ソフトウェア基礎論 35－7
(1990. 5. 24)

グラフ書き換え系と網信頼性解析への応用

On Graph Rewriting Systems and Application to Network Reliability Analysis

岡田 康義 (NTT 基礎研) 林 正博 (NTT 網総研)
Y. Okada and M. Hayashi NTT

Abstract

A new theory, Graph Rewriting Systems is proposed for 'graph reduction' that is very similar to Term Rewriting Systems. By using this theory a serious reduction problem in network reliability analysis can be solved. This problem stems from the fact that graph reduction result often are extremely different , even though all of them from the same original graph. The Term Rewriting Systems theory has a keyword , called the Church Rosser (or sometimes "Confluence") property that has originally appeared in lambda calculus and can produce the same final reduction result from a 'term' under termination.

Using this property, sufficient information can be obtained to solve a reduction problem. The information is based on "critical pairs" obtained by superpositioning the two reduction rules and preserves the Church Rosser property. This property can be used for graph reduction in network reliability analysis, since it gives an orientation to control the mechanics of the graph reduction methods.

1 Introduction

Network reliability analysis is of major importance to computer communications and power networks. Even the simplest models often lead to computational problems that are NP-hard for general networks [2], [12].

The network model used in this paper is an undirected graph $G = (V, E)$ whose lines can fail independently of each other according to known probabilities. In 1977 Rosenthal & Frisque [13] presented a method which reduced network size by transforming three-terminal subnetworks into $Y-shaped$ networks called $\Delta - Y$.

The terminal degrees were thus reduced and possible series combinations were created capable of being reduced to a single line. This transformation method, called the "reduction" method, preserves the original network probabilities between terminals.

In 1985 Satyanarayana & Politof [11] showed that a class of series-parallel networks, for which only exponentially complex algorithms were previously known, can be analyzed in polynomial time by a reduction method. However, reduction methods cause a serious problem. If the application rules and places are correctly selected a simple graph can be obtained, but if the selection is incorrect the resultant graph size is still large. This makes it very hard to compute network reliability due to the size of the graph.

On the other hand Term Rewriting Systems have the Church Rosser property [1], [6], [7], [10] as reduction rules capable of preserving any term reduced to as a unique reduction result for every reduction step at reduction termination. This concept of Church Rosser property was extended to the Abstract Reduction Systems [7],

If you can introduce this Church Rosser property into graph reduction, we can get a simple graph without complex reduction strategy. Therefore this paper proposes a new theory, Graph Rewriting Systems, which contains the Church Rosser property as well as Term Rewriting Systems. It should also be noted that the other kinds of graph rewriting systems exist, (3) and (9), which treat graph as objects having various meanings, but this theory gives sufficient information to solve the reduction problem.

To create this theory a new concept, boundary points, had to be introduced for the graph reduction methods since it is needed when the overlapping between any two rules is well defined. The introduction of this boundary concept into Graph Rewriting Systems, whether the system has a Church Rosser Property or not, is decided by checking the convergence of the critical pairs obtained by the 'superposition' of all possible rule pairs.

2 A problem of Network Reliability Analysis

In this section the fundamental definitions of graph theory will be presented and a problem of network reliability analysis pointed out. Harary's notation [4] for graph theory is followed in this section. It also defines the problem of network reliability analysis as "Why are graph reduction results extremely different, even though all of them from the same orignal graph ?"

The relationship between graph reduction rules and this problem and its cause will be presented later. Furthermore, this section shows that the critical pairs in graph reduction produce sufficient information about the topology of the final reduction graphs.

2.1 Fundamental Definitions in Graph Theory

Definition 1 Graph and Subgraph : A *graph*, G, consists of a finite nonempty set V of points together with a prescribed multiset, E, of unordered pairs of distinct points of V. G is *labeled* when the points are distinguished from each other by names such as $v_1, v_2, ..., v_p$. Each pair, $x = \{u,\ v\}$, of points in E is a line of G, and x, said to be *join* u and v and u and v are *adjacent points* (sometimes denoted u adj v). Point u and lines x are *incident* with each other, as are v and x. If two distinct lines, x and y, are incident with a common point, then they are *adjacent* lines. A *subgraph* of G is a graph having all of its points and lines in G.

Definition 2 Walk and Path : A *walk* of graph G is an alternating sequence of points and lines $v_0, x_1, v_1, ..., v_{n-1}, x_n, v_n$ beginning and ending with points for which each resultant line is incident with two points immediately preceding and following it. The walk is *closed* if $v_0 = v_n$. It is a *path* if all the points (and thus necessarily all the lines) are distinct. If the walk is closed, then it is a *cycle*, provided its n points are distinct and $n \geq 3$.

Definition 3 Connected : A graph is *connected* if every pair of points is joined by a path.

Definition 4 Degree of Point : The degree of a point v_j in graph G, denoted d_i or deg v_i, is the number of lines incident with v_i.

Definition 5 Isomorphic Graphs : Two graphs, G and H, are *isomorphic* if there exists a one-to-one correspondence between their point sets which preserves adjacency.

2.2 Graph Reduction Rules and the Problem of Reduction Strategy in Network Reliability Analysis

Network reliability analysis is of major importance in computer, communication and power networks. The network model used in this paper is graph, $G = (V,\ E)$, whose lines can fail independently each other according to known probabilities. The problem of reliability analysis is how to determine the probability that a specified set of vertices, $K \subseteq V$, will remain connected, i.e., the K -terminal reliability of G.

In conventional reliability analysis, reliability-preserving graph reduction methods are known efficient algorithms [11], [14].

In 1985 Satyanarayana & Wood [14] showed that in the reduction method the way application ordering is chosen and where it is applied to a graph is independent of probability considerations. Our main purpose has been to research graph reduction methods since 1985. The three fundamental reduction rule set [14] is shown below in Fig. 1.

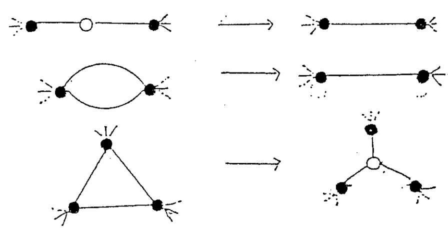


Fig. 1. Fundamental graph reduction rule set

In Fig. 1:

(1) *series reduction*. Suppose that $e_a = (u,\ v)$ and $e_b = (v,\ w)$ with $deg(v) = 2$ are series lines in graph G. A series reduction replaces e_a and e_b with single line $e_c = (u,\ w)$.

(2) *parallel reduction*. Suppose that $e_a = (u,\ v)$ and $e_b = (u,\ v)$ are two parallel lines in graph G. A parallel reduction replaces e_a and e_b with a single line $e_c = (u,\ v)$.

(3) $\Delta - Y$ *reduction*. Suppose $e_a = (u_1,\ u_2)$, $e_b = (u_1,\ u_3)$ and $e_c = (u_2,\ u_3)$ are three lines of Δ. $\Delta - Y$ reduction replaces e_a, e_b, e_c with Y (See Fig. 1).

A lot of graphs are reducible by these rules. One such example is shown below in Fig. 2.

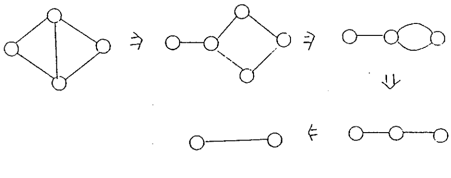


Fig. 2. Graph reduction by the fundamental reduction rule set

When a graph is recursively reduced into a simpler one, a serious trouble occurs when the original graph result in different final forms.

One of the main reasons is that the reduction results depend on where a rule is applied on a graph and the rule application ordering which is applied to it. A typical example is shown below in Fig. 3.

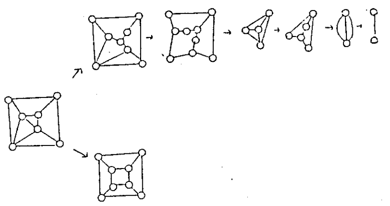


Fig. 3. Example of different reduction results

By using the three reduction rules, the results for a given graph will differ from Fig. 3, because it is only a typical example of the problem. If the correct direction is selected (↗ in Fig. 3), only two terminal points can be reached.

If an incorrect direction is selected (↘ in Fig. 3), a complex form results. Differing final results often occur with different reduction sequences. Therefore sufficient information must be give to solve this problem.

The production of differing results is a typical problem in *ARS* [7]. If in *ARS* a system has the Church Rosser property it produces a unique reduction result upon termination. We tried to introduce this property into graph reduction and succeeded in creating a new theory, Graph Rewriting Systems. In this new theory, the convergence of critical pairs will solve the ramification problems.

3 Introduction to Graph Rewriting Systems and Their Various Properties

3.1 Abstract Reduction Systems

Definition 6 Abstract Reduction Systems : An *Abstract Reduction System* (*ARS*) is a structure $\mathcal{A} = \langle A,\ \longrightarrow_\alpha \rangle$ consisting of a set A and a sequence of binary relations $\longrightarrow$ on A, also called reduction or rewrite relations. If for $a, b \in A$ we have $(a,\ b) \in \longrightarrow_\alpha$, we write a $\longrightarrow_\alpha b$ and call b a one-step ($\alpha - reduct$) of a. This is sometimes briefly described as $\mathcal{A} = \langle A,\ \longrightarrow \rangle$.

Definition 7 Transitive-reflexive Closure :
The transitive-reflexive closure of $\longrightarrow_\alpha$ is denoted by $\stackrel{*}{\longrightarrow}_\alpha$. So $a \longrightarrow_\alpha b$, if there is a possibly empty, finite sequence of 'reduction steps' $a \equiv a_0 \longrightarrow_\alpha a_1 \longrightarrow_\alpha .. \equiv \longrightarrow_\alpha a_n \equiv b$. Here $\equiv$ denotes the identity of the elements of A. Element b is called an $\alpha-$ reduct of a. The equivalence relation generated by $\longrightarrow_\alpha$ is $=_\alpha$, which is also d the *convertibility* relation generated by $\longrightarrow_\alpha$. The reflexive closure of $\longrightarrow_\alpha$ is $\longleftarrow_\alpha^{\equiv}$. The converse relation of $\longrightarrow_\alpha$ is $\longleftarrow_\alpha$ or $\longrightarrow_{\alpha^{-1}}$. The union $\longrightarrow_\alpha \cup \longrightarrow_\beta$ is denoted by $\longrightarrow_{\alpha\beta}$. The composition $\longrightarrow_\alpha \circ \longrightarrow_\beta$ is defined by $a \longrightarrow_\alpha \circ \longrightarrow_\beta b$, if $a \longrightarrow_\alpha c \longrightarrow_\beta b$ for some $c \in A$.

The next definition is needed to preserve stop in the finite reduction step.

Definition 8 Normal form and Normalizing in *ARS* **:**
Let $\mathcal{A} = \langle A,\ \longrightarrow_\alpha \rangle$

(1) $a \in A$ is a *normal form*, if there is no $b \in A$ such that $a \longrightarrow_\alpha b$.

$b \in A$ *has a normal form* if there is a normal form a such that $b \stackrel{*}{\longrightarrow}_\alpha a$.

(2) α is *strongly normalizing* (*SN*) if there are no infinite reduction sequences $a_0 \longrightarrow a_1 \longrightarrow a_2 ....$

Weakly Church Rosser and Church Rosser are essential properties in this paper and they are defined as follows:

Definition 9 Weakly Church Rosser and Church Rosser on *ARS* **:**
$\mathcal{A} = \langle A,\ \longrightarrow_\alpha \rangle$.

(1) The reduction relation of $\longrightarrow_\alpha$ is *weakly confluent* or *weakly Church−Rosser*(*WCR*), if for all $a, b, c \in A$ with $a \longrightarrow_\alpha b$ and $a \longrightarrow_\alpha c$ we can find $d \in A$ such that $b \stackrel{*}{\longrightarrow}_\alpha d$, $c \stackrel{*}{\longrightarrow}_\alpha d$. This d is called the *common reduct* of b and c

(2) $\longrightarrow_\alpha$ is *confluent* or has the *Church – Rosser(CR)* property (see Fig. 4), if for all $a, b, c \in A$ with $a \overset{*}{\longrightarrow}_\alpha b$ and $a \overset{*}{\longrightarrow}_\alpha c$ we can find $d \in A$ such that $b \overset{*}{\longrightarrow}_\alpha d$, $c \overset{*}{\longrightarrow}_\alpha d$.

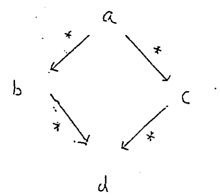


Fig. 4. Church Rosser Property

Note that difference between weakly Church Rosser and Church Rosser. The former has a common after only one reduction in each direction. The latter has one after any reduction even in steps. The concept Church Rosser originally came from lambda calculus [1], [6] and In *TRS* there are a lot of systems [7], [8], [15], [16] containing it.

The following facts are well known about the relation between the Church Rosser property and weakly Church Rosser.

Newman's lemma plays a very important role on *ARS*. Because under *SN* the completeness of $\longrightarrow_\alpha$ can only be tested by checking weakly Church Rosser.

Lemma .1 (Newman) *Let* $\mathcal{A} = \langle A, \longrightarrow_\alpha \rangle$ *be a ARS. If* $\longrightarrow_\alpha$ *is WCR and SN then* $\longrightarrow_\alpha$ *is CR.*

This proof originally came from Newman [10] and the simple proof came from Hue [7]. By the lemma it follows that under $SN \longrightarrow_\alpha$ is WCR, iff $\longrightarrow_\alpha$ is CR.

Proposition 1 *Let* $\longrightarrow_\alpha$ *be a confluent ARS. Then* $\longrightarrow_\alpha$ *has a unique normal form; i.e., if* $n_1 =_\alpha n_2$ *and* n_1, n_2 *are normal forms then* $n_1 \equiv n_2$.

Definition 10 Complete System : $\longrightarrow_\alpha$ with properties *SN* and *CR* is called *complete.*

3.2 Fundamental Definitions in Graph Rewriting Systems and Properties

This section introduces Graph Rewriting Systems and gives their related properties.

Definition 11 Boundary on Graph and Compliment Graph : Let graph G be a connected graph and subgraph g of G. A point in G is called a *natural boundary* for subgraph g, if it is incident with a line not in g. A set of all boundary points in subgraph g of G is denoted by $B_n(g)$. Graph g^c, *in* which all points and lines have been removed from G except for the boundary points in subgraph g, is called the *compliment graph* of g and denoted by g, as shown in Fig. 5.

Natural boundary points and compliment graphs play important roles in GRS. Note that this notation is quite different from Harary's definition [4], since we simply refer to "graph" and "subgraph" instead of "connected graph" and "connected subgraph." henceforth.

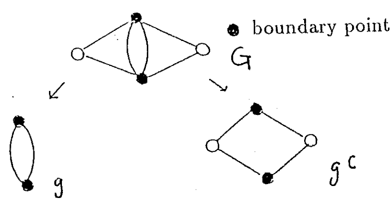


Fig. 5. g , g^c and boundary points

This boundary conception can be extended as follows. To simultaneously handle conventional and special graphs occurring on both sides of the rules representing transformations between graphs.

Definition 12 Extended Concept of Boundary on Graph : Let graph G be a connected graph. Let specially marked (usually black) points in G be defined as *artificial boundary* points, which means indefinite (at least one) lines not containing G. We refer to this as boundary, not artificial boundary.

Definition 13 Boundary Graph : a labeled graph, g, with boundary points is called a *boundary* graph. A set of boundary points in s is denoted by $B(s)$. Points which are not a boundary in g are called *interior* points. A set of interior points in s is denoted by $I(s)$. Graphs having only interior points can also be regarded as boundary graphs without boundary points.

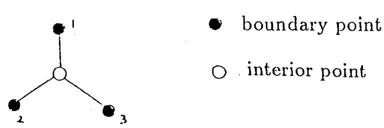


Fig. 6. Boundary graph.

Definition 14 Isomorphic between Boundary Graph : Two boundary graphs G and H are *boundary isomorphic* if there exists a one-to-one correspondence between their point sets which preserve adjacency and labels of boundary points.

Intuitively speaking two boundary isomorphic graphs have the same boundary points preserving the correspondence between them.

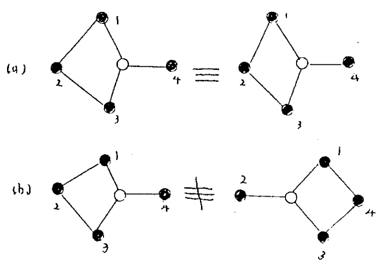


Fig. 7. Isomorphic boundary graph (a) and nonisomorphic boundary graph (b)

Definition 15 Reduction Rule on Graph : $\mathcal{R}$ is a set of pairs $(s,\ t)$ with s, t of the boundary graph subject to the following constraints.

- left side s and t are not a boundary graph, but a single boundary point.
- boundary points are invariant; i.e., if boundary nodes exist in the left hand side, they also exist in the right hand side. The converse must also be true.

Pairs $(s,\ t)$ are called *rewriting rules* and are written as $s \longrightarrow t$ henceforth. Example of the reduction rules are shown in Fig. 8. Black points with labels represent boundary points and white points without labels represent interior points.

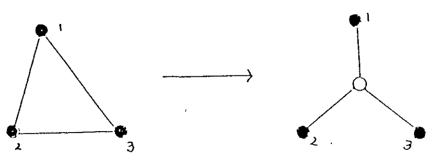


Fig. 8. reduction rules.

Rewriting systems are defined as graphs in the same manner as TRS, which is useful for network reliability analysis. A lot of notations used are quite different from other systems, [3], [9].

Definition 16 Graph Rewriting Systems : A *graph rewriting system* is a pair, $(\mathcal{G},\ R)$. Here, $\mathcal{G}$ is a family consisting of boundary graphs. $\mathcal{R}$ is a set of pairs, $(s,\ t)$. Syntactical identity $\equiv$ is given by a boundary isomorphic map between the boundary graphs.

We usually write $\mathcal{R}$ instead of $(\mathcal{G},\ R)$ and $r : s \longrightarrow t$ or $t \longrightarrow s$.

Definition 17 Imbedding : Let s (not necessarily connected) and G be a boundary graph. Mapping σ from s to G is called *imbedding* from s to G, if σ is an isomorphic map from s to a subgraph denoted by s_σ in G, subject to the following constraints:

For every point v in s

(1) if $v \in I(s)$, then $\sigma(v) \in I(s)$ with $deg(v) = deg(\sigma(v))$.

(2) if $v \in B(s)$, then $\sigma(v) \in B_n(\sigma(s)) \cup B(\sigma(s))$.

We denote this as $G \equiv s_\sigma * {s_\sigma}^c$ henceforth.

An example of imbedding is shown below in Fig. 9.

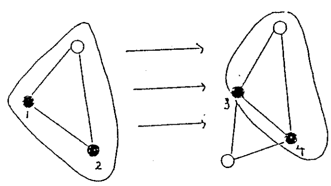
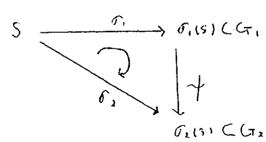


Fig. 9. Imbedding

Proposition 2 *Let s , G_1 , G_2 be a boundary graph such that two imbeddings are $\sigma_1 : s \longrightarrow G_1$ and $\sigma_2 : s \longrightarrow G_2$ with ${\sigma_1}_{|B(s)} = {\sigma_2}_{|B(s)}$.*

(1) There is a boundary isomorphic map $\psi : \sigma_1(s) \longrightarrow \sigma_2(s)$ with $\psi \circ \sigma_1 = \sigma_2$ that insures the following diagram, where $\circ$ is a composite of the maps.



(2) if $G \equiv \sigma_1(s) * \sigma_1(s)^c$ and $G' \equiv \sigma_2(s) * \sigma_2(s)^c$ with $\sigma_1(s)^c \equiv \sigma_2(s)^c$, then $G \equiv G'$.

Proof.

(1) Since σ_1 is imbedding $s \longrightarrow t$, inverse map $\sigma_1{}^{-1}$: $\sigma_1(s) \longrightarrow s$ exists. Let ψ be $\sigma_2 \circ \sigma_1{}^{-1}$ by (1), then ψ is an isomorphic map.

Also if $v \in \sigma_1(s)$, then $\sigma_1{}^{-1}(v) \in B(s)$ from the definition and the following conclusions can be made.

$\psi(v)$
$= (\sigma_2 \circ \sigma_1{}^{-1})(v)$
$= (\sigma_2(\sigma^{-1}(v)))$
$= (\sigma_1(\sigma_1{}^{-1}(v)))$ (by $\sigma_1|_{B(s)} = \sigma_2|_{B(s)}$ and $\sigma_1(s) \in B(s)$)
$= v$.

(2) By $\sigma_1|_{B(s)} = \sigma_2|_{B(s)}$ and $\sigma_1(s)^c \equiv \sigma_2(s)^c$,
G
$\equiv \sigma_1(s) * \sigma_1(s)^c$
$\equiv \sigma_1(s) * \sigma_2(s)^c$ (by $\sigma_1(s)^c \equiv \sigma_2(s)^c$)
$\equiv \sigma_2(s) * \sigma_2(s)^c$ (by $\sigma_1|_{B(s)} \equiv \sigma_2|_{B(s)}$)
G'

Thus, ψ is a boundary isomorphic map.

Definition 18 Reduced Graph : Let $r : s \longrightarrow t$ be a rewriting rule. and $\sigma : s \longrightarrow G$ be an imbedding, which means $G \equiv s_\sigma * s_\sigma{}^c$. G' is a reduced graph for G using r and imbedding σ if $G' \equiv t_{\sigma_r} * t_{\sigma_r}{}^c$, $s_\sigma{}^c \equiv t_{\sigma_r}{}^c$ and $\sigma|B(s) = \sigma_r$ for rewriting rule $r : s \longrightarrow t$, imbedding map σ from s to G, where t_{σ_r}, is a graph obtained by replacing s_σ with rule r.

Intuitively speaking, the G' obtained from G by replacing s_σ with rule r preserves the compliment graph of s_σ.

For the above reduced graph we have the following properties.

Proposition 3 *Let $r : s \longrightarrow t$ be a rewriting rule, $\sigma : s \longrightarrow G$ be an imbedding and G' be a reduced graph for G using r and an imbedding σ, if we have the following properties.*

(1) By reducing G to G', $B(G)$ is invariant, i.e., $B(G) = B(G')$.

(2) Reduced graph G' for G using r and imbedding σ is uniquely determined.

Proof.

(1) If boundary point v is in $s_\sigma{}^c$ or $v \in B(G')$, then $v \in B(G')$ is trivially obtained from definition of the rewriting systems. If boundary point v in s_σ, then $\sigma^{-1}(v) \in B(s) = B(t)$. Hence $v \in B(t)$ from the definition of imbedding. In two case we have $B(G) \subset B(G')$. Similarly, the converse is obtained; i.e., $B(G) \supset B(G')$. Hence $B(G) = B(G')$.

(2) By superposition $G \equiv s_\sigma * s_\sigma{}^c$, $G' \equiv t_{\sigma_r} * t_{\sigma_r}{}^c$, and $t_\sigma{}^c \equiv t_{\sigma_r}{}^c$. Suppose graph G'' is also reduced from G using r and imbedding $\sigma' : t \longrightarrow G''$, then $G'' \equiv t_{\sigma'_r} * t_{\sigma'_r}{}^c$, and $t_{\sigma'_r}{}^c \equiv t_{\sigma_r}{}^c$ from the definition. This concludes $G' \equiv G''$ from proposition 2.

The definition of the next reduction step is well defined by this proposition.

Definition 19 Reduction Step : The rewriting rules of *GRS* $\mathcal{R}$ give rise to the *reduction step*, such that : $G \longrightarrow_{\mathcal{R}} G'$ iff G' is a graph reduced for G using r and imbedding $\sigma : s \longrightarrow G$.

In that case subgraph s_σ of G is called a *redex*. To specify the rewriting rules used in this reduction step, we write $G \xrightarrow{r}_{\mathcal{R}} G'$.

Definition 20 Superposition : Let $r_1 : \alpha \longrightarrow \beta$ and $r_2 : \gamma \longrightarrow \delta$ be two rewriting rules. We have superposition s of two rules, where s is a boundary graph satisfied with the following conditions:
Let subgraph g (not necessarily connected) be in α and imbedding $f : g \longrightarrow \gamma$, where g is a subgraph that does not have only boundary graphs. By imbedding $f : g \longrightarrow \gamma$, g and $f(g)$ are identified with each other as follows: for point v in g we consider

(1) v and $f(v)$ as interior points, if $v \in I(g)$ and $f(v) \in I(f(g))$ with $deg(v) = deg(f(v))$ or if $v \in B(g)$ and $f(v) \in I(f(g))$

or v and $f(v)$ as boundary points, if $v \in B(g)$ and $f(g) \in B(f(g))$

(2) By identifying g with $f(g)$, we naturally have two natural inclusions to map σ_1 from α to itself in s and σ_2 and from γ to itself in s simply as graphs , therefore σ_1 and σ_2 must be imbedding.

One superposition of the two rules is shown in Fig. 10.

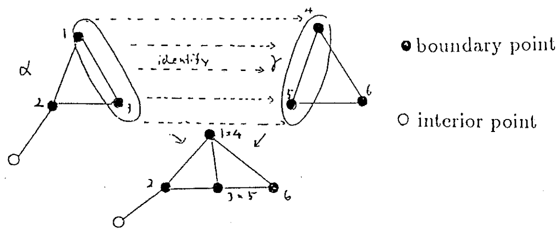


Fig. 10. Superposition

If we represent superposition s of r_1 on r_2, $\alpha_{\sigma_1} * \alpha^c{}_{\sigma_1} \equiv \gamma_{\sigma_2} * \gamma^c{}_{\sigma_2}$ where two imbeddings $\sigma_1 : \alpha \longrightarrow s$ and $\sigma_2 : \gamma \longrightarrow s$ are natural inclusions into superposition s, then it can be reduced two possible ways: $\alpha_{\sigma_1} * \alpha^c{}_{\sigma_1} \longrightarrow \beta_{\sigma_1} * \alpha^c{}_{\sigma_1}$ and $\gamma_{\sigma_2} * \gamma^c{}_{\sigma_1}. \longrightarrow \delta_{\sigma_2} * \gamma^c{}_{\sigma_1}$.

Definition 21 Two Rules Overlapping and Critical Pairs in *GRS* :
The pair of reducts $\langle \beta_{\sigma_1} * \alpha^c{}_{\sigma_1}, \ \delta_{\sigma_2} * \gamma^c{}_{\sigma_2} \rangle$ called a *critical pair* is obtained by the *superposition* of $\alpha \longrightarrow \beta$ and $\gamma \longrightarrow \delta$. Two rules are called overlapping if they produce a superposition. In this case s is called the overlapping redex. Otherwise, it is called a nonoverlapping redex. If $\alpha \longrightarrow \beta$ and $\gamma \longrightarrow \delta$ are the same rewriting rule, we furthermore require that s is required not to be identical to $\gamma \equiv \alpha$.

Intuitively speaking, an overlapping rule can reduce every graph to which the two rules apply. This causes critical pairs to have a a key word for each different graph reduction results. This definition is quite different from *TRS*. Such pair is shown below in Fig. 11.

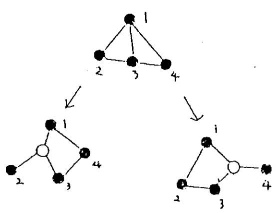


Fig. 11. Example of critical pairs

Definition 22 Convergence of Critical Pair : Critical pair $< s, \ t >$ is called *convergent* if s and t have a common reduct; i.e. , a boundary isomorphic map exists between s and t.

For the above critical pairs we have the following properties.

Lemma .2 Critical Lemma in *GRS* : *GRS $\mathcal{R}$ is WCR iff all critical pairs are convergent.*

Proof. Separate the redexes into nonoverlapping and overlapping cases. In the nonoverlapping redex case two reduction rules are independently applicable to the redex and do not effect each other. Thus, any two redexes separated from any redex will obtain the same final graph after both rules are applied and $\mathcal{R}$ is *WCR*. In the overlapping case, *WCR* is evidently concluded by hypothesis. Thus, the lemma is proved. Conversely, if $\mathcal{R}$ is *WCR* the right hand side assertion is obvious from the assumption.

Theorem 1 *Let $\mathcal{R}$ be a GRS which is SN then $\mathcal{R}$ is CR iff all critical pairs of $\mathcal{R}$ are convergent.*

proof. This proof is trivially obtained from Lemma .1 and Lemma .2.

4 Application to Network Reliability Analysis

This section explains how *GRS* is applied to network reliability analysis. It also give a solution for reduction methods used by *all − terminal reliability* [11].

4.1 Relations between Reduction Properties in *GRS* and Reduction Methods in Network Reliability Analysis

The problems of reduction methods in network reliability analysis can be solved using *GRS*, which gives simple principles with respect to these methods.

4.1.1 Three Reduction Rules in Network Reliability Analysis

The fundamental rule set used in network reliability analysis is shown below in Fig. 12.

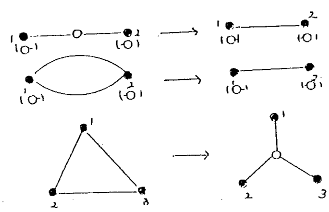


Fig. 12. Fundamental reduction rules

Cycle+Line Reduction Ordering and Strongly Normalizing

Definition 23 Cycle+Line : Let s be a boundary graph. The Cycle+Line in s denoted by $s^{\sharp}$ is the sum of the number of elements in a set defined as follows:

(1) a set of cycles consists of all the closed paths in s.

(2) a set of lines consists of all the lines in s.

Definition 24 Cycle+Line Reduction Ordering : Let $\mathcal{R}$ be a *GRS*, where R has a fundamental rule set. R has cycle+line reduction ordering if $s^{\sharp} > t^{\sharp}$ for any one reduction step : $s \longrightarrow t$

In graph reduction, reduction orientation is often essential because serious trouble often occurs in the nontermination reduction sequences. For example, if $\Delta - Y$ and $Y - \Delta$ are used simultaneously as same reduction rules, infinite cycle results. Here, we define reduction ordering in the rules as the sum of the numbers of cycles and lines that is decreased in each reduction step. By using this ordering it can be trivially concluded that a reduction sequence will terminate in a finite number of steps, since this preserves strongly normalizing.
Note that the $Y - \Delta$ rule is not decreased in cycle+line reduction ordering, because the sum of the cycles and lines conversely increases in each reduction step.

Proposition 4 *$\mathcal{R}$ with cycle+line reduction ordering is strongly normalizing.*

Proof. The proof is trivial since the cycle+line is finite and monotonically decreases.

Completeness of a Set of Four Reduction Rules

When a R is complete, we call R a complete rule set because R is determined by the rule set of the system.

In the systems, any rule in a complete rule set is generated by the fundamental reduction rules. However, it has been proved [5]that any graph that can be reduced to one link by the fundamental rule set with adequate control strategy can be reduced to one link by a complete rule set. The complete rule allows us to reduct graphs without using complex reduction strategy shown below in Fig. 13.

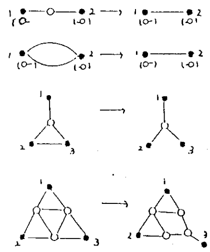


Fig. 13. Example of complete rule sets

(1) A rule set consisting of r_1 and r_2 is called $S - P$.

(2) A rule set consisting of r_1 , r_2 , and r_3 is called $\Delta^1 - Y$.

(3) A rule set consisting of r_1, r_2, r_3, and r_4 is called octahedral or $\Delta^4 - Y$.

The octahedral rule set is not equivalent to the original $\Delta - Y$ set in regard to reduction power, but it does have the following advantages.

Theorem 2 *The above rule sets are complete.*

Proof. It is obvious that each rule set is strongly normalizing in cycle+line reduction. Thus we only have to show that every critical pairs of R is *convergent*, according to the following.

(1) In the $S - P$ rule set case, the proof is trivially obtained.

(2) In the $\Delta^1 - Y$ rule set case, the critical pairs obtained by the superposition of r_1 on r_3, r_2 on r_3 and r_3 on r_3 have to checked, since the critical pairs obtained by the superposition of r_1 on r_2, r_1 on r_1,r_2 on r_2 are convergent for (1). To prove convergence of the critical pairs obtained by the superposition of r_3 on r_3 is slightly complex, but will be shown below in Fig. 14. The proofs for other superposition case are obvious by the same calculation.

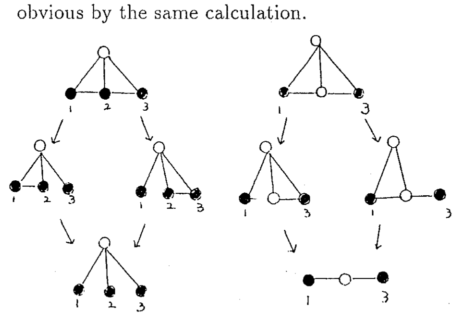


Fig. 14. Convergence of critical pairs obtained by the superposition of r_3 on r_3.

(3) In the octahedral rule set case convergence of the critical pairs obtained by the superposition of r_4 on r_4, and r_4 and others still has to be proved, since we got the result in (2). In these superpositions we only needed to prove the critical pairs obtained by the superposition of r_4 on r_3 and r_4 on r_4.

The convergence of the critical pairs obtained so far by the superposition of r_4 on r_3 and r_4 on r_4 is shown below in Fig. 15. Proofs can be easily obtained for other superposition by the same calculation method.

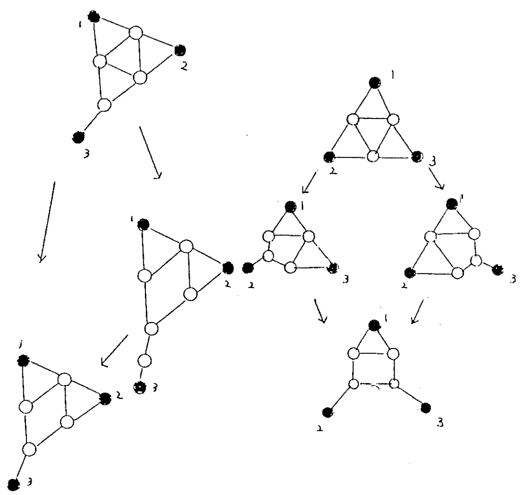


Fig. 15. Convergence of critical pairs obtained by the superposition of r_4 on r_3 and r_4 on r_4.

An example reducible to one link by the above complete reduction rules is shown below in Fig. 16. Note that the original graph is reduced to one link in each reduction sequence and the same final result is obtained for each reduction path.

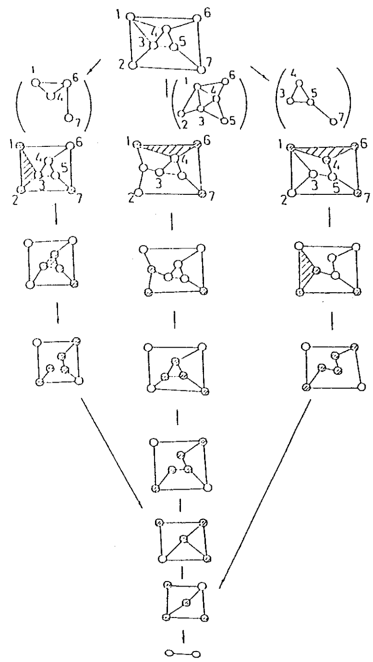


Fig. 16. Reduction by complete reduction rules

4.1.2 Another Interesting Example

In the previous section we showed that the $\Delta - Y$ rule set is complete, but more rules can be added to r_1, r_2, r_3, r_4. Rule r_4 is denoted by Δ^4 and r_5 by Δ^9. This procedure is defined recursively as in Fig. 17 below. $r_{(n-2)}$ is denoted by Δ^{n^2}. Here n^2 stands for the number of triangles in Δ^{n^2}.

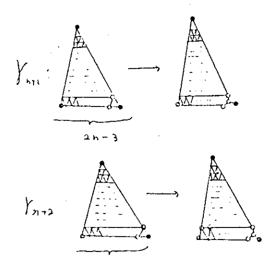


Fig. 17. Recursively defined class.

Here $\cup\Delta^{n^2} = SP \cup \Delta^1 \cup \ldots\Delta^{n^2}$, thus the next theorem is easily obtained.

Theorem 3 *For any n, $\cup\Delta^{n^2}$ is complete.*

Proof. For $n \leq 2$, the proof is obtained from Theorem 2. If $n \geq 3$, let i, j be satisfied with $3 \leq i, j \leq n$, then the critical pair for the superposition of Δ^{i_2} on Δ^{j_2} is obtained by the same as for Δ^4 in Fig. 15. Also, the other critical pairs are convergent according to Theorem 2. Thus, the proof is obtained.

Note the this theorem means we can infinitely produce an increasing chain of complete rule sets.

5 Conclusion

This paper presented a new theory Graph Rewriting Systems and its application to network reliability analysis. In GRS a necessary and sufficient condition is given for the Church Rosser property under termination. Furthermore, the octahedral rule set is complete in GRS, thus allowing graphs to be reduced without taking rule application order or application place into consideration.

6 Acknowledgments

The authors would like to express their gratitude to Takeo Abe, Ken Mano, Seiichi Komura and the members of the Koyama research group at the NTT Basic Research Laboratories for their support and encouragement.

References

[1] Barendreght, H.P.: *"The Lambda Calculus-Its Syntax and Semantics"*, Revised Edition, North Holland, 1984.

[2] Ball, M. O.: *"Complexity of Network Reliability Computations"*, Networks , Vol.10, p.p. 153-165, 1980.

[3] Bauderon, Y. and Courcell, B. *'Graph Expression and Graph Rewritings* Math. Systems Theory, 20, p.p. 83-127, 1987.

[4] Harary, F. : *"Graph Theory"*, Addison-Wesley, Reading, MA 1969.

[5] Hayashi, M. and Okada, Y.: *"Application of Graph Rewriting Systems to Network Reliability Analysis."* to appear Autumn National Convention Record IEICE, 1990.

[6] Hindley & Seldin: *"Introduction to Combinators and Lambda Calculus"*, London Mathematical Society Student Texts, Cambridge University, 1986.

[7] Huet, G.P.: *"Confluent Reductions: Abstract Properties and Applications to Term Rewriting Systems "*, Journal of the Association for Computing Machinery 27(4), p.p. 11-21, 1980.

[8] Knuth, D.E. and Bendix, P.B. : *"Simple Word Problems in Universal Algebras"*, Computational problems in abstract algebras, Pergamon Press, Oxford 1970.

[9] Maggiolo-Schettini, A. and Winkowski, J. *"Process of Transforming Structures"* Jounal of Computer and System Sciences 24, p.p. 245-282, 1982.

[10] Newman, M.H.A. : *"On theories with a combinatorial definition of 'equivalence' "*, Ann. Math., 43, p.p. 223-243, 1942.

[11] Politof, T and Satyanarayana, A : *"Efficient Algorithms for Reliability Analysis of Planar Networks - A survey"*, IEEE Transactions Reliability, Vol R-35, No.3 , August, 1986.

[12] Rosenthal, A. : *"Computing The Realiability of Complex Networks* SIAMJ. APPL. MATH. Vol.32, No.2, March, 1977.

[13] Rosenthal, A and Frisque, D. : *"Transformations for Simplifying Network Reliability Calculations"* Networks, Vol.7, No. 2, pp. 97-111, 1977.

[14] Satyanarayana, A and Wood, R. K. : *"A linear-Time Algorithm from computing K-terminal reliability of normal graphs in series-parallel networks,"* SIAM J. Computing, Vol 14, Nov 1985.

[15] Sethi, R. : *"Testing for the Church-Rosser theorems, "* J.ACM, 21,4 October p.p. 671-679, 1974.

[16] Shenadec, F.L. : *"Canonical forms in Finitely Presented Algebra"*, Research Notes in Theoretical Computer Science, Pitman London, 1986.