Defining, mining and reasoning on rules in tabular data

Marie Agier[1,2] and Jean-Marc Petit[2]

[1] DIAGNOGENE
15000 Aurillac, FRANCE
[2] LIMOS, UMR 6158 CNRS, Univ. Clermont-Ferrand II
63177 Aubière, FRANCE

Abstract. Different rule semantics have been defined successively in many contexts such as functional dependencies in databases or association rules in data mining to mention a few. In this paper, we focus on the class of rule semantics for tabular data for which Armstrong's axiom system is sound and complete, so-called *well-formed semantics*. The main contribution of this paper is to show that an *equivalence* does exist between some syntactic restrictions on the natural definition of a given semantics and the fact that this semantics is well-formed. From a practical point of view, this equivalence allows to prove easily whether or not a new semantics is well-formed. Moreover, the same reasoning on rules can be performed over any well-formed semantics.
We also point out the relationship between our generic definition of rule satisfaction and the underlying data mining problem, i.e. given a well-formed semantics and a relation, discover a cover of rules satisfied in this relation.
This work takes its roots from a bioinformatics application, the discovery of gene regulatory networks from gene expression data.

Keywords Rules, implications, Armstrong's axiom system, data mining.

1 Introduction

The notion of *rules* or *implications* is very popular and appears in different flavors in databases, data mining or artificial intelligence communities. The two more famous examples of rules are association rules [1] and functional dependencies [2]. As such, a simple remark can be done on such rules: their syntax is the same but their semantics widely differs. In this paper, we consider rules to be defined on *tabular datasets*. Basically, tabular dataset is equivalent to a *relation* over a set U of distinguished attributes (or columns) in databases terminology. In this context, a *rule* is an expression of the shape $X \rightarrow Y$ i.e. "X implies Y" with $X, Y \subseteq U$.

The *semantics* of a rule $X \rightarrow Y$ over U is the *meaning*, the *sense* one wants to give to this rule: Given a relation r, a rule $X \rightarrow Y$ is said to be *satisfied* in r if the semantics of the rule is true (or valid) in r. We identified three main

components to specify a semantics for rules in a relation: The type of the data, the subsets of the relation on which the rule applies and the predicates occuring in the "if... then..." part of the rule. We proposed in this paper a natural and "generic" definition of a semantics in order to be able to capture most of existing semantics already known on tabular data.

Note that error measures and quality measures are not taken into account in our generic definition of a semantics. These measures can be generally applied to many different semantics and do not belong to what we believe to be the core definition of a rule semantics. Moreover, we do not want to define as many semantics as there are measures. In a data mining context, error and quality measures can be integrated a posteriori to sort and to qualify the rules.

Furthermore, we focus on those semantics verifying Armstrong's axioms, so-called "well-formed semantics" [3], i.e. semantics for rules on which Armstrong's axiom system applies. The practical interests are twofold:

- Firstly, *reasoning* can be performed on rules from the Armstrong's axioms. For instance, given a set of rules F, it is possible to know if a rule is *implied* by this set of rules in linear time [4].
- It is also possible to work on "small" *covers* of rules [5–7] and to use a discovery process specific to the considered cover, but applicable to *all* well-formed semantics.

Paper contribution The contribution of this paper is to show that an *equivalence* does exist between some syntactic restrictions on the natural definition of a given semantics and the fact that this semantics is *well-formed*.

From a practical point of view, this equivalence allows to prove easily that a new semantics is well-formed: So far, for a given semantics, we had to give a proof of the soundness and the completeness of the Armstrong's axiom system for this semantics, this proof being not always trivial. Now, it is just enough to show that this semantics complies with the proposed syntactic restrictions.

We also point out the relationship between our generic definition of rule satisfaction and the underlying data mining problem, i.e. given a well-formed semantics and a relation, discover a cover of rules satisfied in this relation. More precisely, we show how a base of the closure system for any well-formed semantics can be computed from the dataset.

Application This work takes its roots from a bioinformatics application, the discovery of gene regulatory networks from gene expression data. The challenge is to find out relationships between genes that reflect observations of how expression level of each gene affects those of others. The conjecture that association rules could be a model for the discovery of gene regulatory networks has been partially validated in [8–11]. Nevertheless, we believe that many different kinds of rules between genes could be useful with respect to some biological objectives and the restricted setting of association rules could be not enough to cope with

this diversity. Three different types of rules were already proposed in [12] for gene expression data.

In this context, the main application of this paper is to offer a framework in which biologists may define their "own customized semantics" for rules with regard to their requirements. Once a semantics is proved well-formed, i.e. this semantics just complies with the proposed syntactic restrictions, well-known inference methods on rules can be performed for biologists.

This work has been intended for gene expression data, however, in this paper we extend our proposition to all types of tabular data.

Paper organization We give in Section 2 some examples of rule semantics. In Section 3, we propose a natural definition of a semantics using some syntactic restrictions. In Section 4, we further restrict the syntax and give the main result of this paper. We point out in Section 5 some relationships between our proposition and the underlying data mining problem. In Section 6, we give the related contributions of this work and finally, we conclude and give some perspectives in Section 7.

2 Motivating examples

To show the interest of our proposition, we give in the sequel four examples of semantics for tabular data, some of them in the context of gene expression data [12]. These examples show that many rule semantics can be defined.

Let r be a relation over U and $X, Y \in U$ two subsets of attributes. In the context of gene expression data, an attribute is a gene and the domain of attributes is the set of real numbers.

Example 1. Let s_1 be a semantics studying for example the levels of expression of genes, s_1 can be defined as follows, using two user-supplied thresholds ϵ_1 and ϵ_2:

$r \models_{s_1} X \rightarrow Y$ if and only if $\forall t \in r$, if $\forall A \in X, \epsilon_1 \leq t[A] \leq \epsilon_2$ then $\forall A \in Y, \epsilon_1 \leq t[A] \leq \epsilon_2$.

This semantics points out a semantics close to association rules without discretisation phase.

Example 2. Let s_2 be a semantics studying for example the evolution of gene expression levels, s_2 can be defined as follows, using two user-supplied thresholds ϵ_1 and ϵ_2:

$r \models_{s_2} X \rightarrow Y$ if and only if $\forall t_i, t_{i+1} \in r$, if $\forall A \in X, \epsilon_1 \leq t_{i+1}[A] - t_i[A] \leq \epsilon_2$ then $\forall A \in Y, \epsilon_1 \leq t_{i+1}[A] - t_i[A] \leq \epsilon_2$.

Note that an *order* has to exist among tuples, such a constraint being implicitly expressed with indices on tuples.

Example 3. Let s_d be a new semantics studying the Euclidian distance between for example gene expression profiles, s_d can be defined as follows, using two user-supplied thresholds ϵ_1 and ϵ_2:

$r \models_{s_d} X \rightarrow Y$ if and only if $\forall t_i, t_j \in r$, if $\epsilon_1 \leq d(t_i[X], t_j[X]) \leq \epsilon_2$ then $\epsilon_1 \leq d(t_i[Y], t_j[Y]) \leq \epsilon_2$.

Example 4. Let s_{mvd} be the semantics of multivalued dependencies, s_{mvd} can be defined as follows:

$r \models_{s_{mvd}} X \rightarrow Y$ if and only if $\forall t_i, t_j \in r$, if $\forall A \in X, t_i[A] = t_j[A]$ then $(\forall A \in Y, t_i[A] = t_j[A]$ *or* $\forall A \in U \setminus Y, t_i[A] = t_j[A])$.

These various examples show that with the same syntax, a rule may have very different meanings and from the same dataset, several semantics can be defined and interesting for the experts.

3 Well-formed semantics

We identified three components in the definition of a semantics for rules in a relation:

- The type of the data: Rule semantics can be generally applied for some restricted types of data, for example binary or categorical attributes, temporal data, presence of classes... The nature of the data being analyzed clearly influences the definition of a semantics.
- The subsets of the relation on which a rule applies: An important characteristic of a semantics is the condition on tuples to take into account. We can for example study tuples one by one (like association rules), we can do a pairwise comparison of tuples (like functional dependencies) or compare the tuple i with the tuple i+1 or the tuple i with the tuples j where $j > i$... The semantics widely differ depending on these characteristics.
- The predicates occurring in the "if $Pred_1$ is true then $Pred_2$ is true" part of the rule: Predicates $Pred_1$ and $Pred_2$ are defined on a set of attributes and a subset of the relation. Note that these two predicates can be the same. For example, for functional dependencies, the predicates are the same and can be formulated as: [$\forall A \in X, t_1[A] = t_2[A]$], where t_1, t_2 are two tuples, and X is a subset of attributes. These predicates really give the meaning of the semantics.

A "generic" definition of a semantics based on these three components, is described in the sequel.

3.1 Generic definition of a semantics

Given a relation r, the *satisfaction* of a rule $X \rightarrow Y$ in r for a semantics s, noted $r \models_s X \rightarrow Y$, can be defined in a general way as follows:

Definition 1. *Let $X, Y \subseteq U$ and r a relation over U. The* satisfaction *of the rule $X \rightarrow Y$ in r for a semantics s, noted $r \models_s X \rightarrow Y$, is defined by:*
$r \models_s X \rightarrow Y$ if and only if $\forall r' \subseteq r$ verifying $d_c(r')$, if $Pred_1(X, r')$ is true then $Pred_2(Y, r')$ is true where:

1. *$d_c(r')$ specifies a constraint which has to be verified by $r' \subseteq r$.*
2. *$Pred_1(X, r')$ (resp. $Pred_2(Y, r')$) is a predicate specifying a condition on X (resp. Y) over r'.*

A semantics is thus characterized by a constraint d_c defined on a subset of tuples and by two predicates $Pred_1$ and $Pred_2$ defined for a subset of attributes.

Predicates are logical expressions defined on X and r' only and return true or false. The formal description of these logical expressions is somewhat tedious and will be omitted. We prefer to give the intuition through examples.

Example 5. The semantics s_1 presented in exemple 1 can be characterized by the constraint d_c and the predicate $Pred$ defined as follows ($Pred = Pred_1 = Pred_2$):

1. $d_c(r') = [\ r' = \{t\}$ with $t \in r\ ]$.
2. $Pred(X, \{t\}) = [\ \forall A \in X, \epsilon_1 \leq t[A] \leq \epsilon_2\ ]$.

In the sequel, we shall say that a semantics s *complies* with definition 1 if s can be syntactically expressed within the setting of definition 1.

Moreover, we shall note by C the class of rule semantics complying with definition 1.

It is worth noting that we do not integrate quality or error measures of a rule in the "core" definition of a semantics. Roughly speaking, two types of measures can characterize rules, error measures and quality measures:

- Error measures like for example confidence defined for association rules or error indications defined for functional dependencies [13], allow to append approximate rules to exact rules, i.e. those rules which are almost satisfied. These measures are very interesting since they allow to take into account noise in data.
- Quality measures like support, dependency or informative rate [1, 14], allow at contrary to limit the number of rules and possibly to sort out the obtained rules. These measures allow to give to the experts the rules which seem to be the most surprising, the most "interesting" with regard to the chosen statistical criteria.

They can be integrated a posteriori to sort and to qualify the rules. Those error and quality measures will not be discussed anymore in the rest of this paper.

Example 6. The semantics s_2 belongs to C since it can be characterized by the constraint d_c and the predicate $Pred$ defined as follows ($Pred = Pred_1 = Pred_2$):

1. $d_c(r') = [\ r' = \{t_i, t_{i+1}\}$ with $t_i, t_{i+1} \in r\ ]$.
2. $Pred(X, \{t_i, t_{i+1}\}) = [\ \forall A \in X, \epsilon_1 \leq t_{i+1}[A] - t_i[A] \leq \epsilon_2\ ]$.

Example 7. The semantics s_d belongs to C since it can be characterized by the constraint d_c and the predicate $Pred$ defined as follows ($Pred = Pred_1 = Pred_2$):

1. $d_c(r') = [\ r' = \{t_i, t_j\}$ with $t_i, t_j \in r\ ]$.
2. $Pred(X, \{t_i, t_j\}) = [\ \epsilon_1 \leq d(t_i[X], t_j[X]) \leq \epsilon_2\ ]$.

Example 8. The semantics s_{mvd} of multivalued dependencies belongs to C since it can be characterized by the constraint d_c and the predicates $Pred_1$ and $Pred_2$ defined as follows:

1. $d_c(r') = [\ r' = \{t_i, t_j\}$ with $t_i, t_j \in r\ ]$.
2. $Pred_1(X, \{t_i, t_j\}) = [\ \forall A \in X, t_i[A] = t_j[A]\ ]$,
3. $Pred_2(X, \{t_i, t_j\}) = [\ \forall A \in X, t_i[A] = t_j[A]$ or $\forall A \in U \setminus X, t_i[A] = t_j[A]\ ]$.

To conclude, the definition 1 of a semantics has been devised the more naturally possible to capture a great variety of rules. Nevertheless, this definition does not accept everything as shown in the following example:

Example 9. Consider the semantics of inclusion dependencies defined as follows:
$r \models R[X] \subseteq R[Y]$ if and only if $\forall t \in r, \exists t' \in r$ such that $t[X] = t'[Y]$.
Clearly, inclusion dependencies satisfaction cannot be expressed within definition 1, thus this semantics does not belong to C.

3.2 Framework of well-formed semantics

A general framework can be borrowed from theoretical investigations performed over functional dependencies and Armstrong's axiom system [15, 16]. This framework allows to resume interesting properties defined for functional dependencies like reasoning on rules and generating covers for rules.

To be sure that a semantics fulfills this framework, the notion of well-formed semantics can be defined as follows:

Definition 2. *A semantics s is* well-formed *if Armstrong's axiom system is sound and complete for s.*

Let us recall the Armstrong's axiom system for a set of rules F defined over a set of attributes U:

1. (reflexivity) if $X \subseteq Y \subseteq U$ then $F \vdash Y \rightarrow X$
2. (augmentation) if $F \vdash X \rightarrow Y$ and $W \subseteq U$, then $F \vdash XW \rightarrow YW$
3. (transitivity) if $F \vdash X \rightarrow Y$ and $F \vdash Y \rightarrow Z$ then $F \vdash X \rightarrow Z$

The notation $F \vdash X \rightarrow Y$ means that a proof of $X \rightarrow Y$ can be obtained using Armstrong's axiom system from F. Moreover, given a semantics s, the notation $F \models_s X \rightarrow Y$ means that for all relations r over U, if $r \models_s F$ then $r \models_s X \rightarrow Y$.

In other words, for any well-formed semantics s, $\vdash$ and $\models_s$ coincide.

To know whether or not a given semantics is well-formed, we have to give a proof of the soundness and the completeness of the Armstrong's axiom system for this semantics.

This proof being not always trivial, the idea is to find further syntactic restrictions on rule satisfaction definition in order to ensure the well-formedness of the semantics.

4 More syntactic restrictions

In this setting, we propose some syntactic restrictions on the definition 1, which ensure that a semantics is well-formed. In other words, given a new semantics, we do not have to prove anything to be sure that Armstrong's axioms apply: It is just enough that the semantics complies with these syntactic restrictions.

Definition 3. *Let $X, Y \subseteq U$ and r a relation over U. The* satisfaction *of the rule $X \rightarrow Y$ in r for a semantics s, noted $r \models_s X \rightarrow Y$, is defined by:*
$r \models_s X \rightarrow Y$ if and only if $\forall r' \subseteq r$ verifying $d_c(r')$, if $\forall A \in X$, $Pred(A, r')$ is true then $\forall A \in Y$, $Pred(A, r')$ is true where:

1. *$d_c(r')$ specifies a constraint which has to be verified by $r' \subseteq r$.*
2. *$Pred(A, r')$ is a predicate specifying a condition on A over r'.*

The first item of this new semantic definition does not change with regard to the definition 1. The difference is twofold:

- Firstly, the two predicates $Pred_1$ and $Pred_2$ are the same.
- Secondly, a restriction is posed on the predicate: Now, it must be satisfied for each *single attribute* $A \in X$ instead of being satisfied for the *subset of attributes* X.

In the sequel, we shall note by C_A the class of rule semantics complying with definition 3, C_A being a subset of C.

4.1 Usefulness of these syntactic restrictions

The main result of the paper gives an equivalence between well-formed semantics and semantics complying with definition 3 and is stated as follows:

Theorem 1. *Let $s \in C$ be a rule semantics. The semantics s is well-formed if and only if $s \in C_A$.*

Proof. Let $s \in C$ be a rule semantics. We have first to prove that if $s \in C_A$ then s is well-formed and secondly that if s is well-formed then $s \in C_A$ or equivalently that if $s \notin C_A$ then s is not well-formed.

Lemma 1. *Let $s \in C$ be a rule semantics. If $s \in C_A$ then the semantics s is well-formed.*

Proof. (If) Suppose that $s \in C$ i.e. s complies with definition 3, we have to show that s is well-formed i.e. that Armstrong's axiom system is sound and complete for s.

Lemma 2. *Armstrong's axiom system is sound for s.*

Proof. Let F be a set of rules over U. We need to show that if $F \vdash X \rightarrow Y$ then $F \models_s X \rightarrow Y$, i.e. let r be a relation over U such that $r \models_s F$, if $F \vdash X \rightarrow Y$ then $r \models_s X \rightarrow Y$.

1. (reflexivity) We have to show that if $X \subseteq Y \subseteq U$ then $r \models_s Y \rightarrow X$.
   Let $X \subseteq Y \subseteq U$ and let $r' \subseteq r$ verifying $d_c(r')$ and $\forall A \in Y$, $Pred(A, r')$ is true. Since $X \subseteq Y$, the result follows.
2. (augmentation) We have to show that if $F \vdash X \rightarrow Y$ and $W \subseteq U$, then $r \models_s XW \rightarrow YW$.
   Let $r' \subseteq r$ verifying $d_c(r')$ and $\forall A \in X \cup W$, $Pred(A, r')$ is true. If $F \vdash X \rightarrow Y$, then we have $\forall B \in Y$, $Pred(B, r')$ is true. The result follows.
3. (transitivity) We have to show that if $F \vdash X \rightarrow Y$ and $F \vdash Y \rightarrow Z$ then $r \models_s X \rightarrow Z$.
   Let $r' \subseteq r$ verifying $d_c(r')$ and $\forall A \in X$, $Pred(A, r')$ is true. If $F \vdash X \rightarrow Y$ and $F \vdash Y \rightarrow Z$, then $\forall B \in Y$, $Pred(B, r')$ is true and $\forall C \in Z$, $Pred(C, r')$ is true respectively. The result follows.

Note here that the proof is indeed possible thanks to the restriction posed on the predicate, satisfied for each single attribute $A \in X$, in definition 3.

Lemma 3. *Armstrong's axiom system is complete for s.*

Proof. We need to show that if $F \models_s X \rightarrow Y$ then $F \vdash X \rightarrow Y$ or equivalently, if $F \nvdash X \rightarrow Y$ then $F \not\models_s X \rightarrow Y$. As a consequence, assuming that $F \nvdash X \rightarrow Y$, it is enough to give a counter-example relation r such that $r \models_s F$ but $r \not\models_s X \rightarrow Y$.

Note here that to exhibit the counter-example r, we do not have to explicitly give the data instance. Using the constraint $d_c(r)$, we just have to build a relation verifying this constraint, whatever the real values are.

Let r be a relation verifying $d_c(r)$ such that $\forall A \in X^+$, $Pred(A, r)$ is true and $\forall B \in U \setminus X^+$, $Pred(B, r)$ is false. Let us recall that $X^+ = \{A \in U \,|\, F \vdash X \rightarrow A\}$. One can note, by the construction of r, that $r \not\models_s V \rightarrow W$ if and only if $V \in X^+$ and $\exists A \in W$ such that $A \in U \setminus X^+$. Otherwise, $r \models_s V \rightarrow W$.

Firstly, we have to show that $r \models_s F$. We suppose the contrary that $r \not\models_s F$ and thus, $\exists V \rightarrow W \in F$ such that $r \not\models_s V \rightarrow W$. It follows by the construction

of r that $V \subseteq X^+$ and $\exists A \in W$ such that $A \in U \setminus X^+$. Since $V \in X^+$, we have $F \vdash X \rightarrow V$ and since $F \vdash V \rightarrow W$, we have $F \vdash V \rightarrow A$. Thus, by the transitivity rule, $F \vdash X \rightarrow A$ and thus $A \in X^+$. This leads to a contradiction since $A \in W$, and thus $r \models_s F$.

Secondly, we have to show that $r \not\models_s X \rightarrow Y$. We suppose the contrary that $r \models_s X \rightarrow Y$. It follows by the construction of r that $Y \subseteq X^+$ and thus $F \vdash X \rightarrow Y$. It leads to a contradiction since $F \nvdash X \rightarrow Y$ was assumed, and thus $r \not\models_s X \rightarrow Y$.

The second part of the proof is certainly the more surprising one since it tells us that any well-formed semantics has to comply with definition 3:

Lemma 4. *Let $s \in C$ be a rule semantics. If the semantics s is well-formed, then $s \in C_A$.*

Proof. (Only if) We have to show that if s is well-formed, then $s \in C_A$ or equivalently, if $s \notin C_A$ (i.e. s does not comply with definition 3) then s is not well-formed.

To complete the proof, we need to precisely define what $Pred_1 = Pred_2$ means since two different definitions of predicates can be equivalent. Intuitively, the two predicates are equivalents if for any relation r and for any subset of attributes $X \subseteq U$, $Pred_1$ and $Pred_2$ are satisfied for X on the same subsets of r. Here is the formal definition:

Definition 4. *Two predicates $Pred_1$ and $Pred_2$ are said to be equivalents, denoted by $Pred_1 = Pred_2$, if and only if for any relation r and for any subset of attributes $X \subseteq U$, we have:*
$\{r' \subseteq r \mid r'$ verifying $d_c(r')$ and $Pred_1(X, r')$ is true$\} = \{r' \subseteq r \mid r'$ verifying $d_c(r')$ and $Pred_2(X, r')$ is true$\}$.

Suppose that s does not comply with definition 3, two cases are possible: Either the two predicates are different or they are equal but it does not exist an equivalent predicate which can be formulated as a condition on each attribute.

Let us consider the first case, i.e. $Pred_1 \neq Pred_2$: In that case, it does exist a relation r and a subset of attributes $Y \subseteq U$ such that $\{r' \subseteq r \mid r'$ verifying $d_c(r')$ and $Pred_1(Y, r')$ is true$\} \neq \{r' \subseteq r \mid r'$ verifying $d_c(r')$ and $Pred_2(Y, r')$ is true$\}$.

Two cases are thus possible:

- $\exists r' \subseteq r$ verifying $d_c(r')$ such that $Pred_1(Y, r')$ is true and $Pred_2(Y, r')$ is false:
  Let us assume that s is well-formed. By reflexivity, we have $\forall X \subseteq Y$, $r \models Y \rightarrow X$ and thus $r \models Y \rightarrow Y$ i.e. $\forall r' \subseteq r$ verifying $d_c(r')$, if $Pred_1(Y, r')$ is true then $Pred_2(Y, r')$ is true which is a contradiction.

– $\exists r' \subseteq r$ verifying $d_c(r')$ such that $Pred_1(Y, r')$ is false and $Pred_2(Y, r')$ is true:
Without loss of generality, let us assume there exists $X \in U \setminus Y$ and $Z \in U \setminus Y$ such that $Pred_1(X, r')$ is true and $Pred_2(Z, r')$ is false, as depicted in Table 1. Thus, we have $r' \models X \rightarrow Y$ and $r' \models Y \rightarrow Z$.

r	X	Y	Z	...
...	...	...	...	...
$r'\{$	$Pred_1$ true	$Pred_1$ false $Pred_2$ true	$Pred_2$ false	...
...	...	...	...	...

Table 1. Example

Assume now that s is well-formed. By transitivity, we should have $r' \models X \rightarrow Z$, which is false and leads to a contradiction.

Finally, we have shown that if $Pred_1 \neq Pred_2$, s is not-well-formed.

Now, let us consider the second case, i.e. $Pred_1 = Pred_2$ but $Pred_1 \neq Pred'$ with $Pred'(Y) = [\forall A \in Y, Pred_1(A)]$: In that case, it does exist a relation r and a subset of attributes $Y \subseteq U$ such that $\{r' \subseteq r \mid r'$ verifying $d_c(r')$ and $Pred_1(Y, r')$ is true$\} \neq \{r' \subseteq r \mid r'$ verifying $d_c(r')$ and $\forall A \in Y, Pred_1(A, r')$ is true$\}$. We can show that reflexivity and transitivity axioms are not sound in r'. The proof is equivalent to the previous one and is omitted.

□ This concludes the proof of theorem 1.

This theorem shows that among semantics complying with definition 1, only those complying with definition 3 are well-formed and any well-formed semantics can be expressed within the syntactic restrictions given in definition 3.

Example 10. The semantics s_1 (see example 5), can be characterized by the constraint d_c and the predicate $Pred$ defined as follows:

1. $d_c(r') = [\ r' = \{t\}$ with $t \in r\ ]$.
2. $Pred(A, \{t\}) = [\ \epsilon_1 \leq t[A] \leq \epsilon_2\ ]$.

Thus, the semantics $s_1 \in C_A$ and this is sufficient from Theorem 1, to prove the following result:

Corollary 1. *The semantics s_1 is* well-formed.

Example 11. The semantics s_2 (see example 6), can be characterized by the constraint d_c and the predicate $Pred$ defined as follows:

1. $d_c(r') = [\ r' = \{t_i, t_{i+1}\}$ with $t_i, t_{i+1} \in r\ ]$.

2. $Pred(A, \{t_i, t_{i+1}\}) = [\ \epsilon_1 \leq t_{i+1}[A] - t_i[A] \leq \epsilon_2\ ]$.

Thus, the semantics $s_2 \in C_A$ and this is sufficient from Theorem 1, to prove the following result:

Corollary 2. *The semantics s_2 is* well-formed.

Example 12. For the semantics s_d (see example 7), we have the following result:

Corollary 3. *The semantics s_d is not* well-formed.

Proof. We have to show that the semantics s_d does not comply with definition 3. The semantics s_d complies with definition 1, but the predicate $Pred(X, \{t_i, t_j\}) = [\ \epsilon_1 \leq d(t_i[X], t_j[X]) \leq \epsilon_2\ ]$, for $X \subseteq U$ is obviously different from the predicate $Pred'(X, \{t_i, t_j\}) = [\ \forall A \in X, \epsilon_1 \leq d(t_i[A], t_j[A]) \leq \epsilon_2\ ]$. It is easy to give a counter example where $Pred(X, r)$ is true and $Pred'(X, r)$ is false. Thus, $s_d \notin C_A$ and by Theorem 1, the result follows.

Example 13. For the semantics s_{mvd} (see example 8) of multivalued dependencies, we have the following result:

Corollary 4. *The semantics s_{mvd} is not* well-formed.

Proof. We have to show that the semantics s_{mvd} does not comply with definition 3. The semantics s_{mvd} complies with definition 1, but the two predicates are obviously different. Thus, $s_{mvd} \notin C_A$ and by Theorem 1, the result follows.

4.2 A simple remark

For any rule semantics s complying with definition 1 but not well-formed, it is possible to derive a semantics s' from s as follows:

Definition 5. *$r \models_{s'} X \rightarrow Y$ if and only if $\forall r' \subseteq r$ verifying $d_c(r')$, if $\forall A \in X$, $Pred_1(A, r')$ is true then $\forall A \in Y$, $Pred_1(A, r')$ is true.*

The unique change is on the scope of $Pred_1$ which has to be satisfied for each $A \in X$ instead of being satisfied for $X \subseteq U$.

Moreover, if $Pred_1 \neq Pred_2$, we can also defined a semantics s'' with $Pred_2$ instead of $Pred_1$:

Definition 6. *$r \models_{s''} X \rightarrow Y$ if and only if $\forall r' \subseteq r$ verifying $d_c(r')$, if $\forall A \in X$, $Pred_2(A, r')$ is true then $\forall A \in Y$, $Pred_2(A, r')$ is true.*

Since s' and s'' comply with definition 3, we have the following result:

Corollary 5. *The semantics s' and s'' are* well-formed.

This corollary shows that for each semantics not well-formed, corresponds one or two well-formed semantics. The number of well-formed semantics which can be defined is thus quite important.

Example 14. We have shown the semantics $s_d \in C \setminus C_A$. Let us consider the semantics s'_d characterized by the constraint d_c and the predicate $Pred$ defined as follows:

1. $d_c(r') = [\ r' = \{t_i, t_j\}$ with $t_i, t_j \in r\ ]$.
2. $Pred(A, \{t_i, t_j\}) = [\ \epsilon_1 \leq d(t_i[A], t_j[A]) \leq \epsilon_2\ ]$.

By Theorem 1, we have the following result:

Corollary 6. *The semantics s'_d is* well-formed.

Example 15. From the semantics $s_{mvd} \in C \setminus C_A$, we can define the two following semantics:

- s'_{mvd} characterized by the constraint d_c and the predicate $Pred$ defined as follows:
  1. $d_c(r') = [\ r' = \{t_i, t_j\}$ with $t_i, t_j \in r\ ]$.
  2. $Pred(A, \{t_i, t_j\}) = [\ t_i[A] = t_j[A]\ ]$.

  Note that this new semantics turns out to be the satisfaction of functional dependencies.
- s''_{mvd} characterized by the constraint d_c and the predicate $Pred$ defined as follows:
  1. $d_c(r') = [\ r' = \{t_i, t_j\}$ with $t_i, t_j \in r\ ]$.
  2. $Pred(A, \{t_i, t_j\}) = [\ t_i[A] = t_j[A]$ or $\forall B \in U \setminus A, t_i[B] = t_j[B]\ ]$

By Theorem 1, we have the following result:

Corollary 7. *The semantics s'_{mvd} and s''_{mvd} are* well-formed.

5 Some relationships with data mining

In a KDD context, the discovery of rules in tabular data has mainly been studied in the context of association rules for binary data [1] and functional dependencies [16–18]. In the setting of this paper, we may question about the underlying discovery problem for any well-formed semantics.

We are only interested in the context of this paper of semantics $s \in C_A$ i.e. the well-formed semantics. Rule generation for a semantics $s \in C \setminus C_A$ is also interesting but out of the scope of this paper.

First, recall that a one-to-one correspondence does exist between a set of rules and a closure system[3] [19]. Second, given a relation r and a well-formed semantics s, we can always define a closure system with respect to the set of satisfied rules in the relation r for the semantics s.

Two main techniques do exist to compute a cover of rules:

[3] A closure system C on U is such that $U \in C$ and $\forall X, Y \in C$, $X \cap Y \in C$.

– Those which enumerate the closure system to generate for example a minimum cover [6], generally used for association rules generation [20].
– Those which avoid the enumeration of the closure system to generate the canonical cover, generally used for the inference of functional dependencies [15, ?].

In both cases, the first step is to compute a *base*[4] of the closure system in the dataset, data accesses being made only during this step. This step is obviously specific to the considered semantics.

In this paper, we show how a base of the closure system for any well-formed semantics can be computed from the dataset.

First, we define a *base* with regard to our generic definition of rule semantics and then we give a proof that this is indeed a base:

Definition 7. *Let r a relation over U and s a given well-formed semantics. Let $B_s(r)$ the set defined as follows:*

$$B_s(r) = \bigcup_{r' \subseteq r \mid d_c(r')} \{A \in U \mid Pred(A, r') \text{ is true}\}.$$

We have the following result which extends in our context a well-known result obtained in the setting of functional dependencies [21]:

Proposition 1. *$B_s(r)$ is a* base *of the closure system with respect to the set $F_s(r)$ of satisfied rules in r for the semantics s.*

Proof. Let us recall that a sub-family B of a closure system C is a *base* if $Inf \subseteq B \subseteq C$ where Inf is the set of meet-irreducible sets.

We first prove that $B_s(r) \subseteq C(F_s(r))$ and then that $Inf_s(r) \subseteq B_s(r)$, where $Inf_s(r)$ is the set of meet-irreducible sets of the closure system $C(F_s(r))$.

Let $X \in B_s(r)$, we have to prove that $X \in C(F_s(r))$ i.e. that $X = X^+$. Suppose to the contrary that $\exists A \in U \setminus X$ such that $r \models_s X \rightarrow A$. In that case, $\forall r' \subseteq r$ verifying $d_c(r')$ such that $\forall B \in X, Pred(B, r')$ is true, we must have $Pred(A, r')$ is true. That leads to a contradiction, since $X \in B_s(r)$.

Let $X \in Inf_s(r)$, we have to prove that $X \in B_s(r)$. By definition, $X = X^+$ and thus, for any $A \in U \setminus X$, $r \not\models_s X \rightarrow A$ i.e. for any $A \in U \setminus X$, $\exists Y_A \in B_s(r)$ such that $X \subseteq Y_A$ and $A \notin Y_A$. As above, $Y_A \in C(F_s(r))$ for all $A \in U \setminus X$. Now, we have: $X = \bigcap_{A \in U \setminus X}(Y_A)$ and since $X \in Inf_s(r)$, we must have $X = Y_A$ for some $A \in U \setminus X$. Hence, we have $X \in B_s(r)$.

□

We can note that the base $B_s(r)$ is defined at the level of single attributes $A \in U$, which shows the necessity that the semantics complies with the definition 3.

[4] Also called *agree sets* for functional dependencies.

From a data mining point of view, the computation of $B_s(r)$ is a crucial step since data accesses are only performed here.

Example 16. Consider the semantics s_1 and the relation r made of 5 tuples over a set of 6 attributes, given in Table 2.

r	g_1	g_2	g_3	g_4	g_5	g_6
t_1	1.7	1.5	1.2	-0.3	1.4	1.6
t_2	1.8	-0.7	1.3	0.8	-0.1	1.7
t_3	-1.8	0.4	1.7	1.8	0.6	-0.4
t_4	-1.7	-1.4	0.9	0.5	-1.8	-0.2
t_5	0.0	1.9	-1.9	1.7	1.6	-0.5

Table 2. A running example

The base of the closure system $B_s(r)$ is computed as follows:

$$B_s(r) = \bigcup_{t \in r} \{ A \in U \mid 1.0 \leq t[A] \leq 2.0 \}.$$

For this relation r, we have:

$$B_s(r) = \{\{g_1, g_2, g_3, g_5, g_6\}, \{g_1, g_3, g_6\}, \{g_3, g_4\}, \{\}, \{g_2, g_4, g_5\}\}.$$

For example, a minimal cover of satisfied rules can be computed from $B_s(r)$ to give the following rules:
$\{g_1 \rightarrow \{g_3, g_6\}$, $\{g_1, g_4\} \rightarrow g_2$, $g_2 \rightarrow g_5$, $\{g_2, g_3\} \rightarrow g_1$, $g_5 \rightarrow g_2$, $g_6 \rightarrow g_1\}$.

6 Related contributions

Rule discovery is very popular in data mining with association rules [1, 22] and in databases with functional dependencies [17, 18, 23].

Rule mining often results in a huge amount of rules and as a consequence rules turn out to be useless for experts. This is the well-known post-processing step in a KDD process. To address this problem, different lines of research have been performed.

Firstly, rules may be filtered out a priori, based on user-defined templates of rules [24, 25]. Secondly, many quality measures have been developed to select only the most interesting rules [26, 14] with regard to these measures. Thirdly, inference rules or inference systems have been proposed to reduce the number of rules given to the experts [22, 27–29]. Most of these works consider new inference systems for association rules by taking into account support and confidence thresholds.

In [12], we proposed three new semantics for gene expression data and show their biological interests. In [30, 3], we did a first step towards a formal framework with a first generic definition of a semantics and the notion of a well-formed semantics. In this paper, we go a step beyond by giving a syntactical characterisation of a well-formed semantics.

7 Conclusion

In this paper, we have pointed out that an equivalence does exist between some syntactic restrictions on the natural definition of a given semantics and the fact that this semantics is well-formed. From a practical point of view, this equivalence allows to prove easily that a new semantics is well-formed.

We have illustrated our proposition on many examples of semantics and have shown the usefulness of our proposition as a tool to ensure that classical reasoning on rules with Armstrong's axioms is indeed possible.

We have also pointed out the relationship between our generic definition of rule satisfaction and the underlying data mining problem.

In the context of our application on gene expression data, this work brings some foundations to build new well-formed semantics with biologists which best fit into their requirement. Moreover, the rule discovery process has to be revisited to avoid a costly (exponential in the number of genes) and useless (too many rules being generated to be validated by biologists) generation of rules.

References

1. Agrawal, R., Imielinski, T., Swami, A.N.: Mining association rules between sets of items in large databases. In: Proc. of the ACM SIGMOD, Washington D.C. (1993) 207–216
2. Armstrong, W.W.: Dependency structures of data base relationships. In: Proc. of the IFIP Congress 1974. (1974) 580–583
3. Agier, M., Petit, J.M., Suzuki, E.: Towards ad-hoc rule semantics for gene expression data. In: Proc. of the ISMIS, Saratoga Springs, New-York, USA, Springer-Verlag (2005)
4. Beeri, C., Berstein, P.: Computational problems related to the design of normal form relation schemes. ACM TODS **4** (1979) 30–59
5. Maier, D.: Minimum covers in the relational database model. JACM **27** (1980) 664–674
6. Guigues, J.L., Duquenne, V.: Familles minimales d'implications informatives résultant d'un tableau de données binaires. Math. Sci. Humaines **24** (1986) 5–18
7. Gottlob, G., Libkin, L.: Investigations on Armstrong relations, dependency inference, and excluded functional dependencies. Acta Cybernetica **9** (1990) 385–402
8. Becquet, C., Blachon, S., Jeudy, B., Boulicaut, J.F., Gandrillon, O.: Strong-association-rule mining for large-scale gene-expression data analysis: a case study on human sage data. Genome Biology **3** (2002)

9. Icev, A., Ruiz, C., Ryder, E.F.: Distance-enhanced association rules for gene expression. In: BIOKDD'03, in conjunction with ACM SIGKDD, Washington, DC, USA. (2003)
10. Creighton, C., Hanash, S.: Mining gene expression databases for association rules. Bioinformatics **19** (2003) 79–86
11. Cong, G., Tung, A.K.H., Xu, X., andJiong Yang, F.P.: Farmer: Finding interesting rule groups in microarray datasets. In: Proc. of the ACM SIGMOD. (2004) 143–154
12. Agier, M., Petit, J.M., Chabaud, V., Pradeyrol, C., Bignon, Y.J., Vidal, V.: Vers différents types de règles pour les données d'expression de gènes-Application à des données de tumeurs mammaires. In: Actes du Congrès INFORSID'04, Biarritz. (2004) 351–367
13. Kivinen, J., Mannila, H.: Approximate inference of functional dependencies from relations. TCS **149** (1995) 129–149
14. Tan, P.N., Kumar, V., Srivastava, J.: Selecting the right objective measure for association analysis. Information Systems **29** (2004) 293–313
15. Mannila, H., Räihä, K.J.: Algorithms for inferring functional dependencies from relations. DKE **12** (1994) 83–99
16. Demetrovics, J., Thi, V.: Some remarks on generating Armstrong and inferring functional dependencies relation. Acta Cybernetica **12** (1995) 167–180
17. Huhtala, Y., Kärkkäinen, J., Porkka, P., Toivonen, H.: Efficient discovery of functional and approximate dependencies using partitions. In: Proc. of the 14^{th} IEEE ICDE. (1998) 392–401
18. Lopes, S., Petit, J.M., Lakhal, L.: Functional and approximate dependencies mining: Databases and FCA point of view. JETAI **14** (2002) 93–114
19. Ganter, B., Wille, R.: Formal Concept Analysis. Springer-Verlag (1999)
20. Zaki, M.J.: Generating non-redundant association rules. In: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM Press (2000) 34–43
21. Beeri, C., Dowd, M., Fagin, R., Statman, R.: On the structure of Armstrong relations for functional dependencies. JACM **31** (1984) 30–46
22. Bastide, Y., Pasquier, N., Taouil, R., Stumme, G., Lakhal, L.: Mining minimal non-redundant association rules using frequent closed itemsets. In: Proc. of the 1^{st} Computational Logic, London, UK. Volume 1861. (2000) 972–986
23. Novelli, N., Cicchetti, R.: Fun: An efficient algorithm for mining functional and embedded dependencies. In: Proc. of the ICDT, London, UK. Volume 1973 of LNCS., Springer-Verlag (2001) 189–203
24. Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H., Verkamo, A.: Finding Interesting Rules from Large Sets of Discovered Association Rules. In: Proc. of the 3^{td} CIKM. (1994) 401–407
25. Baralis, E., Psaila, G.: Designing templates for mining association rules. J. Intell. Inf. Syst. **9** (1997) 7–32
26. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proc. of the 20^{th} VLDB, Santiago de Chile, Chile. (1994) 487–499
27. Cristofor, L., Simovici, D.A.: Generating an informative cover for association rules. In: Proc. of the IEEE ICDM, Maebashi City, Japan. (2002) 597–600
28. Li, G., Hamilton, H.: Basic association rules. In: Proc. of the 4^{th} SIAM ICDM, Lake Buena Vista, Florida, USA. (2004)
29. Luong, V.P.: The representative basis for association rules. In: Proc. of the IEEE ICDM. (2001) 639–640
30. Agier, M., Petit, J.M.: Notion de sémantiques bien-formées pour les règles. In: Actes de la conférence EGC. Volume 1. (2005) 19–30