglelefteq G$ implies $\mathrm{twd}(H) \leqslant \mathrm{twd}(G)$, it follows that if $H \in \Lambda(k,n), G \in \boldsymbol{G}_n$ and $H \trianglelefteq G$ then $\mathrm{twd}(G) \geqslant k+1$. Hence,

$$\mathrm{TWD}(k,n) \subseteq \mathrm{FORB}(\Lambda(k,n)).$$

(2) Let $k'' = 20^{2(k+1)^5}$, $k' = k'' + n - 1$. It is proved in [12] that every undirected 0-graph of tree-width $\geqslant k''$ contains Q_{k+1} as a minor. Let H be a (directed) n-graph of tree-width $\geqslant k'+1$. Let H' be obtained from H by removing the sources and the edges incident with the sources and by forgetting the orientation. Then we have

$$\mathrm{twd}(H') \geqslant k''$$

(because adding n vertices and edges between the new vertices and the others increases the tree-width of a graph by at most n).

Hence $Q_{k+1} \trianglelefteq H'$, which implies that some element of $\Lambda(k,n)$ is a minor of H. This proves that

$$\mathrm{FORB}(\Lambda(k,n)) \subseteq \mathrm{TWD}(k',n). \qquad \square$$

In the two next lemmas Γ, Γ' denote the grammars introduced in (5)–(7). From now on we fix an integer k such that

$$k+1 \geqslant \max\{\mathrm{Card}(V_D) \mid (u,D) \text{ is a rule of } \Gamma\}.$$

Lemma 5.8. *Let $(\Omega_1, \dots, \Omega_m)$ be an m-tuple of finite sets of graphs of respective types $n_1, \dots, n_m$ where $n_i = \tau(u_i)$. Then $(\Omega_1, \dots, \Omega_m) = (\Omega(\boldsymbol{L}(\Gamma,u_1)), \dots, \Omega(\boldsymbol{L}(\Gamma,u_m)))$ iff the following conditions hold*:

C1 *$\forall H \in \Omega_i, H$ is a minimal element of $\boldsymbol{G}_{n_i} - \boldsymbol{L}(\Gamma,u_i)$ (with respect to the ordering $\trianglelefteq$)*

C2 $\mathrm{FORB}(\Omega_i) \cap \mathrm{EMPTY}(n_i) = \boldsymbol{L}(\Gamma,u_i) \cap \mathrm{EMPTY}(n_i)$

C3 *the m-tuple $(L'_1, \dots, L'_m)$ where $L'_i = \mathrm{FORB}(\Omega_i)\text{-}\mathrm{EMPTY}(n_i)$ is a solution of $S_{\Gamma'}$*

C4 $\mathrm{FORB}(\Omega_i) \subseteq \mathrm{FORB}(\Lambda(k,n_i))$

Proof. (1) Let us suppose $(\Omega_1, \dots, \Omega_m) = (\Omega(\boldsymbol{L}(\Gamma,u_1)), \dots, \Omega(\boldsymbol{L}(\Gamma,u_m)))$. Condition **C1** is clear from the definition of an obstruction. Conditions **C2**, **C3** follow from the facts that $\mathrm{FORB}(\Omega_i) = \boldsymbol{L}(\Gamma,u_i)$ and equality (7). By Lemma 5.2,

$$\boldsymbol{L}(\Gamma,u_i) \subseteq \mathrm{TWD}(k,n_i).$$

Hence, by Lemma 5.4

$$\boldsymbol{L}(\Gamma,u_i) \subseteq \mathrm{FORB}(\Lambda(k,n_i))$$

which implies Condition **C4**.

(2) Let us suppose that $(\Omega_1, \dots, \Omega_m)$ fulfills conditions **C1–C4**. By **C3**, the definition of Γ' and Theorem 3.3,

$$\mathrm{FORB}(\Omega_i) - \mathrm{EMPTY}(n_i) = \boldsymbol{L}(\Gamma', u_i). \tag{9}$$

Together with **C2** it shows that

$$\mathrm{FORB}(\Omega_i) = \boldsymbol{L}(\Gamma, u_i) = \mathrm{FORB}(\Omega(\boldsymbol{L}(\Gamma, u_i))).$$

Condition **C1** allows then to conclude that

$$\Omega_i = \Omega(\boldsymbol{L}(\Gamma, u_i)). \qquad \square$$

Remark. (1) This proof also shows that $(\Omega_1, \dots, \Omega_m) = (\Omega(\boldsymbol{L}(\Gamma, u_1)), \dots, \Omega(\boldsymbol{L}(\Gamma, u_m)))$ iff the conjunction **C1** $\wedge$ **C2** $\wedge$ **C3** holds. It turns out that **C4** is useful for effectivity purposes, especially in next lemma.

(2) Let us notice that we make an essential use of the *unicity* of solution of Γ' when we deduce equality (9) from condition **C3**. This was the motivation for working with the grammar Γ' rather than with the initial grammar Γ.

Lemma 5.9. *It is decidable whether a given m-tuple* $(\Omega_1, \dots, \Omega_m)$ *of finite sets of graphs of respective types* $n_1, \dots, n_m$ *is equal to* $(\Omega(\boldsymbol{L}(\Gamma, u_1)), \dots, \Omega(\boldsymbol{L}(\Gamma, u_m)))$.

Proof. Let $(\Omega_1, \dots, \Omega_m)$ be some m-tuple of finite sets of graphs of respective types $n_1, \dots, n_m$ and let k' be some integer such that (8) is true for every pair (k, n_i) (such an integer k' exists, by Lemma 5.7). It suffices to show that the conjunction of conditions **C1–C4** given in Lemma 5.5 is decidable. Condition **C1** is decidable (because membership in $\boldsymbol{L}(\Gamma, u_i)$ is decidable). Condition **C2** is decidable, because both sides of the equality are finite sets which can be computed. Condition **C4** is equivalent to

$$\forall H \in \Lambda(k, n_i), \exists K \in \Omega_i, K \trianglelefteq H.$$

This can be tested since $\Lambda(k, n_i)$, Ω_i are finite sets. Let us now assume that condition **C4** is fulfilled. By condition **C4** and the choice of k'

$$L_i' = \mathrm{FORB}(\Omega_i) - \mathrm{EMPTY}(n_i) \subseteq \mathrm{TWD}(k', n_i).$$

By Lemma 5.4, one can construct from Ω_i a MS-formula φ_i such that, for every $H \in \boldsymbol{G}_{n_i}$:

$$H \models \varphi_i \Leftrightarrow H \in \mathrm{FORB}(\Omega_i) - \mathrm{EMPTY}(n_i).$$

By Lemma 5.6, from Γ' and $\varphi_1, \dots, \varphi_m$, one can construct MS formulas ψ_i such that, for every $H \in \boldsymbol{G}_{n_i}$

$$H \models \psi_i \Leftrightarrow H \in \hat{t}_i'(\boldsymbol{L}(\varphi_1, n_1), \dots, \boldsymbol{L}(\varphi_m, n_m))$$

(where t_i' is associated with Γ' as t_i is with Γ in Section 2.3). Condition **C3** is then equivalent to the condition that

$$\text{for all } i \in [1,m], \quad \text{for all } H \in \mathrm{TWD}(k', n_i), \ H \models (\varphi_i \Leftrightarrow \psi_i)$$

which is decidable by Theorem 5.3.

Theorem 5.10. *Let L be a set of graphs defined by a given HR-grammar Γ. One can construct effectively the set of obstructions of the minor-closure of L.*

Proof. By equality (5), $\trianglelefteq(L) = \trianglelefteq(\boldsymbol{L}(\Gamma_0)) = \boldsymbol{L}(\Gamma, u_1)$. It suffices to enumerate m-tuples $\boldsymbol{\Omega} = (\Omega_1, \ldots, \Omega_m)$ and test (by Lemma 5.9) each of them for conditions **C1**–**C4** of Lemma 5.8, until a m-tuple $\boldsymbol{\Omega}$ fulfilling them is reached. Then $\Omega(\trianglelefteq(L)) = \Omega_1$. □

We now discuss the extension of Theorem 5.10 to HR grammars generating sets of undirected graphs.

An edge e in a hypergraph H is undirected if $\mathrm{vert}_H(e)$ is a set of one or two vertices. (If $\mathrm{vert}_H(e) = \{v\}$ then e is a loop incident with v; if $\mathrm{vert}_H(e) = \{u, v\}$ then e is an edge linking u and v). In HR grammars terminal edges only can be undirected. The results of Sections 2–4 extend immediately. Clearly, a minor of an undirected graph is undirected. Theorem 5.1 for undirected graphs is an immediate consequence of its version for directed ones. Tree-width does not depend on orientation. In the proof of Lemma 5.7, the sets $\Lambda(k, n)$ are much smaller because they consist of square (undirected) grids with additional isolated vertices that are sources: we do not have to consider all orientations of the grids. All proofs of Section 5 extend easily.

6. Conclusions

We do not claim to have provided an *efficient* algorithm for constructing the obstructions of a minor-closed set of graphs defined by a HR grammar. We have given a *computability* result only.

Let us compare our result with that of Fellows and Langston [8]. They prove that one can construct the set of obstructions of a minor-closed set of graphs L (not necessarily of bounded tree-width) under the following assumptions:

— one knows an upper-bound on the tree-width of the obstructions,

— one knows a MS formula characterizing the set L; [8] actually uses a congruence but it is easy to obtain a congruence from an MS formula by the results of Courcelle [2].

Hence the result of [8] is incomparable with the one established here: it concerns sets of graphs of tree-width not necessarily bounded but it cannot give Theorem 5.10 because, up to now, there is no method to obtain a congruence from a grammar without using our theorem.

References

[1] M. Bauderon, B. Courcelle, Graph expressions and graph rewritings, Math. Syst. Theory 20 (1987) 83–127.
[2] B. Courcelle, The monadic second-order logic of graphs I: recognizable sets of finite graphs. Inform. Comput. 85 (1990) 12–75.
[3] B. Courcelle, The monadic second-order logic of graphs V: on closing the gap between definability and recognizability, Theoret. Comput. Sci. 80 (1991) 153–202.
[4] B. Courcelle, On constructing obstruction sets of words, Bull. EATCS 44 (1991) 178–185
[5] B. Courcelle, The monadic second-order logic of graphs III: tree decompositions, minors and complexity issues, RAIRO Inform. Théor. Appl. 26 (1992) 257–286.
[6] B. Courcelle, Graph grammars, monadic second-order logic and the theory of graph minors, in: Graph Structure Theory, N. Robertson, P. Seymour, eds., Contemporary Mathematics Vol. 147, American Mathematical Society, Providence, RI, 1993, pp. 565–590.
[7] B. Courcelle, The expression of graph properties and graph transformations in monadic second-order logic, in: G. Rozenberg (Ed.), Handbook of graph transformations, vol. 1. Foundations, World Publishing Company, New York, 1996, pp. 313–400.
[8] M. Fellows, M. Langston, An analogue of the Myhill–Nerode theorem and its use in computing finite basis characterizations, Proc. 30th Symp. FOCS, 1989, pp. 520–525.
[9] A. Habel, Hyperedge Replacement: Grammars and languages, Lecture Notes in Computer Science, vol. 643, Springer, Berlin, 1993.
[10] M.A. Harrison, Introduction to Formal Language Theory, Addison-Wesley, Reading, MA, 1978.
[11] N. Robertson, P. Seymour, Graph minors XX: Wagner's conjecture, 1988, preprint.
[12] N. Robertson, P. Seymour, R. Thomas, Quickly excluding a planar graph, J. Combin. Theory B, 62 (1994) 323–348.
[13] D. Sanders, 1995, private communication.
[14] J. Valdès, R. Tarjan, E. Lawler, The recognition of series-parallel graphs, SIAM J. Comput. 11 (1982) 298–313.