MONADIC SECOND-ORDER DEFINABLE GRAPH TRANSDUCTIONS(*)

Bruno COURCELLE

Université BORDEAUX-1,
Laboratoire d'Informatique(+)
351, Cours de la Libération
33405 TALENCE, FRANCE

ABSTRACT

Formulas of monadic second-order logic can be used to specify graph transductions, i.e., multivalued functions from graphs to graphs. We obtain in this way classes of graph transductions, called *monadic second-order definable graph transductions* (or more simply *definable transductions*) that are closed under composition and preserve the two known classes of context-free sets of graphs, namely the class of Hyperedge Replacement (HR) and the class of Vertex Replacement (VR) sets. These two classes can be characterized in terms of definable transductions and recognizable sets of finite trees. These characterizations are independent of the rewriting mechanisms used to define the HR and VR grammars. When restricted to words, the definable transductions are strictly more powerful than the rational transductions with finite image; they do not preserve context-free languages. We also describe the sets of discrete (edgeless) labeled graphs that are the images of HR and VR sets under definable transductions: this gives a version of Parikh's Theorem (i.e., the characterization of the commutative images of context-free languages) which extends the classical one and applies to HR and VR sets of graphs.

INTRODUCTION

The Theory of Formal Languages investigates finite devices defining sets of finite and countably infinite words and trees, compares their expressive powers, and investigates the solvability of the associated decision problems. These investigations make an essential use of transformations from words or trees to words or trees usually called *transductions*. Of special importance are rational transductions; they are closed under composition and inverse, and they preserve the families of recognizable and context-free languages. Tree transductions are more complicated, and there is no unique notion that can be considered as the analogue of that of a rational transduction. For each class of tree transductions, the

(*) Supported by the ESPRIT Basic -Research project 3299 ("Computing by graph transformations") and by the "Programme de Recherches Coordonnées: Mathématiques et Informatique".

(+) Unité associée au CNRS n° 1304, email : courcell@geocub.greco-prog.fr

closure under composition is a major concern, and so is the preservation of recognizability; we refer the reader to the survey by Raoult [Rao]. Another important transduction is **yield** that maps derivation trees of context-free grammars to the corresponding words. The context-free languages can be characterized as the images of the recognizable sets of (finite) trees under **yield** mappings.

The study of sets of finite and countably infinite graphs (and hypergraphs) by tools like grammars, systems of equations and logical formulas is a relatively recent development of the Theory of Formal Languages. The need for a manageable and powerful notion of graph transduction appears in constructions dealing with graph grammars and is of interest on its own.

This paper is a survey presenting the notion of a *monadic second-order definable graph transduction* (a *definable transduction* for short), which has been introduced more or less explicitely and sometimes in restricted forms in several papers [ALS, Cou5, Cou7-10, CE, Eng2]. We collect the main results and give references to their proofs.

We now introduce informally these transductions. The term "monadic second-order" refers to a logical language, the *monadic second-order logic*. We recall the rôle of logic for defining sets of graphs (or hypergraphs; all what we shall say concerning graphs applies to hypergraphs as well). Graphs can be described by relational structures, i.e., by logical structures with no function symbols. The domain of the structure representing a graph is the set of its vertices and edges put together; basic relations describe the incidence of vertices and edges and possible labellings. (This is actually not the only way to represent a graph; see Sections 1 and 4 for more details.) Hence, formulas of appropriate logical languages define properties of this graph. Monadic second-order logic is popular among logicians because of its expressive power and its decidability properties. (See Gurevich [Gu] for a survey.) For dealing with graphs, it is very useful because it can express many fundamental properties (like planarity, connectivity, k-colorability for fixed k) whereas several general decidability results hold. The sets of words (resp. of trees) characterized by a property expressible in monadic second-order logic are exactly the recognizable sets by the results of Büchi and Elgot [Bü, Elg], see [Tho, Thm 3.2] (resp. of Doner [Don], see [Tho, Thm 11.1]). Sets of graphs defined similarly by characteristic monadic second-order

properties behave very much like recognizable sets of words and trees, in particular in constructions involving context-free graph grammars. Since no notion of finite-state graph automaton is known, monadic second-order formulas are essential in such constructions.

We now come to graph transductions. Since we have no good (general) notion of graph automaton, we have no chance to obtain a good notion of graph transduction based on a finite-state machine model. Alternatively, we propose to define transductions of graphs (or more generally of relational structures) by means of monadic second-order formulas. The idea is to transform a structure S into a structure T by defining T "inside" S by means of such formulas. This is nothing but the classical notion of *semantic interpretation* (see for instance [Rab]), appropriately extended. In particular, we define T inside the *disjoint union of k copies of* S (for some fixed k), which makes possible to construct T with a domain larger than that of S, (larger within the factor k).

The family of definable transductions is closed under composition, but not under inverse. These transductions preserve the so-called HR and VR sets of graphs (namely, the two known types of context-free sets of graphs), and yield grammar independent characterizations of these sets

This paper is organized as follows. Section 1 reviews relational structures, the way they represent graphs and hypergraphs, and monadic second-order logic; Section 2 introduces (monadic second-order) definable graph transductions and presents examples of such transductions dealing with words, trees and graphs; Section 3 gives the main properties of definable transductions of relational structures; Section 4 presents the relationships between definable transduction and the classes of VR and HR sets of graphs (and hypergraphs); Sections 5 compares definable transductions with known transductions of words and trees; Section 6 deals with definable transductions from graphs to commutative words and gives forms of Parikh's Theorem that apply to HR and VR graph grammars.

1 - Hypergraphs and Relational Structures

A binary relation $R \subseteq A \times B$ is also called a *transduction* from A to B. We write aRb for $(a,b) \in R$, and we consider every b such that aRb as *an image of* a *under* R. The *domain* of R is $\mathbf{Dom}(R) := \{a \in A \mid aRb \text{ for some } b \text{ in } B\}$ and the *image* of R is $\mathbf{Im}(R) := \{b \in B \mid aRb \text{ for some } a \text{ in } A\}$. For $L \subseteq A$, the *image of* L *under* R is $R(L) := \{b \in B \mid aRb \text{ for some } a \text{ in } L\}$. The transduction $R^{-1} := \{(b,a) / (a,b) \in R\}$ is called the *inverse* of R. If L is a subset of B, then $R^{-1}(L)$ is the *inverse image of* L *under* R. We

say that R is *functional* if **Card**(R({a})) ≤ 1 for every a in A. We identify functional relations $R \subseteq A \times B$ with partial functions $R : A \longrightarrow B$, and we write b = R(a) instead of $b \in R(a)$. The composition of two transductions $R \subseteq A \times B$ and $S \subseteq B \times C$ is the transduction from A to C, defined as {(a,c) / (a,b) ∈ R and (b,c) ∈ S for some b in B}. By a *mapping*, we shall mean a total function.

Hypergraphs with sources

We shall deal with labeled, directed hypergraphs equipped with a sequence of distinguished vertices (called the *sources*) as in [BC, Cou3-8]. The labels (intended to label hyperedges) are chosen in a finite ranked alphabet A. The rank mapping is $\rho : A \longrightarrow \mathbb{N}$. The rank of the label of a hyperedge must be equal to the length of its sequence of incident vertices. This rank may be 0, i.e., we allow hyperedges with no vertex. Graphs appear as a special case where all labels are of rank 2.

A *concrete n-hypergraph over* A is a quintuple $G = \langle \mathbf{V}_G, \mathbf{E}_G, \mathbf{lab}_G, \mathbf{vert}_G, \mathbf{src}_G \rangle$ where: $\mathbf{V}_G$ is the finite set of vertices; $\mathbf{E}_G$ is the finite set of (hyper)edges, disjoint with $\mathbf{V}_G$; its elements will be called hereafter *edges* for shortness sake; $\mathbf{lab}_G$ is a mapping: $\mathbf{E}_G \longrightarrow A$ that defines the *label* of an edge; $\mathbf{vert}_G$ is a mapping that associates with every edge e the *sequence of its vertices*; this sequence must be of length $\rho(e) := \rho(\mathbf{lab}_G(e))$ (called the *rank* of the edge) and its ith element is denoted by $\mathbf{vert}_G(e,i)$; $\mathbf{src}_G$ is a sequence of vertices of length n (or equivalently a mapping : $[n] \longrightarrow \mathbf{V}_G$), called the *sequence of sources*; we shall denote by $\mathbf{src}_G(i)$ the ith element of this sequence; if n=0, then G has no source.

The integer n is the *rank* of G, denoted by ρ(G). A concrete hypergraph G is *simple* if, for all e, e' in $\mathbf{E}_G$, if $\mathbf{vert}_G(e) = \mathbf{vert}_G(e')$ and $\mathbf{lab}_G(e) = \mathbf{lab}_G(e')$, then e = e'. By a *hypergraph*, we mean the isomorphism class of a concrete hypergraph (possibly with sources). We denote by $\mathbf{HG}(A)_n$ the set of n-hypergraphs over A. We denote by **HG**(A) the set of hypergraphs over A of all ranks.

Operations on **HG**(A) have been defined in [BC, Cou3-5] that make possible to denote hypergraphs by well-formed algebraic expressions (called *hypergraph expressions*) written with symbols denoting these operations and nullary symbols denoting basic hypergraphs. Hence, finite hypergraph expressions denote finite hypergraphs. Infinite hypergraph expressions can also be defined (they are analogous to formal power series), and they denote countable hypergraphs (see

[Cou5-7]). The hypergraph denoted by an expression t is called its value, and is denoted by **val**(t). The operations are the following ones: a disjoint noncommutative union that concatenates the sequences of sources (it associates with G and G' a hypergraph of rank $\rho(G) + \rho(G')$); unary operations indexed by pairs of integers (the operation associated with a pair (i,j) transforms a hypergraph by fusing its ith and jth sources; the rank of the hypergraph is preserved) and unary operations indexed by mappings between integers (the operation associated with a mapping α from [n] to [p] transforms a hypergraph G of rank p into the hypergraph H of rank n such that $\mathbf{src}_H(i) = \mathbf{src}_G(\alpha(i))$ for all i in [n], all other components of H being as in G.

Monadic second-order logic

Let R be a finite ranked set of symbols where each element r in R has a rank $\rho(r)$ in $\mathbb{N}_+$. A symbol r in R is considered as a $\rho(r)$-ary relation symbol. An R-*(relational) structure* is a tuple $S = \langle \mathbf{D}_S, (r_S)_{r \in R} \rangle$ where $\mathbf{D}_S$ is a finite (possibly empty) set, called the *domain* of S, and r_S is a subset of $\mathbf{D}_S^{\rho(r)}$ for each r in R. We denote by $\mathcal{S}(R)$ the class of R-structures.

We review *monadic second-order logic* briefly. Its formulas (called *MS formulas* for short), intended to describe properties of structures S as above, are written with variables of two types, namely lowercase symbols x,x',y,... called *object variables*, denoting elements of $\mathbf{D}_S$, and uppercase symbols X,Y,Y',... called *set variables*, denoting subsets of $\mathbf{D}_S$. The atomic formulas are of the forms x = y, $r(x_1,\ldots,x_n)$ (where r is in R and $n=\rho(r)$), and $x \in X$, and formulas are formed with propositional connectives and quantifications over the two kinds of variables. See [Cou2-Cou8] for more details. For every finite set W of object and set variables, we denote by $\mathfrak{F}(R,W)$ the set of all formulas that are written with relational symbols from R and have their free variables in W; we also let $\mathfrak{F}(R) := \mathfrak{F}(R,\emptyset)$ denote the set of closed formulas.

Let S be an R-structure, let $\varphi \in \mathfrak{F}(R,W)$, and let γ be a W-assignment in S, (i.e., $\gamma(X)$ is a subset of $\mathbf{D}_S$ for every set variable X in W, and $\gamma(x) \in \mathbf{D}_S$ for every object variable x in W; we write this $\gamma : W \longrightarrow S$ to be short). We write $(S,\gamma) \models \varphi$ iff φ holds in S for γ. We write $S \models \varphi$ in the case where φ has no free variable. A set of R-structures L is *definable* if there is a formula φ in $\mathfrak{F}(R)$ such that L is the set of all R-structures S such that $S \models \varphi$.

A hypergraph G in $\mathbf{HG}(A)_n$ can be represented by an $\mathbf{R}_{A,n}$-structure, where $\mathbf{R}_{A,n}$:= $\{\mathbf{edg}_a \mid a \in A\} \cup \{\mathbf{ps}_i \mid i \in [n]\}$ with $\rho(\mathbf{edg}_a) := \rho(a)+1$ and $\rho(\mathbf{ps}_i) := 1$. The structure representing G is $|G|_2 := \langle \mathbf{D}_G, (\mathbf{edg}_{aG})_{a\in A}, \mathbf{ps}_{1G}, \ldots, \mathbf{ps}_{nG} \rangle$ where $\mathbf{D}_G := \mathbf{V}_G \cup \mathbf{E}_G$ (let us recall that $\mathbf{V}_G \cap \mathbf{E}_G = \emptyset$), $\mathbf{edg}_{aG}(x,y_1,\ldots,y_n) : \Leftrightarrow x \in \mathbf{E}_G$, $\mathbf{lab}_G(x) = a$, and $\mathbf{vert}_G(x) = (y_1,\ldots,y_n)$, $\mathbf{ps}_{iG}(x) : \Leftrightarrow x = \mathbf{src}_G(i)$.

Clearly $|G|_2$ is isomorphic to $|G'|_2$ as a relational structure iff G = G' (i.e., G is isomorphic to G'). We shall consider any two isomorphic structures as equal, like for hypergraphs.

A hypergraph G can be represented by another structure $|G|_1 := \langle \mathbf{V}_G, (\mathbf{edg'}_{aG})_{a\in A}, \mathbf{ps}_{1G}, \ldots, \mathbf{ps}_{nG} \rangle$ where $\mathbf{edg'}_{aG}(y_1,\ldots,y_n)$ holds iff $\mathbf{lab}_G(e) = a$ and $\mathbf{vert}_G(e) = (y_1,\ldots,y_n)$ for some e in $\mathbf{E}_G$, and $\mathbf{ps}_{iG}$ is as in $|G|_2$. Clearly, two simple hypergraphs G and G' are equal iff $|G|_1$ is equal (isomorphic) to $|G'|_1$. Hence, simple hypergraphs are unambigously represented by these latter structures whereas arbitrary hypergraphs are not.

The representation of hypergraphs by logical structures makes it possible to express their properties by logical formulas. We shall say that a property P of the hypergraphs G of a class $\mathfrak{C}$ can be *expressed by a logical formula* φ *via a representation* $|G|$ if, for every G in $\mathfrak{C}$, then P(G) holds iff $|G| \models \varphi$.

A property of hypergraphs is *i-definable* (where i is 1 or 2), if it is expressible by a MS formula, relative to the representation $|\text{-}|_i$. To give an example, the following formula expresses that a graph G represented by the structure $|G|_1$ is connected:

$$\forall x \forall y \forall X [\forall u \forall v ((u \in X \wedge \text{"u-v"}) \Rightarrow v \in X) \wedge x \in X \Rightarrow y \in X]$$

where "u-v" stands for the formula expressing that u and v belong to some edge, namely, for the disjunction of the formulas $\mathbf{edg'}_a(u,v) \vee \mathbf{edg'}_a(v,u)$ extended to all labels a. (All labels are assumed to be of rank 2).

The structure $|G|_1$ is less expressive than $|G|_2$, for representing properties of a simple hypergraph G by MS formulas. (Note that G is unambigously represented by both structures). For instance, the existence of a Hamiltonian circuit in a graph is a 2-definable property that is not 1-definable. The results of

[Cou9] comparing the expressive powers of MS formulas in the two cases are recalled in Section 4.

2 - MONADIC SECOND-ORDER DEFINABLE TRANSDUCTIONS

We first define *transductions of relational structures*. Let R and R' be two finite ranked sets of relation symbols. Let W be a finite set of set variables, called here the set of *parameters*. (It is not a loss of generality to assume that all parameters are set variables.) An (R',R)-*definition scheme* is a tuple of formulas of the form

$$\Delta = (\varphi, \psi_1, \ldots, \psi_k, (\theta_w)_{w \in R'^*k})$$

where : $k > 0$, $R'^*k := \{(r,j) \mid r \in R', j \in [k]^{\rho(r)}\}$,
$\varphi \in \mathfrak{S}(R,W)$,
$\psi_i \in \mathfrak{S}(R,W \cup \{x_1\})$ for $i=1,\ldots,k$,
$\theta_w \in \mathfrak{S}(R,W \cup \{x_1,\ldots,x_{\rho(r)}\})$, for $w = (r, j) \in R'^*k$.

Let $S \in \mathcal{S}(R)$, let γ be a W-assignment in S. An R'-structure S' with domain $\mathbf{D}_{S'} \subseteq \mathbf{D}_S \times [k]$ is *defined by* Δ *in* (S,γ) if :

(i) $(S,\gamma) \models \varphi$,

(ii) $\mathbf{D}_{S'} = \{(d,i) \mid d \in \mathbf{D}_S, i \in [k], (S,\gamma,d) \models \psi_i\}$

(iii) for each r in R' :
$r_{S'} = \{((d_1, i_1),\ldots,(d_t,i_t)) \mid (S,\gamma, d_1,\ldots,d_t) \models \theta_{(r,j)}\}$,
where $j=(i_1,\ldots,i_t)$ and $t = \rho(r)$.

(By $(S,\gamma,d_1,\ldots,d_t) \models \theta_{(r,j)}$, we mean $(S,\gamma') \models \theta_{(r,j)}$, where γ' is the assignment extending γ, such that $\gamma'(x_i)=d_i$ for all $i=1,\ldots,t$; a similar convention is used for $(S,\gamma,d) \models \psi_i$.) Since S' is associated in a unique way with S, γ and Δ whenever it is defined, i.e., whenever $(S,\gamma) \models \varphi$, we can use the functional notation $\mathbf{def}_\Delta(S,\gamma)$ for S'.

The *transduction defined by* Δ is the relation $\mathbf{def}_\Delta := \{(S, S') / S' = \mathbf{def}_\Delta(S,\gamma)$ for some W-assignment γ in S$\} \subseteq \mathcal{S}(R) \times \mathcal{S}(R')$. A transduction $f \subseteq \mathcal{S}(R) \times \mathcal{S}(R')$ is *definable* if it is equal to $\mathbf{def}_\Delta$ for some (R',R)-definition scheme Δ. In the case

where $W = \emptyset$, we say that f is *definable without parameters* (note that it is functional).

(2.1) Fact: *If f is a definable transduction, there exists an integer k such that,* $\mathbf{Card}(\mathbf{D}_T) \leq k.\mathbf{Card}(\mathbf{D}_S)$ *whenever* T *belongs to* f(S).

(2.2) Fact : *The domain of a definable transduction is definable.*

Proof: Let Δ be a definition scheme as above with $W = \{X_1,...,X_n\}$. Then $\mathbf{Dom}(\mathbf{def}_\Delta) = \{S \mid S \models \exists X_1,...,\exists X_n\, \varphi\}$. □

These definitions apply to hypergraphs via their representation by relational structures as explained above. However, since we have two representations of hypergraphs by logical structures, we must be more precise. We say that a *hypergraph transduction*, i.e., a binary relation f on hypergraphs is *(i,j)-definable*, where i and j belong to {1,2} iff the transduction of structures $\{(|G|_i, |G'|_j) \mid (G,G') \in f\}$ is definable. We shall also use transductions from trees to hypergraphs. Since a tree t is a graph, it can be represented by either $|t|_1$ or $|t|_2$. However, both structures are equally powerful for expressing monadic second-order properties of trees, and definable transductions from trees to trees and from trees to graphs. (This follows from the results of [Cou9]; see Theorem (4.3) below). When we specify a transduction involving trees (or words which are special trees) as input or output we shall use the symbol * instead of the integers 1 and 2, in order to recall that the choice of representation is not important in these cases. We shall use *definable* for *(*,*)-definable.*

The next three propositions list examples of definable transductions.

(2.3) Proposition: *The following mappings are definable transductions: (1) word homomorphisms, (2) inverse nonerasing word homomorphisms, (3) gsm mappings, (4) the mirror-image mapping on words, (5) the mapping* $\lambda u.[u^n]$ *(where* u *is a word and* n *a fixed integer), (6) the mapping* ***yield*** *that maps a derivation tree relative to a fixed context-free grammar to the generated word, (7) linear root-to-frontier or frontier-to-root tree transductions.*

Proof: The proofs are easy to do. A *gsm mapping* is a transduction from words to words defined by a *generalized sequential machine*, i.e., a (possibly

nondeterministic) transducer that reads at least one input symbol on each move. See Gecseg and Steinby [GS] or the survey by Raoult [Rao] for tree transductions. □

Fact (2.1) which limits the sizes of the output structures, shows that certain transductions are *not definable*. This is the case of inverse erasing word homomorphisms and of ground tree transducers except in degenerated cases (see Dauchet *et al.* [DHLT] or [Rao] on ground tree transducers).

(2.4) Proposition [Cou5] : *The mapping* **val** *that associates with a finite or infinite hypergraph expression the hypergraph it denotes is (*,2)-definable.*

In [Cou7], systems of equations are investigated, that define infinite *equational* hypergraphs. Each equational hypergraph has an infinite *syntax tree* , i.e., some appropriately defined infinite derivation tree, that is close to an infinite hypergraph expression. The mapping from the syntax trees of a fixed system to the corresponding equational graphs is (*,2)-definable, and its inverse is (*,2)-definable.

(2.5) Proposition: *The transductions that associate with a graph G: (1) its spanning forests, (2) its connected components, (3) the connected component of some designated vertex (for instance the first source), (4) its minors, (5) its subgraphs satisfying some fixed 2-definable property, (6) its maximal subgraphs satisfying some fixed 2-definable property (maximal for subgraph inclusion), (7) the graph consisting of the union of two disjoint copies of G, are all (2,2)-definable. The mapping associating with a graph its line graph is (2,1)-definable but not (2,2)-definable.*

Proof: See [Cou6] for minors. See [CE] for line graphs. All other assertions are easy consequences of the existence of a MS formula expressing that two given vertices are linked by some path. See [Cou3-5]. □

3 - Properties of Definable Transductions of Relational Structures

The following proposition is the basic fact behind the notion of semantic interpretation ([Rab]). It says that if $S = \mathbf{def}_{\Delta}(T,\mu)$ then the monadic second-order properties of S can be expressed as monadic second-order properties of (T,μ).

Let $\Delta = (\varphi, \psi_1,\ldots, \psi_k, (\theta_w)_{w\in R^*k})$ be a (R,R')-definition scheme, written with a set of parameters $\mathcal{W}$. Let $\mathcal{V}$ be a set of set variables disjoint from $\mathcal{W}$. For every variable X in $\mathcal{V}$, for every i=1,...,k, we let X_i be a new variable. We let $\overline{\mathcal{V}} = \{X_i / X \in \mathcal{V}, i=1,\ldots,k\}$. For every $\eta: \overline{\mathcal{V}} \longrightarrow \mathcal{P}(D)$, we let $\nu : \mathcal{V} \longrightarrow \mathcal{P}(D\times[k])$ be defined by: $\nu(X) = \eta(X_1)\times\{1\} \cup \ldots \cup \eta(X_k)\times\{k\}$. With these notations we can state:

(3.1) Proposition [Cou8, Cou9]: *For every formula* β *in* $\mathcal{FS}(R,\mathcal{V})$, *one can construct a formula* $\overline{\beta}$ *in* $\mathcal{FS}(R',\overline{\mathcal{V}}\cup\mathcal{W})$ *such that, for every* T *in* $\mathcal{S}(R')$, *for every* $\mu : \mathcal{W} \longrightarrow T$, *for every* $\eta : \overline{\mathcal{V}} \longrightarrow T$, *we have :*

$\mathbf{def}_\Delta(T,\mu)$ *is defined (if it is, we denote it by* S),
ν *is a* $\mathcal{V}$*-assignment in* S, *and* $(S,\nu) \models \beta$

iff

$(T, \eta\cup\mu) \models \overline{\beta}$.

From this proposition, we get easily:

(3.2) Proposition: *(1) The inverse image of a definable set of structures under a definable transduction is definable.*

(2) The composition of two definable transductions is definable.

We could define more powerful transductions by which a structure S would be constructed "inside" T×T instead of "inside" a structure formed of k disjoint copies of T for fixed k. However, with this variant, one could construct a *second-order* formula $\overline{\beta}$ as in Proposition (3.1) (with quantifications on binary relations), but not a *monadic* second-order one (at least in general). We wish to avoid (full) second-order logic because most constructions and decidability results (like those of [Cou4]) break down.

(3.3) Proposition: *The union of two definable transductions is definable. So is the intersection of a definable transduction with a transduction of the form A×B where A and B are definable sets.*

Proof : See [Cou9] for the first assertion and [Cou8] for the second. □

Here are now some negative properties.

(3.4) Proposition: *(1) The image of a definable set under a definable transduction is a set that is not definable in general.*

(2) The inverse of a definable transduction is a transduction that is not definable in general.

(3) The intersection of two definable transductions is a transduction that is not definable in general

Proof : (1) The transduction of words mapping a^n to $b^nc^nb^n$ for $n > 0$ is definable (easy construction like those of Proposition (2.3)). The image of the definable language a^* is a language that is not regular (and even not context-free), hence not definable by the basic result of Büchi and Elgot ([Tho, Thm 3.2]) recalled in the introduction.

(2) The inverse of this transduction is not definable since, if it would be, its domain is would be definable (by Fact (2.2)), hence regular, which is not the case.

(3) The intersection of the definable transductions of words that map a^nb^m to c^n, and a^nb^m to c^m is the one that maps a^nb^n to c^n. It is not definable because its domain is not a definable language. □

4 - Definable Transductions and Context-Free Graph Grammars

There are two classes of context-free graph (and hypergraph) grammars, the class of HR (Hyperedge Replacement) and the class of VR (Vertex Replacement) grammars. *HR grammars* (which generate the *HR sets* of graphs and hypergraphs) are based on the replacement, in a hypergraph, of a hypergraph for a (hyper)edge of the same rank: the labels of edges play the rôle of terminal and nonterminal symbols in context-free grammars. These grammars can be also defined as systems of fixed point equations, written with set union and the operations on hypergraphs recalled in Section 1. (See [Cou1] for a general theory of such systems of equations). We refer the reader to [BC, Cou3, Hab, HK] for definitions and basic properties. HR grammars are context-free in the sense of [Cou2]. This means in particular that they are *confluent*, i.e., that independent derivation steps can be permuted and that the equivalence classes of derivation sequences w.r.t. permutations of independent steps can be characterized by derivation trees. Another characteristic property of context-free grammars is that the generated

sets can be characterized as forming the least solutions of systems of equations canonically associated with the considered grammar.

The relations between HR grammars and definable transductions are collected in the following theorem.

(4.1) Theorem: *(1) The mapping* ***yield*** *that associates with a derivation tree of a (fixed) HR grammar the generated hypergraph is (*,2)-definable.*

(2) A set of hypergraphs is HR iff it is the image of a recognizable set of finite trees under a (,2)-definable transduction.*

(3) The class of HR sets of hypergraphs is closed under (2,2)-definable transductions.

Proof : (1) is proved in [Cou8]; the "if" part of (2) is a difficult theorem established in [CE], whereas the "only if" part follows immediately from (1); (3) follows immediately from (2) since the composition of a (*,2)-definable transduction from trees to hypergraphs and a (2,2)-definable transduction from hypergraphs to hypergraphs is (*,2)-definable. □

Let us say that a HR grammar is *linear* if each righthand side of a production rule has at most one nonterminal, and that a set of hypergraphs is *linear HR* if there exists a linear HR grammar generating it. From the constructions establishing Theorem (4.1), we get that a set of hypergraphs is linear HR iff it is the image of a recognizable language under a (*,2)-definable transduction, and that the class LIN-HR of linear HR sets of hypergraphs is closed under (2,2)-definable transductions.

We now come to the more complex class of *VR ("Vertex Replacement")* sets of *simple* graphs. A few words of history are in order because the definition of this class has emerged from a sequence of papers. It comes out of the NLC grammars introduced by Janssens and Rozenberg [JR]; (NLC means "Node Labeled Controlled"). Not all NLC grammars are context-free because they are not always confluent (independent derivation steps cannot in general be permuted). The BNLC ("Boundary" NLC) grammars form a restriction of the general case and are confluent ([RW]). Whereas NLC grammars generate undirected graphs with unlabeled edges, the more general edNCE grammars can generate directed labeled graphs, and edge labels can be modified during the rewriting. Only the

confluent ones (the C-edNCE grammars) are context-free. Other types of grammars, the S-HH ("Separated Handle Hypergraph") grammars, that are always context-free, generate exactly the same sets of graphs as the C-edNCE grammars and give more easily equivalent systems of fixed point equations ([CER]). See also [ER2] for more details and references. A grammar independent characterization of C-edNCE sets of graphs, slightly weaker than Theorem (4.2.2) is given by Engelfriet in [Eng2]. A *VR set of graphs* is a set of directed or undirected graphs generated by either a C-edNCE or a S-HH grammar, or defined as a component of the least solution of a systems of fixed point equations written with certain appropriate operations (see [CER]). Actually, the simplest way to define them is by systems of equations (see also [Cou12-13] for these systems).

(4.2) Theorem : *(1) The mapping* ***yield*** *that maps a derivation tree of a fixed C-edNCE or S-HH grammar to the generated graph is (*,1)-definable.*

(2) A set of simple graphs is VR iff it is the image of a recognizable set of finite trees under a (,1)-definable transduction.*

(3) The class of VR sets of graphs is closed under (1,1)-definable transductions.

Proof : (1) See [Cou10] or [Eng2]; (2) The "if" part is proved in [Eng2] for a restricted notion of transduction; the full form is Theorem (3.2) of [CE]; an alternative proof is given in [Cou10]; (3) follows from (2) like do the corresponding assertions of Theorem (4.1). □

As consequences, one gets that the set of *line graphs* of the graphs of a HR set is VR, and that the set of *chordal graphs* (i.e., of graphs in which every cycle of length at least 4 has a chord) is not VR.

The *linear* VR sets of graphs, i.e., those defined by *linear* C-edNCE or S-HH grammars (with at most one nonterminal in each righthand side of a production) can be characterized as the images of recognizable languages under (*,1)-definable transductions, and their class (let us denote it by LIN-VR) is closed under (1,1)-definable transductions.

Every HR set of simple graphs is VR. Conditions ensuring that a VR set is HR are given in [CE] and in [Cou11]. In particular, a VR set is HR iff its members do not contain arbitrary large complete bipartite graphs as subgraphs, and this is decidable as a consequence of Theorem (6.1) below.

What about VR sets of simple *hypergraphs*? We *define* them as the images of recognizable sets of finite trees under (*,1)-definable transductions. Theorem (4.6) of [Cou10] gives a characterization in terms of systems of fixed point equations built with appropriate operations. Let us mention that S-HH grammars generate VR sets of simple hypergraphs, but not all of them ([CER]).

Definable transductions as a tool for studying hypergraphs as relational structures

Transductions of structures can be used in particular for two purposes: (1) in order to compare several representations of a same object by relational structures, and (2) to formalize encodings of hypergraphs by graphs, or more generally, of combinatorial objects by others. We illustrate successively these two uses.

We have defined two relational structures, $|G|_1$ and $|G|_2$ able to represent unambigously simple hypergraphs. For every set $\mathcal{C}$ of simple hypergraphs, we let $\mathbf{tr}(\mathcal{C})$ be the transduction: $\mathbf{tr}(\mathcal{C}) = \{(|G|_1, |G|_2) / G \in \mathcal{C}\}$. It is functional since we are dealing with simple hypergraphs and any two isomorphic graphs or structures are equal. If $\mathcal{C}$ is the set of simple graphs without sources, then $\mathbf{tr}(\mathcal{C})$ is not definable (otherwise, by Theorems (4.1) and (4.2), the classes of HR and VR sets of graphs would be the same which is not the case).

The *tree-width* of a graph is an integer that characterizes how close it is to be a tree: forests have tree-width 1, the tree-width of a clique with n vertices is n. The graphs in a HR set have tree-width bounded by some integer computable from the HR grammar generating it (see [Cou6, Cou12]).

(4.3) Theorem: *Let* k *be an integer. Let* $\mathcal{C}$ *be either the set of simple graphs of degree at most* k *or that of graphs of tree-width at most* k. *The transduction* $\mathbf{tr}(\mathcal{C})$ *is definable.*

Proof: See [Cou9] and [CE, Lemma 3.8]. □

This theorem has consequences relevant to the comparison between HR and VR sets of graphs.

(4.4) Corollary : *If a VR set of graphs has bounded degree, bounded tree-width, or is a subset of some HR set, then it is HR.*

Proof : Immediate application of Theorems (4.1), (4.2) and (4.3), and the fact that a HR set of graphs has bounded tree-width. The case of sets of graphs of bounded degree is also known from [EH2] and [Bra]. □

We now consider encodings of hypergraphs by graphs, along the lines of [ER1]. For H in $\mathbf{HG}(A)_0$, we denote by **gra**(H) the graph G such that:

$\mathbf{V}_G = \mathbf{V}_H \cup \mathbf{E}_H$ (recall that $\mathbf{V}_H \cap \mathbf{E}_H = \emptyset$),
$\mathbf{E}_G = \{(e,i,v) \ / \ e \in \mathbf{E}_H, v \in \mathbf{V}_H, v = \mathbf{vert}_H(e,i)\}$
$\cup \{(e,a) \ / \ e \in \mathbf{E}_H , \mathbf{lab}_H(e) = a \} \cup \{(v,*) \ / \ v \in \mathbf{V}_H \}$,
$\mathbf{vert}_G((e,i,v)) = (e,v)$ and $\mathbf{lab}_G((e,i,v)) = i$,
$\mathbf{vert}_G((e,a)) = (e)$ and $\mathbf{lab}_G((e,a)) = a$, and,
$\mathbf{vert}_G((v,*)) = (v)$ and $\mathbf{lab}_G((v,*)) = *$.

Thus, **gra**(H) belongs to $\mathbf{HG}(A \cup [m] \cup \{*\})_0$ where m is the maximum rank of an element of A. In this graph, the vertices represent the vertices of H as well as the edges of H. If $\mathbf{vert}_H(e) = (v_1,\ldots,v_n)$ then, in **gra**(H), there is an i-labeled edge from the vertex representing e to the one representing v_i. If $\mathbf{lab}_H(e) = b$, then the vertex representing e is incident with an edge of rank 1 having the label b in **gra**(H). The vertices of G representing vertices of H have the "new" label *. The following result from [CE] is a generalization of Theorem 1 of [ER1] and the given proof is independent of the original one. It shows that the C-edNCE grammars generate the same sets of hypergraphs (via the coding **gra**) as the HR grammars.

(4.5) Theorem: *(1) The transduction* **gra** *is (2,2)-definable and the transduction* $\mathbf{gra}^{-1}$ *is (1,2)-definable.*

(2) For every subset L *of* $\mathbf{HG}(A)_0$, **gra**(L) *is* VR *iff* L *is* HR

Proof: It is easy to verify that **gra** is (2,2)-definable. Hence, if L is HR, then **gra**(L) is HR, hence also VR. Conversely, the transduction $\mathbf{gra}^{-1}$ is (1,2)-definable. Hence, by Theorems (4.1) and (4.2), $\mathbf{gra}^{-1}(L')$ is HR if L' is VR. Hence, if **gra**(L) is VR then $L = \mathbf{gra}^{-1}(\mathbf{gra}(L))$ is HR. □

5 - Definable Transductions of Words and Trees.

We review some relations between definable transductions and classical notions in language theory. The following proposition shows that the class of definable transductions and that of rational transductions are incomparable.

(5.1) Proposition: *A rational transduction is definable iff the image of every word is a finite language.*

Proof: The "only if" part follows from Fact(2.1). Conversely, a rational transduction such that every word has a finite image is of the form $\lambda L.[h(k^{-1}(L) \cap K)]$ where K is a recognizable language, h and k are homomorphisms with k non erasing (i.e., the image of a letter is not the empty word): see [Ber, Exercise 7.2, p. 87]. It is thus definable by Propositions (2.3), (3.2) and (3.3). □

We have observed that **yield**, the transduction from a derivation tree relative some fixed context-free grammar to the generated word, is definable. Its inverse is not definable in general even if the grammar is unambigous (otherwise the set of Polish prefix notations of terms over a finite ranked alphabet would be the domain of a definable transduction and would be definable whence recognizable which is not the case).

For some (but not all) HR grammars Γ, there exists a (2,*)-definable transduction associating with every hypergraph G, a derivation tree of this graph relative to Γ whenever G belongs to the set generated by the Γ. Hence, for these grammars, the definable transduction **yield** has a definable inverse (see [Cou8]). However, these grammars, analogous to left-linear grammars, are less powerful than the general HR grammars.

The example used in the proof of Proposition (3.4) shows that the image of a context-free language under a definable transduction is not context-free. The following is more precise.

(5.2) Theorem: *(1) The following classes of languages are identical:*

(1.1) the sets of words defined by HR (or VR) grammars,

(1.2) the images of HR (resp. VR) sets by (2,)-definable (resp. (1,*)-definable) transductions from hypergraphs to words,*

(1.3) the output languages of deterministic tree-walking tree-to-word transducers.

(2) The following classes of languages are identical:

(2.1) the images of regular languages under definable transductions from words to words,

(2.2) the languages in LIN-HR (or in LIN-VR),

(2.3) the images of LIN-HR (resp. LIN-VR) sets by (2,)-definable (resp. (1,*)-definable) transductions, from hypergraphs to words*

(2.4) the output languages of deterministic 2-way gsm mappings.

Proof: That (1.1) = (1.2) follows from Theorems (4.1.3) and (4.2.3); that (1.1) = (1.3) is the main result of [EH1]; that (2.1) = (2.4) is proved in [EH1]; the other equalities follow from previous remarks.□

Open problems: What is the class of images of context-free languages under definable (word-to-word) transductions? It is in between classes (1) and (2) of Theorem (5.2), but is it *strictly in between*? It is closely related, perhaps identical, to the class of images of context-free languages under deterministic 2-way gsm mappings. This later class has been considered in [ERS] where it is proved (Corollary 4.10) that it equals the class of *context-free controlled ETOL systems of finite index*, also considered in [Lan]. It would also be interesting to have "machine" characterizations of definable word-to-word transductions (in terms of 2-way generalized sequential machines or of related devices) and of definable tree-to-word transductions.

6 - PARIKH'S THEOREM AND DEFINABLE TRANSDUCTIONS FROM GRAPHS TO COMMUTATIV WORDS.

Let $A = \{a_1,\ldots,a_n\}$ be an alphabet. A commutative word over A can be identified with the n-tuples of numbers of occurrences of letters $a_1,\ldots,a_n$ in the word or with the edgeless graph with one vertex labeled by a_i for each occurrence of a_i in the word and for each $i = 1,\ldots,n$. We shall denote by $A^{\$}$ the set of such words, tuples and graphs.

In Theorem (5.2), we have characterized the images of HR and VR sets of hypergraphs under definable transductions from hypergraphs to words. We do the same here for those from hypergraphs to commutative words.

A subset of $A^{\$}$ is *semilinear* if it is semilinear in the usual sense when considered as a set of tuples of numbers, i.e. if it is a finite union of sets of the form $\{w + \lambda_1 z_1 + \ldots + \lambda_k z_k \;/\; \lambda_1,\ldots, \lambda_k \in \mathbb{N}\}$ where $w, z_1,\ldots, z_k \in \mathbb{N}^n$.

From Corollary 3.2.7 of [ERS] saying that the commutative images of the languages of the class (1.3) of Theorem (5.2) are semilinear sets, we obtain the following result.

(6.1) Theorem: *The following classes of sets of commutative words are identical:*

(1) the class of semilinear sets

(2) the class of HR (or VR) sets of commutative words,

(3) the images of HR (resp. VR) sets under (2,)- (resp.(1,*)- definable transductions from hypergraphs to commutative words.*

A proof that does not rest on the constructions of [ERS] can be given with the help of the following lemma of independent interest. Let ϕ be a formula in $\mathfrak{F}(R,\{X_1, ..., X_k\})$ for some ranked set R of relational symbols. Let S be an R-relational structure. We define $\mathbf{sat}(\phi,S) := \{ (X_1,...., X_k) / (S, X_1,...., X_k) \models \phi \}$. With ϕ and a n×k matrix B of nonnegative integers, we can form the transduction $f_{\phi,B}$ from R-structures to $A^\$$ (identified with $\mathbb{N}^n$) such that:

$$f_{\phi,B}(S) := \{(\mathbf{Card}(X_1), ..., \mathbf{Card}(X_k)).B \ / \ (X_1,...., X_k) \in \mathbf{sat}(\phi,S)\}.$$

(6.2) Lemma: *A transduction from structures to commutative words is definable iff it is of the form* $f_{\phi,B}$ *for some* ϕ *and* B.

Theorem (6.1) provides us with an extension of Parikh's Theorem.

(6.3) Corollary: *Let* L *be a HR (resp. VR) set of hypergraphs. Let* L' *the associated set of structures (of type 2 or 1 respectively). Then* $f_{\phi,B}(L')$ *is semilinear.*

This result extends the version of Parikh's Theorem of [Hab] in the sense that it does not only count vertex or edge labels but cardinalities of sets satisfying MS formulas.

7 - Conclusion

Definable transductions form a quite powerful class of graph transformations, that is netherless manageable, as shown by the closure theorems we have stated above. We have no space to review algorithmic aspects. Let us only mention that

they form a key tool in [ALS], and that for every input graph G of tree-width at most some k, an output graph (relative to a fixed definable transduction) can be constructed in time **O(size**(G)) by [CM].

Can they be extended so as to include all inverse graph homomorphisms? The answer is no if we wish them to preserve the classes HR and VR, because, roughly speaking, the set of all finite graphs is neither HR nor VR. This is a striking difference with the case of words (the set of all words over a finite alphabet is context-free): there is no hope to generalize everything from words (or trees) to graphs.

ACKNOWLEDGEMENTS: I thank J. Engelfriet for pointing out the relevance of [ERS] and [Lan] to this work. I also thank A. Arnold and M. Mosbah for their comments on a first version of this paper.

REFERENCES

L.N.C.S. means Lecture Notes in Computer Science, Springer Verlag, Heidelberg, Berlin, New-York.

[ALS] ARNBORG S., LAGERGREN J.,SEESE D., Problems easy for tree-decomposable graphs, J. of Algorithms 12 (1991) 308-340

[BC] BAUDERON M., COURCELLE B., Graph expressions and graph rewritings, Mathematical Systems Theory 20 (1987) 83-127

[Ber] BERSTEL J., Transductions and context-free languages, Teubner Verlag, Stuttgart, 1979

[Bra] BRANDENBURG F.-J., The equivalence of boundary and confluent graph grammars on graph languages with bounded degree, L.N.C.S. 488 (1991)

[Bü] BÜCHI J., Weak second-order logic and finite automata, Z. Math. Logik Grundlagen Math. 5 (1960)66-92

[Cou1] COURCELLE B., Equivalences and transformations of regular systems. Applications to recursive program schemes and grammars, Theoret. Comput. Sci. 42 (1986) 1-122

[Cou2] COURCELLE B., An axiomatic definition of context-free rewriting and its application to NLC graph grammars, Theoret. Comput. Sci. 55 (1987) 141-181

[Cou3] COURCELLE B., Graph rewriting : An algebraic and logic approach, in "Handbook of Theoretical Computer Science, Volume B", J. Van Leeuwen ed., Elsevier,1990, pp.193-242

[Cou4] COURCELLE B., The monadic second-order logic of graphs I: Recognizable sets of finite graphs. Information and Computation 85 (1990) 12-75

[Cou5] COURCELLE B., The monadic second-order logic of graphs II: Infinite graphs of bounded width, Mathematical Systems Theory, 21(1989)187-221

[Cou6] COURCELLE B., The monadic second-order logic of graphs III: Tree-decompositions, minors and complexity issues, RAIRO Informatique Théorique et Applications, to appear.

[Cou7] COURCELLE B., The monadic second-order logic of graphs IV: Definability properties of equational graphs, Annals Pure Applied Logic 49(1990)193-255

[Cou8] COURCELLE B., The monadic second-order logic of graphs V: On closing the gap between definability and recognizability, Theoret. Comput. Sci. 80 (1991) 153-202

[Cou9] COURCELLE B., The monadic second order logic of graphs VI: On several representations of graphs by relational structures, Report 89-99, Discrete Applied Mathematics, to appear (see also Logic in Computer Science 1990, Philadelphia)

[Cou10] COURCELLE B., The monadic second order logic of graphs VII: Graphs as relational structures, Theoret. Comput. Sci., in press, Research Report 91-40, short version in the proceedings of the 4th International Workshop on Graph Grammars, L.N.C.S.532 (1991) 238-252

[Cou11] COURCELLE B., On the structure of context-free sets of graphs generated by vertex replacement, Research Report, Bordeaux-1 University, to appear.

[Cou12] COURCELLE B., Graph grammars, monadic second-order logic and the theory of graph minors, Proccedings of the Graph Minors Conference, Seattle, June 1991, Contemporary Mathematics, American Mathematical Society, to appear.

[CE] COURCELLE B., ENGELFRIET J., A logical characterization of the sets of hypergraphs generated by hyperedge replacement grammars, Research Report 91-41, Bordeaux-1 University, 1991, submitted.

[CER] COURCELLE B., ENGELFRIET J., ROZENBERG G., Handle-rewriting hypergraph grammars, Report 90-84, Bordeaux-I University, to appear in J.C.S.S.;short version in the proceedings of the 4th International Workshop on Graph Grammars, L.N.C.S. 532, (1991) 253-268

[CM] COURCELLE B., MOSBAH M., Monadic second-order evaluations on tree-decomposable graphs, Research report 90-110, Bordeaux-1 University, to appear in Theoret.Comput. Sci., (extended abstract in the Proceedings of WG'91, L.N.C.S., to appear.)

[DHLT] DAUCHET M., HEUILLARD T., LESCANNE P., TISON S., Decidability of the confluence of finite ground term rewrite systems and of other related term rewrite systems, Information and Computation 88 (1990) 187-201

[Don] DONER J., Tree acceptors and some of their applications, J.Comput.Syst.Sci.4 (1970) 406-451

[Elg] ELGOT C., Decision problems of finite automata design and related arithmetics, Trans. A.M.S. 98 (1961)21-52

[Eng1] ENGELFRIET J., Context-free NCE graph grammars, Proc. FCT' 89, L.N.C.S. 380 (1989) 148-161

[Eng2] ENGELFRIET J., A characterization of context-free NCE graph languages by monadic second-order logic on trees, L.N.C.S. 532 (1991) 311-327

[EH1] ENGELFRIET J., HEYKER L., The string generating power of context-free hypergraph grammars, J. Comp. Syst. Sci. 43 (1991) 328-360

[EH2] ENGELFRIET J., HEYKER L., Hypergraph languages of bounded degree, report 91-01, Univ. Leiden, 1991

[ER1] ENGELFRIET J., ROZENBERG G., A comparison of boundary graph grammars and context-free hypergraph grammars, Information and Computation 84 (1990) 163-206

[ER2] ENGELFRIET J., ROZENBERG G., Graph grammars based on node rewriting: an introduction to NLC graph grammars, L.N.C.S. 532 (1991) 12-23

[ERS] ENGELFRIET J., ROZENBERG G., SLUTZKI G., Tree transducers, L systems and two-way machines, J. Comput. System Sci. 20 (1980) 150-202

[GS] GECSEG F., STEINBY M., Tree automata, Akademiai Kiado, Budapest, 1984

[Gu] GUREVICH Y., Monadic second-order theories, in J. Barwise and S. Feferman eds.,"Model theoretic logic", Springer, Berlin, 1985, pp. 479-506

[Hab] HABEL A., Hyperedge replacement: grammars and languages, Doctoral dissertation, Bremen 1989

[HK] HABEL A., KREOWSKI H.J., May we introduce to you: Hyperedge replacement?, Proceedings of the 3rd International Workshop on Graph Grammars, L.N.C.S. 291 (1987) 15-26

[JR] JANSSENS D., ROZENBERG G., A survey of NLC grammars, L.N.C.S. 159 (1983) 114-128

[Lan] LANGE K.-J., Context-free controlled ETOL systems, Proceedings of 10th ICALP, L.N.C.S. 154 (1980) 723-733

[Rab] RABIN M., A simple method for undecidability proofs and some applications, in "Logic, Methodology and Philosophy of Science II", Y. Bar-Hilleled., North-Holland, Amsterdam, 1965, pp.58-68

[Rab] RAOULT J.-C., A survey of tree transductions, INRIA report 1410, to appear in the proceedings of an ASMICS workshop held in LeTouquet, France,June 1990, M. Nivat and A. Podelski eds.

[RW] ROZENBERG G., WELZL E., Boundary NLC grammars, Basic definitions, normal forms and complexity, Information and Control 69 (1986) 136-167

[Tho] THOMAS W., Automata on infinite objects, same volume as [Cou3] pp.133-192