DISCRETE MATHEMATICS 9 (1974) 61–70. © North-Holland Publishing Company

AN EXTRINSIC CHARACTERIZATION OF ADDRESSABLE DATA GRAPHS *

Arnold L. ROSENBERG
Mathematical Sciences Department, IBM T.J. Watson Research Center, Yorktown Heights, N.Y. 10598, USA

Received 20 September 1973 **

Abstract. Previous characterizations of the class of addressable data graphs have been intrinsic in nature. In this note, the auxiliary concept of a *monoid system* is used to derive an extrinsic characterization of the class. Specifically, a partial transformation of the class of data graphs is found which fixes (up to isomorphism) precisely the addressable data graphs.

1. Introduction

In earlier papers [1, 3], we found four properties, each of which characterizes the class of addressable data graphs. All of these characterizations are intrinsic in that they depend on uniformities in the structure of the data graph (such as rootedness, [1]) or on manipulations which the data graph admits (e.g., uniform self-insertability, [3]). This note is devoted to establishing an extrinsic characterization of the addressable data graphs. Using the auxiliary notion of a *monoid system*, we specify a partial function from the class of data graphs into itself: The addressable data graphs are precisely the fixed points (up to isomorphism) of this function.

2. Preliminaries

In this section we present the background on data graphs which is needed in the sequel, and we introduce monoid systems.

* This research was supported in part by ONR Contract N00014-69-C-0023.
** Original version received 7 September 1972.

2.1. *Data graphs and their morphisms*

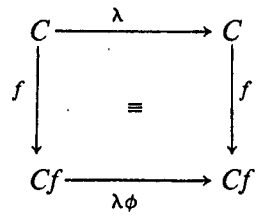
Definition 2.1.1. A *data graph* is a system $\Gamma = (C, \Lambda)$, where
(a) C is a countable set of *data cells*;
(b) Λ is a finite set of partial transformations of C, the *atomic link-transformations;*
subject to the condition [1]
(c) for all $c, d \in C$, there is a $\xi \in \Lambda^{\tau}$ such that $c\xi = d$.

One views the system (C, Λ) as specifying a labelled directed graph in the following manner. The graph has node set C. There is a directed edge from node $c \in C$ to node $d \in C$, labelled with $\lambda \in \Lambda$, precisely when $c\lambda = d$. Thus node c has a λ-edge leaving it precisely when $c \in \text{domain}(\lambda)$. Condition (c) above postulates the strong connectivity of data graphs.

Definition 2.1.2. Let $\Gamma = (C, \Lambda)$ and $\Gamma' = (C', \Lambda')$ be data graphs. Let $F = \langle f, \phi \rangle$ be a pair of total maps

$$f: C \to C',$$
$$\phi: \Lambda^{\tau} \to (\Lambda')^{\tau}.$$

F is a data graph *morphism* (in particular, a morphism of Γ) if
(a) ϕ is a monoid homomorphism;
(b) for each $\lambda \in \Lambda$, the following diagram commutes:



That is, for each $\lambda \in \Lambda$, the partial functions λf and $f(\lambda\phi)$ from C into Cf are simultaneously defined and agree on their common domain.

A data graph morphism $\langle f, \phi \rangle$ is a data graph *isomorphism* whenever both f and ϕ are injective.

[1] Throughout the paper, Λ^{τ} denotes the monoid generated by Λ under functional composition, with identity 1_C (the identity transformation on C).

A straightforward induction on the definition of data graph morphism establishes the well-foundedness of this notion. The reader can verify the following:

Lemma 2.1.3. *Let $F = \langle f, \phi \rangle$ be a morphism of $\Gamma = (C, \Lambda)$. The system $\Gamma F = (Cf, \Lambda\phi)$ is again a data graph.*

Data graph automorphisms were studied at length in [2] where they are called "symmetries".

2.2. *Addressable data graphs*

We now present those portions of the theory of addressable data graphs which are needed in the subsequent development.

Fix on a data graph $\Gamma = (C, \Lambda)$.

Definition 2.2.1. An *addressing scheme* for Γ is a total function $\alpha : C \to \Lambda^\tau$ such that

(a) there is a designated *base cell* $b \in C$ for which $b\alpha = 1_C$;

(b) for all $\lambda \in \Lambda$ and all $c \in \text{domain}(\lambda)$, $(c\lambda)\alpha = (c\alpha) \cdot \lambda$. [2]

Γ is *addressable* if it admits an addressing scheme.

Lemma 2.2.2 (Rosenberg [1]). *Let Γ admit the addressing scheme α.*

(a) *α is injective.*

(b) *The set $C\alpha$ is that submonoid of Λ^τ comprising all and only total functions.*

(c) *For all $c, d \in C$ and $\lambda \in \Lambda$, $(c\alpha) \cdot \lambda = d\alpha$ iff $c\lambda = d$.*

Definition 2.2.3. The cell $r \in C$ is a *root* of Γ if, for all $\xi, \eta \in \Lambda^\tau$ defined at r, $\xi = \eta$ whenever $r\xi = r\eta$.

Lemma 2.2.4 (Rosenberg [1]). *Every transformation defined at a root cell is one-one and total.*

[2] That is, the *address* $(c\lambda)\alpha$ of cell $c\lambda$ is the product in Λ^τ of $c\alpha \in \Lambda^\tau$ and $\lambda \in \Lambda$.

Definition 2.2.5. A *self-insertion* of Γ is a total function $\theta : C \to C$ satisfying: For all $\lambda \in \Lambda$ and all $d \in \text{domain}(\lambda)$, $(d\lambda)\theta = (d\theta)\lambda$.

Γ is *uniformly self-insertable* if there is a cell $a \in C$ which is arbitrarily relocatable in the following sense: For all $c \in C$, there is a self-insertion θ_c of Γ for which $a\theta_c = c$.

Theorem 2.2.6 (Rosenberg [1, 3]). *The following assertions about a data graph Γ are equivalent:*

(a) Γ *is addressable;*

(b) Γ *is rooted;*

(c) Γ *is uniformly self-insertable.*

Moreover, the base cell of each addressing scheme for Γ is a root cell; each root cell is arbitrarily relocatable; each relocatable cell is the base cell of a unique addressing scheme.

The following result, although intuitively obvious, is crucial later; hence we present a full proof.

Lemma 2.2.7. *Let $\Gamma = (C, \Lambda)$ be a data graph, and let $F = \langle f, \phi \rangle$ be an isomorphism of Γ. If Γ is addressable, then so also is ΓF.*

Proof. We show that the image of an arbitrarily relocatable cell in Γ is arbitrarily relocatable in ΓF, whence ΓF is uniformly self-insertable whenever Γ is.

Let $a \in C$ be arbitrarily relocatable. For each cell $c \in C$, let θ_c be the self-insertion of Γ satisfying $a\theta_c = c$. (One easily verifies the uniqueness of θ_c since any two self-insertions of Γ are either equal or disjoint [3].) For each cell $c \in C$, define the mapping θ_{cf}: $Cf \to Cf$ by

$$(df)\theta_{cf} = (d\theta_c)f \quad \text{for all } d \in C.$$

We verify that each θ_{cf} is a self-insertion of ΓF for which $(af)\theta_{cf} = cf$. Fix on an arbitrary $c \in C$.

(i) The mapping θ_{cf} is a function.

Obvious since F is an *iso*morphism (so $d = e$ whenever $df = ef$).

(ii) $(af)\theta_{cf} = cf$.

$(af)\theta_{cf} = (a\theta_c)f = cf$ by choice of a.

(iii) θ_{cf} is a self-insertion of ΓF.

Let $d \in C$ and $\lambda \in \Lambda$ be such that $(df)(\lambda\phi) \in Cf$; i.e., $df \in \text{domain}(\lambda\phi)$.

We then have

$$\begin{aligned}((df)(\lambda\phi))\theta_{cf} &= ((d\lambda)f)\theta_{cf} && \text{by definition of morphism;}\\ &= ((d\lambda)\theta_c)f && \text{by definition of } \theta_{cf};\\ &= ((d\theta_c)\lambda)f && \text{by definition of self-insertion;}\\ &= ((d\theta_c)f)(\lambda\phi) && \text{by definition of morphism;}\\ &= ((df)\theta_{cf})(\lambda\phi) && \text{by definition of } \theta_{cf}.\end{aligned}$$

Since c was arbitrary, the lemma is proved.

2.3. *Monoid systems*

The notion of a monoid system facilitates our proof of the main result.

Definition 2.3.1. (1) A *monoid system* is a system $\mathcal{M} = [M, S; G]$, where
(a) M is a finitely generated monoid; [3]
(b) S is a (not necessarily finitely generated) submonoid of M;
(c) G is a finite set of generators for M.
(2) The monoid system $\mathcal{M} = [M, S; G]$ is *transitive* if, for all $s, t \in S$, there exists a sequence $g_1, \ldots, g_n$, each $g_i \in G$, such that
(i) $sg_1 \ldots g_n = t$, and
(ii) for each $i \in \{1, \ldots, n\}$, $sg_1 \ldots g_i \in S$.

Conditions which imply the transitivity of a monoid system will be of interest in the next section. One such condition is immediate.

Lemma 2.3.2. *Let $\mathcal{M} = [M, S; G]$ be a monoid system. If S is a group generated by a subset of G, then $\mathcal{M}$ is transitive.*

3. Data graphs and monoid systems

Our main result requires the specification of a partial transformation of the class of data graphs. This transformation is derived by composition of a total map which associates a monoid system with each data graph (the DG–MS map) and a nontotal map which associates a data

[3] As usual, we denote the product of $a, b \in M$ by $a \cdot b$ or, when no confusion can arise, merely by ab.

graph with certain monoid systems, namely the transitive ones (the MS–DG map). The fixed points (up to isomorphism) of the composite map are precisely the addressable data graphs.

3.1. *The* DG–MS *map* $\mathcal{F}$

Let $\mathcal{F}$ be the function which associates with each data graph $\Gamma = (C, \Lambda)$ the monoid system

$$\Gamma\mathcal{F} = [\Lambda^\tau, \Lambda^T; \Lambda],$$

where Λ^T denotes the submonoid of Λ^τ comprising all and only total functions.

Theorem 3.1.1. *If* Γ *is an addressable data graph, then* $\Gamma\mathcal{F}$ *is a transitive monoid system.*

Proof. Say that $\Gamma = (C, \Lambda)$ admits the addressing scheme α. Then, by Lemma 2.2.2(b), $C\alpha = \Lambda^T$, so $\Gamma\mathcal{F} = [\Lambda^\tau, C\alpha; \Lambda]$. Therefore, given arbitrary $\xi, \eta \in \Lambda^T$, there exist $c, d \in C$ such that $\xi = c\alpha$ and $\eta = d\alpha$. Since Γ is strongly-connected, there is a $\zeta \in \Lambda^\tau$ satisfying $c\zeta = d$. An obvious induction on the definition of addressing scheme establishes the equation $(c\zeta)\alpha = (c\alpha)\cdot\zeta = d\alpha$. Since c and d were arbitrary, we have shown that the equation $\xi x = \eta$ is always solvable in Λ^τ for arbitrary $\xi, \eta \in C\alpha$. Since the prefix (in Λ^τ) of a total function is total, this solvability implies that the monoid system $\Gamma\mathcal{F}$ is transitive as was claimed.

Lemma 2.3.2 supplies numerous counterexamples to the converse of Theorem 3.1.1. In particular:

(1) Let $\Gamma = (C, \Lambda)$ be such that (i) $\# C \geqslant 2$, and (ii) each $\lambda \in \Lambda$ is nontotal. Then Γ is not addressable (by Lemma 2.2.4); but $\Gamma\mathcal{F}$ is transitive since $\Lambda^T = \{1_C\}$ is a group generated by a subset of Λ.

(2) Let $\Gamma = (C, \Lambda)$ be such that (i) each $\lambda \in \Lambda$ is a nonidentity self-inverse permutation of C, and (ii) each $c \in C$ is fixed by some $\lambda \in \Lambda$. Then Γ is not addressable since no cell can be a root (by (ii)); but Γ is transitive since $\Lambda^\tau = \Lambda^T$ is a group generated by Λ.

Examples. An example of (1) is $\Gamma_1 = (C_1, \Lambda_1)$, where

(a) $C_1 = \{0,1\}$;
(b) $\Lambda_1 = \{\lambda_1, \lambda_2\}$; $0\lambda_1 = 1$, $1\lambda_2 = 0$.
An example of (2) is $\Gamma_2 = (C_2, \Lambda_2)$, where
(a) $C_2 = \{0,1,2\}$;
(b) $\Lambda_2 = \{\lambda_1, \lambda_2, \lambda_3\}$; in cycle notation for permutations, $\lambda_1 = (0)(12)$, $\lambda_2 = (1)(02)$, $\lambda_3 = (2)(01)$.

The reader can easily verify that not all data graphs map under $\mathcal{F}$ onto transitive monoid systems. For instance, if Λ contains a total transformation λ which is not injective, the equation $\lambda x = 1_C$ is not solvable in Λ^*. We are left with the following open problem.

Open problem. What class of data graphs map (under $\mathcal{F}$) onto transitive monoid systems?

3.2. *The* MS–DG *map* $\mathcal{G}$

Let $\mathcal{G}$ be the function which associates with each monoid system $\mathcal{M} = [M, S; G]$ the system

$$\mathcal{M}\mathcal{G} = (S, \Omega_{\mathcal{M}}),$$

where $\Omega_{\mathcal{M}}$ comprises the following partial transformations of S:
(a) For each $g \in G$, $\Omega_{\mathcal{M}}$ contains the function $\omega_g : S \to S$ defined by

$$s\omega_g = \begin{cases} s \cdot g & \text{if } s \cdot g \in S, \\ \text{undefined} & \text{otherwise,} \end{cases}$$

for all $s \in S$.
(b) $\Omega_{\mathcal{M}}$ is exhausted by the functions described in (a).

Theorem 3.2.1. *The following assertions about a monoid system* $\mathcal{M}$ *are equivalent*:
(a) $\mathcal{M}$ *is transitive*;
(b) $\mathcal{M}\mathcal{G}$ *is an addressable data graph;*
(c) $\mathcal{M}\mathcal{G}$ *is a data graph*.

Proof. We prove the chain of implications (a)→(b)→(c)→(a). Fix on the monoid system $\mathcal{M} = [M, S; G]$..

(a)→(b) Since $\mathcal{M}$ is transitive, for all $s, t \in S$, there exist $g_1, \ldots, g_n \in G$ with $sg_1 \ldots g_i \in S$ and $sg_1 \ldots g_n = t$. Hence $s\omega_{g_1} \ldots \omega_{g_n} = t$ in $\mathcal{M}\mathcal{S}$, so $\mathcal{M}\mathcal{S}$ is a data graph. We prove that "cell" $e \in S$ (the identity in S) is arbitrary relocatable; the addressability of $\mathcal{M}\mathcal{S}$ then follows by Theorem 2.2.6.

For each $s \in S$, define the total map $\theta_s : S \to S$ by

$$t\theta_s = s \cdot t$$

for all $t \in S$. Clearly, θ_s is a total function. Moreover, $e\theta_s = s \cdot e = s$. Finally, let the transformation ω_g be defined at cell t; that is, say that $t \cdot g \in S$. We then have

$$(t\omega_g)\theta_s = (t \cdot g)\theta_s = s \cdot (t \cdot g) = (s \cdot t) \cdot g = (s \cdot t)\omega_g = (t\theta_s)\omega_g,$$

using associativity of monoid multiplication. Thus θ_s is a self-insertion of $\mathcal{M}\mathcal{S}$ which maps cell e onto cell s. Since s was arbitrary, the implication is established.

(b)→(c) Obvious.

(c)→(a) If $\mathcal{M}\mathcal{S}$ is a data graph, the strong-connectivity condition must obtain. That is, given arbitrary $s, t \in S$, there exist $\omega_{g_1}, \ldots, \omega_{g_n} \in \Omega_{\mathcal{M}}$ such that $s\omega_{g_1} \ldots \omega_{g_n} = t$. By definition of the ω_{g_i}, then, $sg_1 \ldots g_n = t$ and $sg_1 \ldots g_i \in S$ for each $i \in \{1, \ldots, n\}$. The proof is completed by noting that $s, t \in S$ were arbitrary.

Remark 3.2.2. Note that the transitivity of $\mathcal{M}$ in the proof that (a) implies (b) was used only to insure that the system $\mathcal{M}\mathcal{S}$ is a data graph; it was not used in establishing uniform self-insertability, which property holds for arbitrary $\mathcal{M}$. This suggests a generalization of our notions of data graph and addressability, which may be worth pursuing.

Remark 3.2.3. The connections between data graphs and monoid systems, which underlie Theorems 3.1.1 and 3.2.1, suggest the following. "Addressability" and/or uniform self-insertability of the generalized data graphs just alluded to may be an appropriate generalization of the relationship between trees as graphs and trees as given *via* tree domains (see, for instance, [4]). That is, the connections may delimit in a natural way the class of graphs which admit the type of algebraic presentation suggested by tree domains.

3.3. *The main result*

Let $\mathscr{H}$ be the partial transformation of the class of data graphs defined by: $\Gamma\mathscr{H} = \Gamma\mathcal{F}\mathcal{G}$ if $\Gamma\mathcal{F}\mathcal{G}$ is a data graph. It is this transformation which yields our main result.

Theorem 3.3.1. *A data graph* $\Gamma = (C, \Lambda)$ *is addressable if and only if* $\Gamma\mathscr{H}$ *is isomorphic to* Γ.

Proof. Assume first that Γ admits the addressing scheme α. Then

$$\Gamma\mathcal{F} = [\Lambda^{\tau}, C\alpha; \Lambda]$$

and

$$\Gamma\mathscr{H} = \Gamma\mathcal{F}\mathcal{G} = (C\alpha, \Omega_{\Gamma\mathcal{F}}).$$

We specify a pair of maps $f: C \to C\alpha$ and $\phi: \Lambda^{\tau} \to (\Omega_{\Gamma\mathcal{F}})^{\tau}$. We claim that $F = \langle f, \phi \rangle$ is a data graph isomorphism and that $\Gamma F = \Gamma\mathscr{H}$.

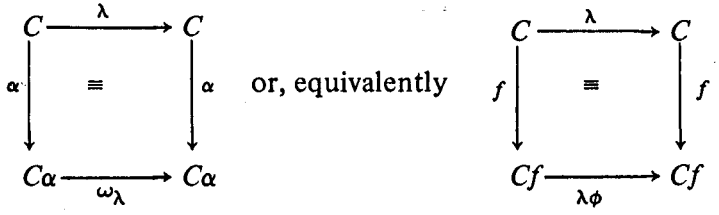
(1) Define f by $cf = c\alpha$ for all $c \in C$.

(2) Define ϕ inductively by

$$\begin{aligned} 1_C\phi &= 1_{C\alpha}; \\ \lambda\phi &= \omega_{\lambda} \qquad \text{for } \lambda \in \Lambda; \\ (\xi\cdot\eta)\phi &= (\xi\phi)\cdot(\eta\phi) \quad \text{for } \xi, \eta \in \Lambda^{\tau}. \end{aligned}$$

(i) By Lemma 2.2.2(a), f maps C one-one onto $C\alpha$.

(ii) By Lemma 2.2.2(c), the following equivalent diagrams commute for each $\lambda \in \Lambda$:



In particular, $(\text{domain}(\lambda))\alpha = \text{domain}(\omega_{\lambda})$.

(iii) It follows from (ii), by induction, that ϕ is a monoid isomorphism (i.e., that ϕ is a function and is injective). To wit, consider $\lambda_1 \ldots \lambda_n$ and $\mu_1 \ldots \mu_m$, each $\lambda_i, \mu_j \in \Lambda$. For arbitrary $c \in \text{domain}(\lambda_1 \ldots \lambda_n) \cap \text{domain}(\mu_1 \ldots \mu_m)$, we have

$$(c\lambda_1 \dots \lambda_n)\alpha = (c\alpha)\cdot\lambda_1 \dots \lambda_n = (c\alpha)\omega_{\lambda_1} \dots \omega_{\lambda_n}$$

and

$$(c\mu_1 \dots \mu_m)\alpha = (c\alpha)\cdot\mu_1 \dots \mu_m = (c\alpha)\omega_{\mu_1} \dots \omega_{\mu_m}.$$

Thus, $\lambda_1 \dots \lambda_n = \mu_1 \dots \mu_m$ iff $\omega_{\lambda_1} \dots \omega_{\lambda_n} = \omega_{\mu_1} \dots \omega_{\mu_m}$ since α is a function and is injective.

Thus F is a data graph isomorphism, so that Γ is isomorphic to $\Gamma\mathcal{H}$.

Assume conversely that $\Gamma\mathcal{H}$ is isomorphic to Γ. Then $\Gamma\mathcal{H}$ is, perforce, a data graph. Moreover, by Theorem 3.2.1, $\Gamma\mathcal{H}$ is addressable – since it is $(\Gamma\mathcal{F})\mathcal{G}$ and it is a data graph. Hence Γ is isomorphic to an addressable data graph and is, therefore, itself addressable by Lemma 2.2.7.

The theorem is proved.

Acknowledgment

It is a pleasure to acknowledge several stimulating and helpful conversations with Drs. H.R. Strong and J.W. Thatcher.

References

[1] A.L. Rosenberg, Data graphs and addressing schemes, J. Comput. System Sci. 5 (1971) 193–238.
[2] A.L. Rosenberg, Symmetries in data graphs, SIAM J. Comput. 1 (1972) 40–65.
[3] A.L. Rosenberg, Addressable data graphs, J. Assoc. Comput. Mach. 19 (1972) 309–340.
[4] J.W. Thatcher, Characterizing derivation trees of context-free grammars through a generalization of finite automata theory, J. Comput. System Sci. 1 (1967) 317–322.