ISSN 1344-8803, CSSE-11 September 26, 2000

Formalizing the Definition and Evolution of Models in a Repository using the Relational Graph Expressions [1]

Yoshihiro Mizoguchi[†] and Pericles Loucopoulos[††]

[†]Department of Control Engineering and Science,
Kyushu Institute of Technology
680-4 Kawazu, Iizuka, Fukuoka 820-8502, Japan
E-mail: ym@ces.kyutech.ac.jp

[††]Department of Computation,
University of Manchester Institute of Science and Technology
P. O. Box 88, Manchester M60 1QD, UK
E-mail: pl@co.umist.ac.uk

Abstract: The use of repository technology for the management of models of an information system or an enterprise knowledge is regarded as increasingly important by researchers and practitioners alike. The primary usage of repositories is in the creation and management of evolution of models, conceptual models, the majority of which have graphical external representation properties. In this paper we introduce a formalisation for the creation and evolution of models stored in a repository. We define both models and repositories in terms of grpah structures using relation expressions. References to stored objects in a repository are expressed in terms of partial functions between graphs. Modifications of models, in terms of their editing during their lifetime, are also represented by partial functions. An important consideration is the maintainance of integrity of stored objects by multiple models when there exists an intermodel relationship. We define these dependencies formally and prove properties on them.

Key Words: Graph rewritings; Relational calculus; Formal methods; Information models;

[1]Technical Report in Computer Science and Systems Engineering, Log Number CSSE-11, ISSN 1344-8803.

Formalizing the Definition and Evolution of Models in a Repository using the Relational Graph Expressions

Yoshihiro Mizoguchi[1] and Pericles Loucopoulos[2]

[1] Kyushu Institute of Technology,
680-4 Kawazu, Iizuka, 820-8502, JAPAN
ym@ces.kyutech.ac.jp
[2] University of Manchester Institute of Science and Technology,
P.O. Box 88, Manchester M60 1QD, UK
pl@co.umist.ac.uk

1 Introduction

The use of repository technology for software developments is regarded as increasingly important by researchers and practitioners alike.

Developing an information system requires a suitable environment designed to support the process of capture, representation, amplification and dissemination of knowledge from a variety of sources, both internal and external. Typically these tools will provide a repository management system for maintaining knowledge specifications using some DBMS; editing facilities to insert new information in the knowledge base; browsing functions, etc.

A repository therefore manages these models for their entire life, from birth to their deletion. During its lifetime a model may undergo many changes. Furthermore, these models represent in essence different views onto a common knowledge. In dealing with the creation and evolution of its models, a repository needs to cater for:

– multiplicity of views on a common concept
– modification of graphical representations and
– modification of stored components.

The majority of models are of a graphical nature. In this paper we focus exclusively on this type of model. We advocate a formal approach to defining the graphical models as well as their changes, both at an external visual level and in their representation in a repository. This formalisation of the definition and evolution of models provides a strong theoretical basis for ensuring consistency of a repository throughout the life time of all its stored models. The formalisation is based on the use of relational calculus whereby both a model and a repository are defined as a graph.

2 Fundamentals on Relational Calculus

In this section, we summerize basic notation and properties of relational calculus. A *relation* α of a set A into another set B is a subset of the cartesian product $A \times B$ and denoted by $\alpha : A \rightharpoondown B$. The *inverse relation* $\alpha^\sharp : B \rightharpoondown A$ of α is a relation such that $(b,a) \in \alpha^\sharp$ if and only if $(a,b) \in \alpha$. The *composite* $\alpha\beta : A \rightharpoondown C$ of $\alpha : A \rightharpoondown B$ followed by $\beta : B \rightharpoondown C$ is a relation such that $(a,c) \in \alpha\beta$ if and only if there exists $b \in B$ with $(a,b) \in \alpha$ and $(b,c) \in \beta$.

As a relation of a set A into a set B is a subset of $A \times B$, the inclusion relation, union, intersection and difference of them are available as usual and denoted by $\subseteq$, $\cup$, $\cap$ and $-$, respectively. The *identity relation* $\mathrm{id}_A : A \rightharpoondown A$ is a relation with $\mathrm{id}_A = \{(a,a) \in A \times A \mid a \in A\}$ (the diagonal set of A).

The followings are the basic properties of relations and indicate that the totality of sets and relations forms a category **Rel** with involution (or shortly I-category).

A *partial function* f of a set A into a set B is a relation $f : A \rightharpoondown B$ with $f^\sharp f \subseteq \mathrm{id}_B$ and it is denoted by $f : A \rightarrow B$. A *(total) function* f of a set A into a set B is a relation $f : A \rightharpoondown B$ with $f^\sharp f \subseteq \mathrm{id}_B$ and $\mathrm{id}_A \subseteq ff^\sharp$, and it is also denoted by $f : A \rightarrow B$. Clearly a function is a partial function. Note that the

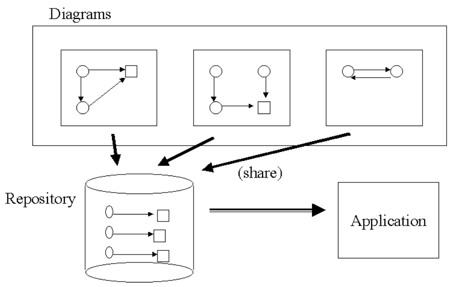

Fig. 1. Common data are stored in a repository

identity relation id_A of a set A is a function. The definitions of partial functions and (total) functions here coincide with ordinary ones. A partial function $f : A \to B$ is injective if and only if $ff^\sharp \subseteq \mathrm{id}_A$ and surjective if and only if $f^\sharp f = \mathrm{id}_B$. For a subset $X \subseteq A$, we denote the inclusion function by $i_X : X \to A$.

Given a relation $\alpha : A \rightharpoondown B$, the *domain* is defined by the set $\mathrm{dom}(\alpha) = \{a \in A | (a, a) \in \alpha\alpha^\sharp\}$, and *domain relation* $d(\alpha) : A \rightharpoondown A$ of α is a relation defined by $d(\alpha) = \alpha\alpha^\sharp \cap \mathrm{id}_A$. The domain relation $d(\alpha^\sharp) : B \rightharpoondown B$ of $\alpha^\sharp$ corresponds with the image of α. A partial function $f : A \to B$ is a function if and only if $d(f) = \mathrm{id}_A$.

We denote the category of sets and functions by **Set** and the category of sets and partial functions by **Pfn**. Both of **Set** and **Pfn** have all small limits and colimits, so in particular, they have pushouts. Note that **Pfn** is equivalent to the category of sets with a base point (a selected element) and base point preserving functions. We assume that the readers are familiar with pushout constructions in **Pfn**. A singleton set $\{*\}$ is denoted by 1 and the maximum relation from a set A into 1 by $\Omega_A : A \rightharpoondown 1$, that is, $\Omega_A = \{(a, *) | a \in A\}$. We define a relation $\Theta_A = \Omega_A \Omega_A^\sharp$.

3 Models in a Repository

In this section, we formalize models, repositories, and those modifications and changes using graphs and relational expressions. We consider a model and a repository as graphs (cf. Figure 1).

Common data in a model is stored in a repositry, this stuation is expressed by a partial function between graphs. At the case of editing some model, the modification of the model is expressed also by a partial function between graphs. A modification of a model which common data is stored in a repository induce some change of the repository. The change of the repository is defined by a graph using relational expressions (cf. Figure 2). A change of a repository induce modifications of other models which common data are stored in the repository. The modification of the other model also defined by a graph using relational expressions. We show some properties about those modifications and changes. Especially, we prove that we obtain the same graph by doing step-by-step the sequence of modifications and doing all-step-together of them. We also have same results about step-by-step changing and all-step-together changing.

We fix a set Σ which represent labels of nodes which are stored in repositories. Let M is an integer greater than 1 and $\Sigma = \{\cdot, \wedge, \vee, G_1, G_2, \cdots, G_M\}$. It is not essential to fix elements in a Σ in our arguments. In our expression of a model using a graph, we use a edge connecting to a node in Σ for a label of a node (cf. Figure 3).

A model is considered as a simple graph. A repository which contains common data of models is also expressed by a simple graph. Since a repository stores only informations of labels of nodes, a graph which expresses a repository does not have any edge between nodes not in Σ. This condition is denoted by a relational expression $\Theta_{R-\Sigma} \cap \xi = \phi$. Common data of a model stored in a repository are expressed by a partial injective morphism. We show those things formally as follows.

Definition 1. A *model* $\langle A, \alpha \rangle$ is a pair of a set A and a relation $\alpha : A \rightharpoondown A$. A *repository* $\langle R, \xi \rangle$ is a pair of a set R and a relation $\xi : R \rightharpoondown R$ where $\Theta_{R-\Sigma} \cap \xi = \phi$. A *model in a repository* $\langle \langle A, \alpha \rangle, i_A, \langle R, \xi \rangle \rangle$ is a triple of a diagram $\langle A, \alpha \rangle$, an injective partial morphism $i_A : A \to R$, and a repository $\langle R, \xi \rangle$.

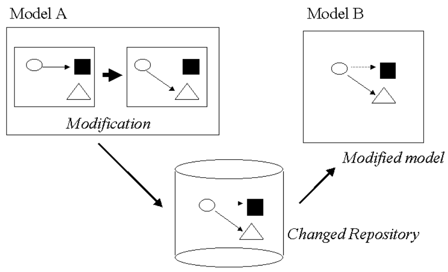

Fig. 2. Modifications of models and a change of a repository

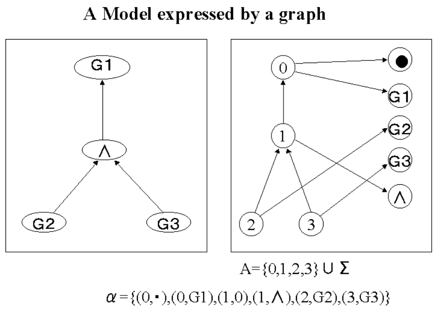

Fig. 3. A model expressed by a graph

Example 1. Figure 4 shows a model in a repository $\langle\langle A,\alpha\rangle, i_A, \langle R,\xi\rangle\rangle$. A model $\langle A,\alpha\rangle$ consists of $A = \{0,1,2,3\} \cup \Sigma$ and $\alpha = \{(0,\cdot),(0,G_1),(1,0),(1,\wedge),(2,G_2),(3,G_3)\}$. A repository $\langle R,\xi\rangle$ consists of $R = \{0,2,3\} \cup \Sigma$ and $\xi = \{(0,G_1),(2,G_2),(3,G_3)\}$. And $i_A = \{(0,0),(2,2),(3,3)\}$.

To modify a model using a editing tool are expressed by a graph transformation from a graph to an edited graph. The transformation is denoted by a partial function f between two graphs. A transformation preserve labels of nodes, so we restrict f by a relational expression $\mathrm{id}_\Sigma \cap f = \Theta_\Sigma \cap f$. We assume f a partial injective morphism to be simplify following arguments. Further we do not allow a transformation which merge two nodes to one node. Many practical editings are deletion and insertion of nodes, so it is enough to discuss with those restrictions to f.

Definition 2. A *modification* of a model $\langle\langle A,\alpha\rangle, f, \langle B,\beta\rangle\rangle$ is a triple of models $\langle A,\alpha\rangle$ and $\langle B,\beta\rangle$ and an injective partial morphism $f : A \to B$ where $\mathrm{id}_\Sigma \cap f = \Theta_\Sigma \cap f$.

Example 2. The Figure 5 is an example of a modification of a model, where $f = \{(0,4),(2,6)\} \cup \mathrm{id}_\Sigma$. In other words, the modification remove a goal G_3, insert a goal G_4 and change $\wedge$ to $\vee$.

At the case of modifying a model in a repository, common data stored in a repository are influenced by the modification. The definition of the influenced modifications of a repository, which we call *a change of a repository*, is not straightforward. There are many possibility of defining. In this paper, we define it by a set of nodes constructed using a property of pushout and a set of edges defined by a relational mathematical expression (Mizoguchi and Kawahara 1995 [3]).

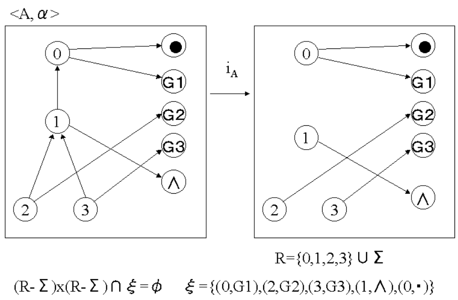

Fig. 4. A model in a repository

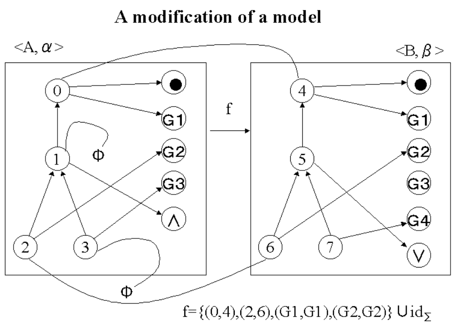

Fig. 5. A modification of a model

Definition 3. Let $\langle\langle A,\alpha\rangle, i_A, \langle R,\xi\rangle\rangle$ be a model in a repository, and $\langle\langle A,\alpha\rangle, f, \langle B,\beta\rangle\rangle$ a modification. Construct a pushout

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "i_A"'] & B \arrow[d, "i_B"] \\ R \arrow[r, "f_+"'] & S \end{tikzcd}$$

in **Pfn** and define $\sigma = i_B^\sharp \beta i_B \cup f_+^\sharp(\xi - i_A^\sharp \Theta_A i_A) f_+$. Then we have a repository $\langle S,\sigma\rangle$ and a model in a repository $\langle\langle B,\beta\rangle, i_B, \langle S,\sigma\rangle\rangle$. We call $\langle S,\sigma\rangle$ a *changed repository* induced by the modification $\langle\langle A,\alpha\rangle, f, \langle B,\beta\rangle\rangle$ and the modification $\langle\langle A,\alpha\rangle, f, \langle B,\beta\rangle\rangle$. And simply we write the situation with a diagram

$$\begin{tikzcd} \langle A,\alpha\rangle \arrow[r, "f"] \arrow[d, "i_A"'] & \langle B,\beta\rangle \arrow[d, "i_B"] \\ \langle R,\xi\rangle \arrow[r, "f_+"'] & \langle S,\sigma\rangle \end{tikzcd}$$

.

The set S defined in above definition is roughly considered as a set $B \cup R$ in which common data in A are identified. The relational expressin for σ means that a set of edge contains all edges in β and ξ removing edges in ξ related to nodes in A.

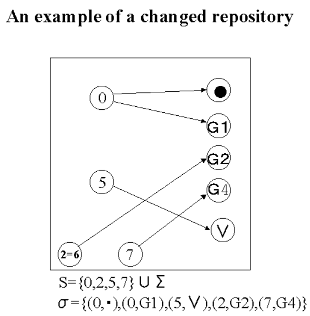

Fig. 6. An example of a changed repository

Example 3. Figure 6 shows a changed repository $\langle S,\sigma\rangle$ induced from the model in Figure 4 by the modification in Figure 5. where $S = \{0,6,7\} \cup \Sigma$ and $\sigma = \{(0,G_1),(6,G_2),(7,G_4)\}$.

We defined a change of a repository induced by a modification of a model by our own way. To confirm our definition's correctness, we prove the following property. Let $\langle\langle A,\alpha\rangle, i_A, \langle R,\xi\rangle\rangle$ be a model in a repository. When we modify a model twice from A to B to C, we can change the repository step by step from R to S to T. We call this changing *step-by-step change*. When we consider those two modification as a single modification from A to C, we have another changed repository T' from R by our definition. We call the last changing *all-step-together change*. A correct and useful definition of a changed repository should garantee that those two change step-by-step change and all-step-toghether change are same. In next proposition, we prove this property using simple but formal relational caluculations.

Proposition 1. *Let*

$$\text{\underline{step-by-step:}}\quad \begin{tikzcd} \langle A,\alpha\rangle \arrow[r, "f"] \arrow[d, "i_A"'] & \langle B,\beta\rangle \arrow[r, "g"] \arrow[d, "i_B"] & \langle C,\gamma\rangle \arrow[d, "i_{C'}"] \\ \langle R,\xi\rangle \arrow[r, "f_+"'] & \langle S,\sigma\rangle \arrow[r, "g_+"'] & \langle T,\tau\rangle \end{tikzcd}$$

and

$$\text{\underline{all-step-together:}}\quad \begin{tikzcd} \langle A,\alpha\rangle \arrow[r, "fg"] \arrow[d, "i_A"'] & \langle C,\beta\rangle \arrow[d, "i_C"] & \text{(Model modification)} \\ \langle R,\xi\rangle \arrow[r, "(fg)_+"'] & \langle T',\tau'\rangle & \text{(Changed repository)} \end{tikzcd}$$

are changes of repositories. Then $\langle T, \tau\rangle$ and $\langle T', \tau\rangle$ are same.

Proof. Since T and T' are defined by pushouts, we can prove $T = T'$ using simple pushout properties. Next we show $\tau = \tau'$ using relational calculuses.

$$\begin{aligned}
\tau &= i_C^\sharp \gamma i_C \cup g_+^\sharp(\sigma - i_B^\sharp \Theta_B i_B) g_+ \\
&= i_C^\sharp \gamma i_C \cup g_+^\sharp((i_B^\sharp \beta i_B \cup f_+^\sharp(\xi - i_A^\sharp \Theta_A i_A) f_+) - i_B^\sharp \Theta_B i_B) g_+ \\
&= i_C^\sharp \gamma i_C \cup g_+^\sharp((f_+^\sharp \xi f_+ - f_+^\sharp i_A^\sharp \Theta_A i_A f_+) - i_B^\sharp \Theta_B i_B) g_+ \\
&= i_C^\sharp \gamma i_C \cup g_+^\sharp((f_+^\sharp \xi f_+ - i_B^\sharp \Theta_B i_B) - f_+^\sharp i_A^\sharp \Theta_A i_A f_+) g_+ \\
&= i_C^\sharp \gamma i_C \cup g_+^\sharp((f_+^\sharp \xi f_+ - (f_+^\sharp \xi f_+ \cap i_B^\sharp \Theta_B i_B)) - f_+^\sharp i_A^\sharp \Theta_A i_A f_+) g_+
\end{aligned}$$

Since $f_+^\sharp \xi f_+ \cap i_B^\sharp \Theta_B i_B \subseteq f_+^\sharp \Theta_R f_+ \cap i_B \Theta_B i_B \subseteq f_+^\sharp i_A^\sharp \Theta_A i_A f_+$, We have

$$\begin{aligned}
\tau &= i_C^\sharp \gamma i_C \cup g_+^\sharp(f_+^\sharp \xi f_+ - f_+^\sharp i_A \Theta_A i_A f_+) g_+ \\
&= i_C^\sharp \gamma i_C \cup g_+^\sharp f_+^\sharp(\xi - i_A^\sharp \Theta_A) g_+ \\
&= i_C^\sharp \gamma i_C (f_+ g_+)^\sharp (\xi - i_A^\sharp \Theta_A i_A)(f_+ g_+) \\
&= \tau'.
\end{aligned}$$

$\square$

Next, we consider modifications of other models effected by a change of a repository which the models have common data in. There also exist many possibilities of definitions of an effected modified model. We introduce a definition of a modified model using relational expressions. Let $\langle\langle X, \chi\rangle, i_X, \langle R, \xi\rangle\rangle$ be a model in a repository, $\langle S, \sigma\rangle$ a changed repository, $f_+ : \langle X, \chi\rangle \to \langle S, \sigma\rangle$ a relation between repositories. A set of edges of modified model are expressed by the expression $\chi' = (\chi - i_X i_X^\sharp \chi i_X i_X^\sharp) \cup i_X f_+ \sigma f_+^\sharp i_X^\sharp$. This means to remove edges related to edges in R from ξ and to insert all edges in σ.

Definition 4. Let

$$\begin{tikzcd}
\langle A, \alpha\rangle \arrow[r, "f"] \arrow[d, "i_A"'] & \langle B, \beta\rangle \arrow[d, "i_B"] \\
\langle R, \xi\rangle \arrow[r, "f_+"'] & \langle S, \sigma\rangle
\end{tikzcd}$$

be a changed repository and $\langle\langle X, \chi\rangle, i_X, \langle R, \xi\rangle\rangle$ a model in a repository. A *modified model* induced by a changed repository is a model in a repository $\langle\langle X, \chi'\rangle, i_X f_+, \langle S, \sigma\rangle\rangle$ where $\chi' = (\chi - i_X i_X^\sharp \chi i_X i_X^\sharp) \cup i_X f_+ \sigma f_+^\sharp i_X^\sharp$.

We prove a similar property of a modified model as Proposition 1 of a changed repository. We show that a *step-by-step* modification induce a same result as an *all-step-together* modification.

Proposition 2. *Let*

$$\begin{tikzcd}
& \langle A, \alpha\rangle \arrow[r, "f"] \arrow[d, "i_A"] & \langle B, \beta\rangle \arrow[r, "g"] \arrow[d, "i_B"] & \langle C, \gamma\rangle \arrow[d, "i_C"] \\
& \langle R, \xi\rangle \arrow[r, "f_+"'] & \langle S, \sigma\rangle \arrow[r, "g_+"'] & \langle T, \tau\rangle \\
\underline{\textit{step-by-step:}} & \langle X, \chi\rangle \arrow[u, "i_X"] & \langle X, \chi'\rangle \arrow[u, "i_X f_+"'] & \langle X, \chi''\rangle \arrow[u, "i_X f_+ g_+"']
\end{tikzcd}$$

and

$$\begin{tikzcd}
& \langle A, \alpha\rangle \arrow[r, "fg"] \arrow[d, "i_A"] & \langle C, \beta\rangle \arrow[d, "i_B"] \\
& \langle R, \xi\rangle \arrow[r, "(fg)_+"'] & \langle T, \tau\rangle \\
\underline{\textit{all-step-together:}} & \langle X, \chi\rangle \arrow[u, "i_X"] & \langle X, \chi^*\rangle \arrow[u, "i_X (fg)_+"']
\end{tikzcd}$$

are modified models. Then $\langle X, \chi''\rangle$ and $\langle X, \chi^*\rangle$ are same, that $\chi'' = \chi^*$.

Proof.

$$\begin{aligned}\chi'' &= (\chi' - i_{X'} i^\sharp_{X'} \chi' i_{X'} i^\sharp_{X'}) \cup i_{X'} g_+ \tau g^\sharp_+ i^\sharp_{X'} \\ &= (((\chi - i_X i^\sharp_X \chi i_X i^\sharp_X) \cup i_X f_+ \sigma f^\sharp_+ i^\sharp_X) - i_{X'} i^\sharp_{X'} \chi' i_{X'} i^\sharp_{X'}) \cup i_X f_+ g_+ \tau g^\sharp_+ f^\sharp_+ i^\sharp_X)\end{aligned}$$

Since $i_X f_+ \sigma \chi' f^\sharp_+ i^\sharp_X \subseteq i_{X'} i^\sharp_{X'} \chi' i_{X'} i^\sharp_{X'}$ and $i_{X'} i^\sharp_{X'} \chi' i_{X'} i^\sharp_{X'} \cap \chi \subseteq i_X i^\sharp_X \chi i_X i^\sharp_X$, we have

$$\begin{aligned}\chi'' &= ((\chi - i_X i^\sharp_X \chi i_X i^\sharp_X) - i_{X'} i^\sharp_{X'} \chi' i_{X'} i^\sharp_{X'}) \cup i_X f_+ g_+ \tau g^\sharp_+ f^\sharp_+ i^\sharp_X \\ &= (\chi - i_X i^\sharp_X \chi i_X i^\sharp_X) \cup i_X f_+ g_+ \tau g^\sharp_+ f^\sharp_+ i^\sharp_X \\ &= \chi'\end{aligned}$$

□

4 Conclusion

In this paper, we introduce a formalization of definitions and evolutions of models in a repository using graph structures and relational expressions. We analysed some properties by formal ways. We proved a property of the step-by-step modification and the all-step-together modification in our formalizing. We are going to investigate further related properties with our introduced mechanisms and find out some useful insights to develop model modification tools.

References

1. Klimathianakis, P. and Loucopoulos, P. DELOS - A Repository Based Environment for Developing Network Centric Application, *9th International Conference on Advanced Information Systems Engineering (CAiSE'97)*, A. Olive (ed.), Springer-Verlag, Barcelona, June 16-20, 1997.
2. Loucopoulos, P. and Kavakli, V. Enterprise Knowledge Management and Conceptual Modelling, *International Symposium on 'Past, Present and Future of Conceptual Modeling'*, P. P. Chen (ed.), Springer Verlag, Los Angeles, USA, 1997.
3. Mizoguchi, Y. and Kawahara, Y. Relational Graph Rewritings. Theoretical Computer Science, Vol.141, pp.311-328, 1995.