PUC

Monografias em Ciência da Computação
nº 15/92

Runs, Conditions and Problems: An Algebraic Approach

Maria Claudia Meré
Paulo A. S. Veloso

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO
RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22453
RIO DE JANEIRO - BRASIL

PUC RIO - DEPARTAMENTO DE INFORMÁTICA

Monografias em Ciência da Computação, Nº 15/92
Editor: Carlos J. P. Lucena

Maio, 1992

Runs, Conditions and Problems: An Algebraic Approach *

Maria Claudia Meré
Paulo A. S. Veloso

* This work has been sponsored by the Secretaria de Ciência e Tecnologia da Presidência da República Federativa do Brasil.

Abstract

The purpose of this paper is to analyze the algebraic structure of sets of functional complete subproblems of relations. The motivation was due to the fact that these functions correspond to the **runs** of a *virtual machine* when fed with data belonging to an *application domain.* We also discuss the relationship between the presented structure and the algebraic structure of *partial relations.*

KEY WORDS

functional complete subproblem, partial binary relation, program run, viable problem, Skolem function.

Resumo

O propósito deste trabalho é analisar a estrutura algébrica dos conjuntos dos subproblemas funcionáis completos de relações. A motivação deve-se ao fato de que estas funções correspondem às **corridas** de uma *máquina virtual* quando alimentada com dados pertencentes a um *dominio de aplicação.* Também, discutimos a relação entre a estrutura apresentada e a estrutura algébrica das *relações parciais.*

PALAVRAS CHAVES

Subproblema funcional completo,relação binaria parcial, corrida de um programa, problema viável, função de Skolem.

Contents

Acknowledgements

We wish to thank Professor Armando M. Haeberer for helpful discussions and suggestions during the elaboration of this work.

1 Introduction

The purpose of this paper is to analyze the algebraic structure of the functions corresponding to the **runs** of a *virtual machine* (by convention a virtual machine m_{pH} means the device build up by a given program p, an interpreter h, and the piece of hardware H acting as the object machine of such an interpreter) when fed with data belonging to an *application domain*, i.e. acceptable data as inputs for the "real problem" in consideration.

Following [HV90b] we consider that an *application* A means the extensional knowledge (the input-output information) about the "real problem". Thus, inputs and outputs are any pair of "observable events" related by the application, in the sense of belonging to the extension of the real problem. Also, we take from [HV90b] the relation of *being-an-engineering-model* which connects a virtual machine m_{pH} to an application A (denoted $m_{pH} < A$) and it is defined through an operation that involves a systematic activity with the following steps :

- An input δ from the domain A is selected and introduced into the machine m_{pH}, then, if m_{pH} does not halt, it is not the case that $m_{pH} < A$.
- If m_{pH} halts but the output is not an acceptable output for A corresponding to the input δ then $\neg(m_{pH} < A)$.
- If m_{pH} halts and it gives an acceptable output then, we may assume $m_{pH} < A$ (this is the "correspondent concept" in the Observational Layer to the total correctness in the Syntactic Layer).

Actually, there exists a translation-abstraction function g between data and result of A and m_{pH}, but this is of no importance to our subject.

For the present discussion we will accept the scenario described in [HV90b], [HV90c] and [HV90a], which is despicted in figure 1.

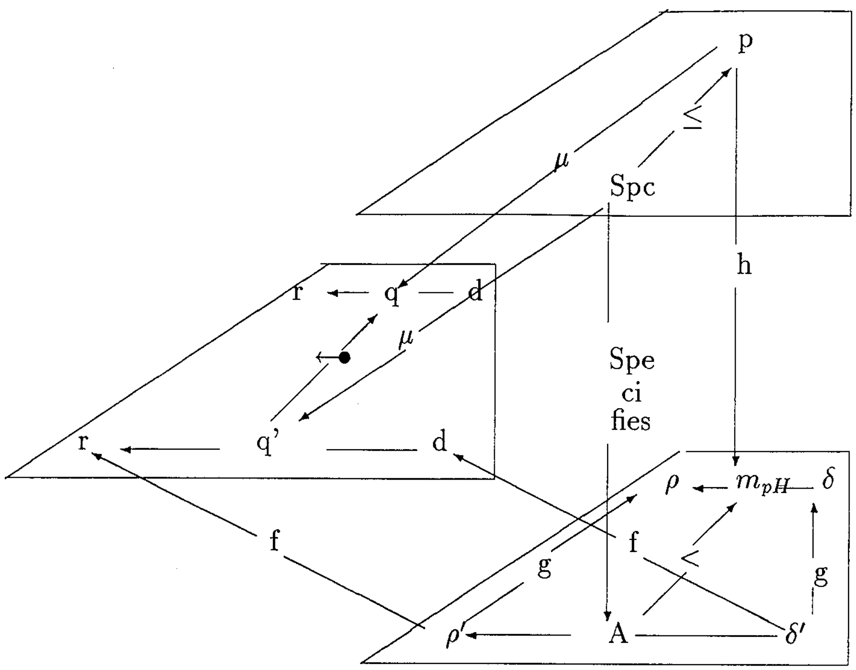

Figure 1

In figure 1 we consider an application A with some known instances (δ', ρ'), specified by a specification Spc denoting a relation q' build up by pairs (d, r) which are, also, the component-wise translation of (δ', ρ') through a given function f.

It is important to notice that the application A is an observable object, which can be (and almost always is) ill-determined because of lack of knowledge about its extension. If we knew the whole application it would perform a relation, but, because we have a partially known object we are unable to define the relation it performs. We permit ourselves to consider the relation performed by an application A in order to clarify some ideas.

The translation-abstraction functions (i.e. f, g) do the needed connection between inputs-outputs of different layers or between different abstraction levels of the same layer. Because we are not going to deal with these connections and they don't play any role in our discussion, we work in a loosely way forgetting the effect of these functions.

As it was discussed in [HV90c] and [VH], in the Syntactic Layer we have terms of the algebra $U =< P_u, +, \bullet, -, \infty, 1, 1_B, 1_\lambda, ;, \sim, \nabla, \times >$, which denote partial relations in the Semantic Layer. Thus, Spc and p are terms of U, p involving only algorithmic operations and relational constants "understandable" by h. It should be noticed that $q = \mu[p] \longleftarrow\!\!\bullet \mu[Spc] = q'$ (this means that q is a subrelation of the relation q' and $Dom(q) = Dom(q')$, in this case we say that q is a *complete subproblem* of q') is the semantic counterpart of the relation $p \leq Spc$, where $\leq$ denotes total correctness. This is so, because μ is the unique homomorphism assigning to each term of the Syntactic Layer the problem it describes (in the Semantic Layer),i.e., μ is the function that assigns one semantic value to each syntatic term, in our case the semantics is given by partial relations which we call *problems*.

Here, "Spc specifies A" means that $q' = \mu[Spc] \longleftarrow\!\!\bullet A$ (where we allow a notational license because, actually, we are refering to the relation realized by A) [HV90b]. On the other hand, $q = \mu[p]$ is the relation realized by m_{pH} and since $q \longleftarrow\!\!\bullet q'$ then, $m_{pH} < A$ denotes, if we admit some notational liberality, $q \longleftarrow\!\!\bullet A$.

Therefore, we will establish the meaning of **run** as follows. If we introduce in a virtual machine m_{pH} a data δ belonging to the domain of an application A, it eventually halts and produces a result ρ, which, if m_{pH} is an engineering model of A, will be an acceptable output for A corresponding to input δ. We have called each pair (δ, ρ) an *instance* of A. Also, the set

$\{(\delta_i, \rho_i) \ : \ i \in I\}$ (where $\{\delta_i : i \in I\} = \{\delta : \varphi(\delta)\}$ if φ is the precondition of p) of pairs resulting of the introduction in m_{pH} of a set $\{\delta_i \ : \ i \in I\}$ of data belonging to the domain of A, will be called a **run** (notice that ρ_i may not be an expected result by the application A). Let us consider, for the sake of simplicity and without any loss of generality, **complete runs**, i.e., **runs** exhausting the set of data denoted by the precondition of p. Each **complete run** of m_{pH} is the set of pairs of a Skolem function of $\mu[p]$ restricted to the precondition of p, and, if $m_{pH} < A$, each **complete run** of m_{pH} is, consequently, the set of pairs (the graph) of a Skolem function of A.

We should notice that what we say above is a somewhat formal non-operational definition of "being-an-engineering-model". Since in the Observational Layer we are dealing with extensional objects, the "image" in this layer of the syntactic correctness relation $p \leq Spc$ (or, what is the same, of the relation $\mu[p] \leftarrow\!\!\bullet \mu[Spc]$, in the Semantic Layer) is the relation of non-strict inclusion of the corresponding sets of graphs of Skolem functions. In other words, if we call $\mathbf{M}$ the set of graphs of Skolem functions of the relation realized by m_{pH},i.e. of q, and $\mathbf{A}$ the set of graphs of Skolem functions of the relation realized by A, then $m_{pH} < A \leftrightarrow \mathbf{M} \subseteq \mathbf{A}$.

Since, as it was discussed in [Leh84] and formally shown in [HV89], testing (specification validation, validation of each one of the products of the different derivation steps, and finally program testing) is inevitable, and it is apparent, on the basis of the above discussion, that the formal analysis of black-box testing should be done on the basis of the algebraic structure of the inclusion of sets of graphs of Skolem functions, it seems to be of great importance to analyze in deep such a structure in itself and its relationship with the algebraic structure of partial relations discussed in [HV90a].

2 Basic Concepts of the Algebraic Theory of Problems

In this section we present some theoretic concepts which are needed for understanding our future developments.

In [HV90c] , [HV90a] is presented a Calculus of Binary Relations in order to develop a Programming Calculus. In those papers, it was shown that if we want to derive programs using partial relations, the needed information we should give about relations can be presented as a 3-tuple $< D, R, q >$ such that $q \subseteq D \times R$. Such objects are called *problems*, since their structure resembles the notion of problem introduced by G. Polya [Pol57] and later developed by P. Veloso [Vel84].

Definition 2.1 A *problem* P over an universal set U (U is the closure of a given set B under $\times$, $+$ and the operation $\bigtriangledown$) is a 3-tuple $P =< D, R, q >$ where:

- D is a nonempty subset of the universe $\mathcal{U}$. We will refer to the set D as *the domain of data.*
- R is a nonempty subset of the universe $\mathcal{U}$. We will refer to the set R as *the domain of results.*
- $q \subseteq D \times R$, *is the condition or requirement of the problem.*

Definition 2.2 We call a problem $P =< D, R, q >$ *viable* iff for every $d \in D$ there is $r \in R$ such that $(d, r) \in q$, that is, the relation q is total over D.

Definition 2.3 We say a problem $P' =< D', R', q' >$ is a *subproblem* of a problem $P =< D, R, q >$ (denoted by $P' \sqsubseteq P$) iff :

- All data of P' is data of P $(D' \subseteq D)$.
- All result of P' is result of P $(R' \subseteq R)$.
- If a result of P' is admissible, then it is admissible in P $(q' \subseteq q)$.

When occurs the special case $D' = D$, $R' = R$, and $q' \subseteq q$, we say that P' is a *proper subproblem* of P.

Definition 2.4 We say a problem $P' =< D', R', q' >$ is a *complete subproblem* of a problem $P =< D, R, q >$ (denoted by $P' \leftarrow\!\bullet P$) iff : $P' \sqsubseteq P$ and $Dom(q') = Dom(q)$.

The notion of complete subproblem involves the idea of total correctness, i.e. it means that all input-output pairs which belong to the problem P' must also belong to the problem P , and all inputs for which the condition q gives a result in R, the condition q' must also give an admissible result in P'. In other words, if P and P' are models of specifications, P is a weaker specification (more general) than P'.

When the special case $D' = D$, $R' = R$, $Dom(q) = Dom(q')$ and $q' \subseteq q$ occurs, we say that P' is a *restriction* of P. If we consider that P and P' are models of (angelic) non-deterministic programs, then P' is a more deterministic program than P. In operational terms, the set of runs of P (the set of Skolem functions of q) includes the set of runs of P' (the set of Skolem functions of q').

Definition 2.5 Let $P =< D, R, q >$, $P' =< D', R', q' >$ and $P'' =< D'', R'', q'' >$ be problems then we say that P is *the addition of* P' *and* P'' (and we writte $P = P' + P''$) iff $D = D' \cup D''$, $R = R' \cup R''$ *and* $q = q' \cup q''$.

Definition 2.6 Let $P =< D, R, q >$, $P' =< D', R', q' >$ and $P'' =< D'', R'', q'' >$ be problems then we say that P is *the intersection of* P' *and* P'' (and we writte $P = P' \diamond P''$) iff $D = D' \cap D''$, $R = R' \cap R''$ *and* $q = q' \cap q''$.

Let $P_{\mathcal{U}}$ be all problems over the universe $\mathcal{U}$ then, $(P_{\mathcal{U}}, \sqsubseteq)$ is a lattice with the addition of two problems as their *l.u.b.* (the *least upper bound*), and the intersection as their *g.l.b.* (the *greatest lower bound*). This lattice is algebraic, distributive, modular, complete, copseudocomplement, is bounded by $< \mathcal{U}, \mathcal{U}, \mathcal{U} \times \mathcal{U} >$ and by $< \emptyset, \emptyset, \emptyset >$ and has atoms (its atoms are problems with the form $0_y =< \emptyset, \{y\}, \emptyset >$ or $0_x =< \{x\}, \emptyset, \emptyset >$) [HV90a].

It is important to notice that whereas in the addition case we have the nice fact that $Dom(p \cup q) = Dom(p) \cup Dom(q)$, in the intersection case, in general, only holds $Dom(p \cap q) \subseteq Dom(p) \cap Dom(q)$ [Sup60]. This has an immediate consequence, for example, that the intersection of two viable problems can not be viable. Furthermore, as is formally shown in [HV90a], there is not a componentwise manner of determining the intersection of two partial relations . This means that the *g.l.b.* of two problems (or two partial relations), as defined above, does not behave exactly as an "expected" *g.l.b.*, in the sense that, from the components of p and q we are not able to determine all of the components of $p \cap q$.

Let P_+ be the set of all subproblems of a given problem P then, $(P_+, \sqsubseteq)$ is a complete lattice. And, let $P_\oplus$ the set of all complete subproblem of a given problem P then, $(P_\oplus, \hookleftarrow\!\!\bullet)$ is an upper semilattice which is an upper subsemilattice of $(P_+, \sqsubseteq)$.

If we call $\Theta = \{0_D =< D, \emptyset, \emptyset > , \forall D \in \mathcal{U}\} \cup \{0^R =< \emptyset, R, \emptyset > , \forall R \in \mathcal{U}\} \cup \{0_D + 0^R , \forall D, R \in \mathcal{U}\}$ then, $(\Theta, \sqsubseteq)$ is a sublattice of $(P_{\mathcal{U}}, \sqsubseteq)$ with first element $< \mathcal{U}, \mathcal{U}, \emptyset >$

and last element $<\emptyset,\emptyset,\emptyset>=0$, [HV90a].

Let $V_{\mathcal{U}}$ be the set of all viable problems in $\mathcal{U}$ then, $(V_{\mathcal{U}},\sqsubseteq)$ is an upper semilattice which is an upper subsemilattice of $(P_{\mathcal{U}},\sqsubseteq)$.

3 Algebraic Structures

3.1 The functions Σ and $\dot{\Sigma}$

Let D and R be a domain of data and a domain of results respectively. We define the function

$\dot{\Sigma}_{D,R} : \mathcal{P}(D \times R) \longrightarrow \mathcal{P}(D \rightharpoonup R)$[1] such that

$$q \longmapsto \dot{\Sigma}_{D,R}(q) = \{f : D \longrightarrow R \;/\; f \subseteq q\,,\;\; Dom(f) = Dom(q)\}.$$

$\dot{\Sigma}_{D,R}(q)$ is the set of functional complete subproblems of the relation $q \subseteq D \times R$, we call it the *set of runs of* q.

Each $q \subseteq D \times R$ corresponds with exactly a problem, the problem $P =< D, R, q >$. Thus, for briefness we will use without distinction the relation $q \subseteq D \times R$ and the problem $P =< D, R, q >$. In the following, we consider arbitrary but fixed D and R, and so, we will write $\dot{\Sigma}(q)$ instead of $\dot{\Sigma}_{D,R}(q)$.

Let's study some properties of the function $\dot{\Sigma}$.

Proposition 3.1 $\dot{\Sigma}(q) \subseteq \dot{\Sigma}(q') \implies q \subseteq q'$.

Proof: Let be $(a,b) \in q \neq \emptyset$ *then, there exists* $f \in \dot{\Sigma}(q) \neq \emptyset$ *s.t.* $f(a) = b$. *Then,* $f \in \dot{\Sigma}(q')$ *and thus,* $(a,b) \in q'$.

If $\dot{\Sigma}(q) = \emptyset \subseteq \dot{\Sigma}(q')$ *then,* $q = \emptyset \subseteq q'$. □

Corollary 3.2 $\dot{\Sigma}$ *is an 1-1 map.*

[1]Here $D \rightharpoonup R$ is a set of all (partial and total) functions from D to R.

Proposition 3.3 *If $q \subseteq q'$ and $Dom(q) = Dom(q')$ then $\dot{\Sigma}(q) \subseteq \dot{\Sigma}(q')$.*

Proof : *Let be $f \in \dot{\Sigma}(q)$ and $q \neq \emptyset$ then, $f \subseteq q \subseteq q'$, and $f : D \longrightarrow R$. Also, we have $Dom(f) = Dom(q) = Dom(q')$ and then, $f \in \dot{\Sigma}(q')$.*

If $q = \emptyset = q'$, then, $\dot{\Sigma}(\emptyset) = \{\emptyset_{D,R}\} = \dot{\Sigma}(q')$. □

We have as a result that $\dot{\Sigma}$ is not an onto map because if we pick D and R with 2 elements $d_1 \neq d_2 \in D$ and $r \neq r' \in R$ we can take $f, f' : D \longrightarrow R$ constant functions such that, $f(d) = r, \quad f'(d) = r' \;\; \forall d \in D$.

If $f \in \dot{\Sigma}(q)$ then, $\forall d \in D \;\; (d, r) \in q$. And, if $f' \in \dot{\Sigma}(q)$ then, $\forall d \in D \;\; (d, r') \in q$. Now, we can define

$$g(d) = \begin{cases} r & \text{if } d = d_1 \\ r' & \text{if } d \neq d_1 \end{cases}$$

Then, $g \in \dot{\Sigma}(q)$ but $g \neq f$ and $g \neq f'$.

Notice that it is not the case that if $|D| < 2$ and $|R| < 2$ then $\dot{\Sigma}$ is onto. Let be: $|D| = \{d\}$ and $|R| = \{r\}$. Then we have that $\mathcal{P}(D \times R) = \{\emptyset, \{(d, r)\}\}$.

And $D \rightharpoonup R = \{f_0 = \emptyset_{D,R}, f_1 = \{d \mapsto r\}\}$[2].

And, there is not any relation q in $\mathcal{P}(D \times R)$ s.t. $\dot{\Sigma}(q) = \emptyset$[3] $\in \mathcal{P}(D \rightharpoonup R)$. □

Some others remarks about this functions are:

1. $\{\emptyset_{D,R}\} \in Img(\dot{\Sigma})$, since $q = \emptyset$ performs $\dot{\Sigma}(\emptyset) = \{\emptyset_{D,R}\}$.

[2]Here, f_0 is the partial function which has the element d in its data carrier, the element r in its result carrier but it does nothing.

[3]$\emptyset$ is the empty set.

2. All unitary sets $\{f\}$ of functions from D to R are in the $Img(\dot{\Sigma})$, since if $q = f$ then $\dot{\Sigma}(q) = \{f\}$.

3. $R^D \in Img(\dot{\Sigma})$, since $q = D \times R$ is such that $\dot{\Sigma}(q) = R^D$.

Lemma 3.4 $\forall q \ \ q = \bigcup \dot{\Sigma}(q)$.

Proof: Let be $(a,b) \in q \neq \emptyset$ *iff* $\exists f \in \dot{\Sigma}(q)$ *s.t.* $f(a) = b$ *iff* $(a,b) \in \bigcup \dot{\Sigma}(q)$. *If* $q = \emptyset = \bigcup \dot{\Sigma}(q)$.

□

$Img(\dot{\Sigma})$ comprises the sets of runs of all problems with domain of data D and domain of results R.

Maybe it would be more natural to define the set of runs of a problem $P =< D, R, q >$ as the set of total functions in D which are included in q, i.e., the set of total Skolem functions of its relation. This can be done through the function $\Sigma_{D,R} : \mathcal{P}(D \times R) \longrightarrow \mathcal{P}(R^D)$ s.t.
$q \longmapsto \Sigma_{D,R}(q) = \{f : D \longrightarrow R \ / \ f \subseteq q \ , \ \ f$ total function$\}$. But, we can observe that Σ and $\dot{\Sigma}$ perform the same result when q is a total relation and in any other case Σ is equal to $\emptyset$. Then, in a more general way, we can work for all practical purposes with $\dot{\Sigma}$, because all we can do with Σ we can also obtain with $\dot{\Sigma}$. We have presented the function $\dot{\Sigma}$ which reflects the fact that the runs of a "partial program" are the Skolem functions of its "partial condition". Or what is the same, if we think that in the result carrier R we have a distinguished element $\perp$ (this element contains no information, it serves to model the values of computations that never produce any information), runs should be the total functions defined by : if $d \in Dom(q)$ then, $f(d) = r$ s.t. $(d,r) \in q$, if $d \in D - Dom(q)$ then, $f(d) = \perp$. It is important to notice that if q is viable then, $\dot{\Sigma}$ only includes total functions and if q is non-viable then, $\dot{\Sigma}$ only includes partial functions.

In other words, given a program P its runs are all total functions or all partial functions in agreement with its condition.

3.2 The algebraic structure induced by $\dot{\Sigma}$

If we are interested in obtaining the algebraic structure induced by the function $\dot{\Sigma}$, we have to define an adequate relationship between the elements in $Img(\dot{\Sigma})$. For example, it must exist the $l.u.b.(F, F')$, when $F, F' \in Img(\dot{\Sigma})$ and F is a set of total functions and F' is a set of partial functions. If our relationship is the $\subseteq$ of sets then, F and F' are included (as sets) in the $l.u.b.(F, F')$. Thus, $l.u.b(F, F')$ is a set with some total functions and with some partial functions then, $l.u.b(F, F') \notin Img(\dot{\Sigma})$ because we have already observed that a set in $Img(\dot{\Sigma})$ only includes total functions or only includes partial functions. Thus, $Img(\dot{\Sigma})$ will not be a sublattice of $(\mathcal{P}(D \rightharpoonup R), \subseteq)$.

We define the following partial order relation :

Definition 3.5 $\forall F, F' \in Img(\dot{\Sigma}_{D,R}) we say that$

$F \ll F' \quad \textit{iff} \quad \forall\, f \in F \;\exists\, f' \in F' \;\; s.t. \;\; \forall\, d \in \; Dom(f) \;\; f(d) = f'(d).$

Note that we have not defined $F \ll F'$ iff $\bigcup F \subseteq \bigcup F'$ because this is not a partial order relation (the antisymmetry law doesn't hold). Moreover, let's observe that if we take $F, F' \in Img(\Sigma)$ then, the relation $\ll$ corresponds to the set inclusion, but $\ll$ allows us to hand together sets of partial and total functions without confusion.

Also, note that $\ll$ is not a partial order relation in $\mathcal{P}(D \rightharpoonup R)$ (the antisymmetry law doesn't hold, because any set of functions $\{f_1, \ldots, f_n\} \ll \{\emptyset_{D,R}, f_1, \ldots, f_n\}$ and $\{\emptyset_{D,R}, f_1, \ldots, f_n\} \ll \{f_1, \ldots, f_n\}$, but they are different sets).

Lemma 3.6 *For all F in $Img(\dot{\Sigma}_{D,R})$ we have that $\dot{\Sigma}(\bigcup F) = F$.*

Proof :

Obvious. If $F \in Img(\dot{\Sigma})$ we have that exists $q \subseteq D \times R$ s.t. $F = \dot{\Sigma}(q)$ and $q = \bigcup F$. □

Proposition 3.7 *$(Img(\dot{\Sigma}_{D,R})\ ,\ \ll)$ is a lattice.*

Proof:

- *Let's define for all $F, F' \in Img(\dot{\Sigma}_{D,R})$, $l.u.b.(F,F') = \dot{\Sigma}(\ (\bigcup F)\ \bigcup\ (\bigcup F'))$.*
  - *$l.u.b.(F,F')$ is in $Img(\dot{\Sigma})$.*
  - *We have that $F \ll F \cup F'$ then*

    $\bigcup F \subseteq \bigcup (F \cup F') = (\bigcup F) \bigcup (\bigcup F')$. *Then,*

    $F = \dot{\Sigma}(\bigcup F) \ll \dot{\Sigma}(\ (\bigcup F) \bigcup (\bigcup F')\ )\ )$. *Idem for F'.*
  - *Let's assume that there exists another u.b. Z then, $F \cup F' \ll Z$. Then,*

    $\bigcup (F \cup F') = (\bigcup F) \bigcup (\bigcup F') \subseteq (\bigcup Z)$. *Thus,*

    $\dot{\Sigma}\ ((\bigcup F) \bigcup (\bigcup F')) \ll \dot{\Sigma}(\bigcup Z) = Z$.
- *Let's define for all $F, F' \in Img(\dot{\Sigma}_{D,R})$, $g.l.b.(F,F') = \dot{\Sigma}(\ (\bigcup F) \bigcap (\bigcup F'))$*
  - *$g.l.b.(F,F')$ is in $Img(\dot{\Sigma})$.*
  - *We have that $(\bigcup F) \bigcap (\bigcup F') \subseteq (\bigcup F)$.Then,*

    $\dot{\Sigma}\ (\ (\bigcup F) \bigcap (\bigcup F')) \ll \dot{\Sigma}(\bigcup F) = F$.
  - *Let's assume that there exists another l.b. W then, $W \ll F$ and $W \ll F'$.*

    Thus, $\bigcup W \subseteq \bigcup F$ and $\bigcup W \subseteq \bigcup F'$ and then, $\bigcup W \subseteq \bigcup F \bigcap \bigcup F'$. Therefore,

    $\dot{\Sigma}\ (\bigcup W) = W \ll \dot{\Sigma}\ (\bigcup F \bigcap \bigcup F')$.

- *The element called unit is* R^D*, because for all* F *we have* $F \ll R^D$.

- *The element called zero is* $\{\emptyset_{D,R}\}$*, because for all* F *we have* $\{\emptyset_{D,R}\} \ll F$. □

Proposition 3.8 $\dot{\Sigma}_{D,R}$ *is a monotonic map with the order* $\ll$.

Proof:

Let be $\emptyset \neq q \subseteq q'$ *and* $Dom(q) = Dom(q')$*. Then for all* $f \in \dot{\Sigma}(q)$ *we have that* $f \in \dot{\Sigma}(q')$*. Thus, for all* $f \in \dot{\Sigma}(q)$ *there exists* $f' = f \in \dot{\Sigma}(q')$ *s.t.* $f(d) = f'(d)$*. Therefore,* $\dot{\Sigma}(q) \ll \dot{\Sigma}(q')$*. Let be* $\emptyset \neq q \subseteq q'$ *and* $Dom(q) \subset Dom(q')$*. Let's take* $f \in \dot{\Sigma}(q)$*, thus,* $\bigcup f \subseteq q \subseteq q'$*. Also, let's pick* $f' \in \dot{\Sigma}(q')$ *s.t.* $f'/_{Dom(q)} = f$ *(*f' *is an extension of* f*, this function exists because* $q \subseteq q'$*). Thus, for all* $f \in \dot{\Sigma}(q)$ *exists* $f' \in \dot{\Sigma}(q')$ *s.t.* $f(d) = f'(d)$ *for all* d *in* $Dom(q)$*. And then,* $\dot{\Sigma}(q) \ll \dot{\Sigma}(q')$.

If $\emptyset = q \subseteq q'$ *is obvious.* □

Proposition 3.9 *If* $\dot{\Sigma}(q) \ll \dot{\Sigma}(q')$ *then,* $q \subseteq q'$.

Proof: Let be $(a, b) \in q \neq \emptyset$ *then, there exists* $f \in \dot{\Sigma}(q)$ *s.t.* $f(a) = b$*. Then, there is* $f' \in \dot{\Sigma}(q')$ *s.t.* $f'(a) = f(a) = b$*. And, therefore,* $(a, b) \in q'$. □

The next result gives the meaning of the *l.u.b.* and the *g.l.b.* If F is the set of runs of a problem $P =< D, R, q >$, and F' is the set of runs of the problem $P' =< D, R, q' >$, the *l.u.b.*(F, F') is the set of runs of the problem $P + P' =< D, R, q \cup q' >$, and the *g.l.b.*$(F, F')$ is the set of runs of the problem $P \diamond P' =< D, R, q \cap q' >$. Formally, we have:

Proposition 3.10 *If* $F = \dot{\Sigma}(q)$, $F' = \dot{\Sigma}(q')$ *then,*

l.u.b.$(F, F') = \dot{\Sigma}(q \cup q')$ *and g.l.b.*$(F, F') = \dot{\Sigma}(q \cap q')$.

Proof:

1. $l.u.b.(F,F') = \dot{\Sigma}((\cup F)\bigcup(\cup F')) = \dot{\Sigma}(\cup\dot{\Sigma}(q) \bigcup \cup\dot{\Sigma}(q')) = \dot{\Sigma}(q \cup q') = \dot{\Sigma}(l.u.b.(q,q'))$

2. $g.l.b.(F,F') = \dot{\Sigma}((\cup F)\bigcap(\cup F')) = \dot{\Sigma}(\cup\dot{\Sigma}(q) \bigcap \cup\dot{\Sigma}(q')) = \dot{\Sigma}(q \cap q') = \dot{\Sigma}(g.l.b.(q,q'))$ □

If we take $F \in Img(\dot{\Sigma})$, this set of functions univocally determines the problem $P =< D, R, q >$ s.t. $\dot{\Sigma}(q) = F$. If we go down through the lattice $(Img(\dot{\Sigma}), \ll)$, starting from a fixed F, and taking any chain in address to the first element, we have a descendant succession of sets of functions which are sets of functional complete subproblems of the proper subproblems of P. This succession always includes the unitary sets (the atoms of the lattice), which are the problems with functional relation, and it finishes at the set $\{\emptyset_{D,R}\}$. The different options to choose a chain accord with the following idea : "given a problem there is more than a way to refine it". Thus, we have the next result that means that for a non-deterministic program we have a more non-deterministic program than it, which gives the same runs if restricted to the domain of the first program. (i.e., the sets of functional complete subproblems of all the proper subproblems of a given problem P, are relationated by $\ll$ with the set of functional complete subproblems of P).

Proposition 3.11 *Given q, we define $\mathcal{R}_q = \{q' \mid q' \subseteq q\}$ and $\mathcal{F}_{\dot{\Sigma}(q)} = \{F' \in Img(\dot{\Sigma}) \mid F' \ll \dot{\Sigma}(q)\}$. Then, $\dot{\Sigma}(\mathcal{R}_q) = \mathcal{F}_{\dot{\Sigma}(q)}$.*

Proof :

- *If we take $q' \in \mathcal{R}_q$ we have that $\dot{\Sigma}(q) \in \dot{\Sigma}(\mathcal{R}_q)$. Because $q' \subseteq q$ we can argue, $\dot{\Sigma}(q') \ll \dot{\Sigma}(q)$ and then $\dot{\Sigma}(q') \in \mathcal{F}_{\dot{\Sigma}(q)}$.*

- *Let be $F' \in \mathcal{F}_{\dot{\Sigma}(q)}$, that is $F' \ll \dot{\Sigma}(q)$ and $F' \in Img(\dot{\Sigma})$. Then, there is a q' that $\dot{\Sigma}(q') = F'$. Thus, we have $\dot{\Sigma}(q') \ll \dot{\Sigma}(q)$, and then, $q' \subseteq q$. Therefore, $\dot{\Sigma}(q') \in \dot{\Sigma}(\mathcal{R}_q)$.*

3.3 The algebraic structure induced by Σ

Given $F \subseteq \mathcal{P}(R^D)$ we can define its closure by means of the following three sets :

1. The functional closure is the set:

   $\widehat{F} = \{g : D \longrightarrow R \ / \ \forall\, d \in D\ \exists\, f \in F\ \ g(d) = f(d)\ \} \ = \ \{g : D \longrightarrow R \ / \ g \subseteq \cup F\}$

2. The under "gluing" closure

   Let P be a partition of D, we call an assignment $a : P \longrightarrow F$ of gluing data. The result of gluing is a function $\underline{a} : D \longrightarrow R$ s. t. $d \longmapsto (\ a([d])\ )\ (d)$. And we define the closure under gluing of F to be the set:

   $\underline{F} = \{\underline{a} \ / \ a : P \longrightarrow F\ ,\ \ P \textit{ partition on } D\}$

3. The product closure (or "Bertrand Russell version ")

   Let $F\!\uparrow (d) = \{f(d) \in R \ / \ f \in F\}$. Then, $F\!\uparrow : D \longrightarrow \mathcal{P}(R)$ can be viewed as the family of D-sorted set $F\!\uparrow = (F\!\uparrow (d))_{d \in D}$. Thus, we define the closure of F:

   $\Pi F\!\uparrow = \{h : D \longrightarrow R \ / \ h(d) \in F\!\uparrow (d)\}$

Proposition 3.12 $\widehat{F} = \underline{F} = \Pi F\!\uparrow$

Proof :

- $\underline{F} \subseteq \widehat{F}$

  Let be $\underline{a} \in \underline{F}$ then, there exists the assignment $a : P \longrightarrow F$ for some partition P of D. Because $a([d])$ belongs to F we can define $f = a([d]) \in F$. And, $\underline{a}(d) = a([d])(d) = f(d)\ \ \forall d \in D$.

- $\Pi F\!\uparrow \subseteq \underline{F}$

  Let be $h \in \Pi F\!\uparrow$. Consider the partition P of D with singleton blocks. Notice that for *each* $\{d\} \in D$*, since* $h(d) \in F\!\uparrow(d)$*, there exists* f *in* F *such that* $h(d) = f(d)$*.*

So, the Axiom of Choice gives us a function $a : P \longrightarrow F$ *such that for every* d *in* D*,* $a([d])(d) = h(d)$*. Hence,* $h = \underline{a} \in \underline{F}$*.*

- $\widehat{F} \subseteq \Pi F\!\uparrow$

  Let be $g : D \longrightarrow R$ *s.t.* $\forall d \in D \;\; \exists f \in F \;\; g(d) = f(d)$*. Because* $f(d) \in F\!\uparrow(d)$ *holds for each* f *in* F*, we have that* $g(d) = f(d) \in F\!\uparrow(d)$*. And so,* $g \in \Pi F\!\uparrow$ *.* □

Observations :

1. In the under "gluing" case $\underline{a}$ behaves as $a([d])$ on the block $[d]$ of P.

2. In the under "gluing" case we consider all the partitions P of D, but the best partition is the partition whose elements are the unitary sets of elements of D. With this only partition (and all the assignments a for it) we can obtain all functions which are included in F.

3. In the product closure, $h : D \longrightarrow R$ are the Skolem functions of the D-sorted set $F\!\uparrow$.

Proposition 3.13 $\;\widehat{\ }: \mathcal{P}(R^D) \longrightarrow \mathcal{P}(R^D)$ *is a closure operator over* R^D*.*

Proof :

1. $F \subseteq \widehat{F}$

Let be $f : D \longrightarrow R$ *s.t.* $f \in F$ *then,* $f \subseteq \bigcup F$ *and so,* $f \in \widehat{F}$*.*

2. $F \subseteq F' \implies \widehat{F} \subseteq \widehat{F'}$

Let be $F \subseteq F'$ *then,* $\bigcup F \subseteq \bigcup F'$*.And so, we have that* $\forall g,\; g : D \longrightarrow R$ *if* $g \subseteq \bigcup F$

then, $g \subseteq \bigcup F'$. *And then,*

$$\widehat{F} = \{g : D \longrightarrow R \;/\; g \subseteq \bigcup F\} \subseteq \{g : D \longrightarrow R \;/\; g \subseteq \bigcup F'\} = \widehat{F'}.$$

3. $\widehat{\widehat{F}} = \widehat{F}$

   First let's observe that : $\bigcup \widehat{F} = \bigcup F$

   - $\bigcup F \subseteq \bigcup \widehat{F}$, *direct of 1.*
   - $\bigcup \widehat{F} \subseteq \bigcup F$.

     Let be $(a, g(a)) \in \bigcup \widehat{F}$ *then, by definition of* $\widehat{F}$, *we have that if* $a \in D$ *then,* $g(a) \in R$ *and there exists* $f \in F$ *s.t.* $g(a) = f(a)$ *and then* $(a, g(a)) \in \bigcup F$.

     And so, we have $\widehat{F} = \{g : D \longrightarrow R \;/\; g \subseteq \bigcup F\} = \{g : D \longrightarrow R \;/\; g \subseteq \widehat{F}\} = \widehat{\widehat{F}}$. □

Also we have the following results :

- $\widehat{\emptyset} = \emptyset$.
- $\widehat{\{f\}} = \{f\}$.
- $\widehat{F} \bigcup \widehat{F'} \neq \widehat{F \bigcup F'}$. We see this inequality with an example :

  (1) Let be $D = \{1, 2\}$ and $R = \{A, B\}$ then $R^D = \{\; f_1 = \{(1, A), (2, B)\}$, $f_2 = \{(1, A), (2, A)\}$, $f_3 = \{(1, B), (2, B)\}$, $f_4 = \{(1, B), (2, A)\}\; \}$

  Let be $F = \{f_2\}$ and $F' = \{f_3\}$, then $\widehat{F} \bigcup \widehat{F'} = \{f_2, f_3\}$ but $\widehat{F \bigcup F'} = \{f_1, f_2, f_3, f_4\}$.

Proposition 3.14 *If* D *is infinite and* $|R| \geq 2$ *then,* $\widehat{\ }$ *is not an algebraic operator.*

Proof:

We have to prove that it is not the case that if $f \in \widehat{F}$, *then* $f \in \widehat{F'}$ *for some finite* $F' \subseteq F$.

Let us see an example :

Let be $F = \{f_0, f_1, f_2, \ldots, f_n, \ldots\}$, *where*

$f_0 : N \longrightarrow N$ *s.t.* $\quad f_0(x) = x \;\; \forall x \in N$

$f_1 : N \longrightarrow N$ *s.t.*

$$f_1(x) = \begin{cases} 0 & \text{if } x = 1 \\ x & \text{otherwise} \end{cases}$$

$f_2 : N \longrightarrow N$ *s.t.*

$$f_2(x) = \begin{cases} 0 & \text{if } x = 2 \\ x & \text{otherwise} \end{cases}$$

$f_n : N \longrightarrow N$ *s.t.*

$$f_n(x) = \begin{cases} 0 & \text{if } x = n \\ x & \text{otherwise} \end{cases}$$

We can pick $f : N \longrightarrow N$ *s.t.* $\forall x \in N \;\; f(x) = 0$ *belongs to* $\widehat{F}$. *But it does not exist any finit subset of* F, F', *such that* $\{(0,0), (1,0), (2,0), \ldots, (n,0), \ldots\}$ *that belongs to* $\bigcup F' = \bigcup \widehat{F'}$ *(which means* $f \in \widehat{F'}$*).* □

Proposition 3.15 $Img(\Sigma)$ *is the set of fixed points of* $\widehat{\ }$*, that is* $Img(\Sigma) = \{F \;/\; F = \widehat{F}\}$.

Proof :

- *Let be* $F \neq \emptyset \in Img(\Sigma)$, *that is, there is a viable* $q \in \mathcal{P}(D \times R)$ *s.t.* $\Sigma(q) = F$. *Then,* $\bigcup F = \bigcup(\Sigma(q)) = q$.

  Then, $\widehat{F} = \{g : D \longrightarrow R \;/\; g \subseteq \bigcup F\} = \{g : D \longrightarrow R \;/\; g \subseteq q\} = F$.

- *Let be* $\emptyset \neq \widehat{F} = F = \{g : D \longrightarrow R \;/\; g \subseteq \bigcup F\}$. *Let us define* $q = \bigcup F$. *Then,* $\Sigma(q) = \Sigma(\bigcup F) = \widehat{F} = F$.

  Let's see that $\Sigma(\bigcup F) = \widehat{F}$. *Let be* $f \in \Sigma(\bigcup F)$ *iff (because* $\bigcup F \in \mathcal{P}(D \times R)$*)* $f \subseteq (\bigcup F)$ *iff* $f \in \widehat{F}$.

- *The case with* $\emptyset$ *is obvious.* □

Proposition 3.16 *$Img(\Sigma)$ is a complete lattice with respect to the set inclusion.*

Proof : see [Gr78, pag.184]. □

Lemma 3.17 *Let $F, F' \in Img(\Sigma)$. The l.u.b.$(F, F') = \widehat{F \bigcup F'}$, and g.l.b.$(F, F') = F \bigcap F'$.*

Proof :

- – $\widehat{F \bigcup F'} \in Img(\Sigma)$, *because* $\widehat{F \bigcup F'} = \widehat{\widehat{F \bigcup F'}}$.
  - – $\widehat{F \bigcup F'}$ *is an upper bound of* F *and* F', *because* $F \subseteq F \bigcup F' \subseteq \widehat{F \bigcup F'}$.
  - – $\widehat{F \bigcup F'}$ *is the least upper bound of* F *and* F'.

    Let Z *be another upper bound of* F *and* F'. *And so ,* $F \bigcup F' \subseteq Z$, *then,*

    $\widehat{F \bigcup F'} \subseteq \hat{Z} = Z$.

- – *Let be* $F, F' \in Img(\Sigma)$. *We have that* $F = \widehat{F}$, *and* $F' = \widehat{F'}$, *and then,*

    $F \bigcap F' = \widehat{F} \bigcap \widehat{F'}$.

    Let be $g \in \widehat{F \bigcap F'}$ *but this is the same that to say* $\forall d \in D\ \exists f \in F \bigcap F'$ *s.t.* $g(d) = f(d)$ *then,* $\forall d \in D$

    $\exists f \in F$ *s.t.* $g(d) = f(d)$ *and* $\forall d \in D\ \exists f \in F'$ *s.t.* $g(d) = f(d)$ *iff* $g \in \widehat{F}$ *and* $g \in \widehat{F'}$ *iff* $g \in \widehat{F} \bigcap \widehat{F'}$ *iff* $g \in F \bigcap F'$. *Thus, we have* $F \bigcap F' = \widehat{F \bigcap F'}$, *and so* $F \bigcap F' \in Img(\Sigma)$.

  - – $F \bigcap F'$ *is a lower bound of* F *and* F'.
  - – $F \bigcap F'$ *is the greatest lower bound of* F *and* F'. □

Some properties that hold in this lattice are:

1. $Img(\Sigma)$ is not a distributive lattice.

   Let be, for example, $F, F', F'' \in Img(\Sigma)$, s.t. $F = \{f_2\}$, $F' = \{f_3\}$, $F'' = \{f_1\}$ are as in the example (1).

Then, $F''\bigcap(F\widehat{\bigcup}F') = \{f_1\}$. But, $(F''\bigcap F)\widehat{\bigcup}(F''\bigcap F') = \emptyset$.

2. $Img(\Sigma)$ is not a pseudocomplemented lattice.

   Let D and R be s.t. we can define three or more different functions $f : D \longrightarrow R$ (it is sufficient that $|D| \geq 2$ and $|R| \geq 2$, or $|R| \geq 3$). Let be $F, F', F'' \in Img(\Sigma)$ s.t. F, F', F'' are atoms of $Img(\Sigma)$. Then, $g.l.b(F, F') = \emptyset = g.l.b.(F, F'')$ and $F' \not\subseteq F''$ and $F'' \not\subseteq F'$.

3. $Img(\Sigma)$ is not a copseudocomplemented lattice.

   Let $F = \{f_1, f_2\}$, $F' = \{f_1, f_3\}$, $F'' = \{f_2, f_4\}$ be as in example (1). Then, $l.u.b(F, F') = R^D = l.u.b(F, F'')$ and $F' \not\subseteq F''$ and $F'' \not\subseteq F'$.

4. $Img(\Sigma)$ is a bounded lattice because $\emptyset$ and R^D are in $Img(\Sigma)$.

5. $Img(\Sigma)$ is a complemented lattice (note that $Img(\Sigma)$ is not a distributive lattice) .

   The complement of $F \in Img(\Sigma)$ is $\Sigma(\overline{\bigcup F})$. Let's prove this result:

   - $F \bigcap \Sigma(\overline{\bigcup F}) = \emptyset$. Suppose that $F \bigcap \Sigma(\overline{\bigcup F}) \neq \emptyset$, then there is some f in $F = \widehat{F}$ and f is in $\Sigma(\overline{\bigcup F})$. Then, $f \subseteq (\overline{\bigcup F})$, then $f \not\subseteq (\bigcup F)$, and so $f \notin F = \widehat{F}$, but this is a contradiction.

   - $F \mathbin{\widehat{\bigcup}} \Sigma(\overline{\bigcup F}) = R^D$ .The only thing to see is that $F \mathbin{\widehat{\bigcup}} \Sigma(\overline{\bigcup F}) \supseteq R^D$. If $f \in R^D$ and $f \in F$ then, $f \in F \mathbin{\widehat{\bigcup}} \Sigma(\overline{\bigcup F})$. If $f \notin F = \widehat{F}$ then, $f \not\subseteq \bigcup F$ then, $f \subseteq \overline{\bigcup F}$ then, $f \in \Sigma(\overline{\bigcup F})$ and so, $f \in F \mathbin{\widehat{\bigcup}} \Sigma(\overline{\bigcup F})$.

6. $Img(\Sigma)$ is not a modular lattice: a lattice L is called *modular* sii

$\forall x \geq z \Longrightarrow (x \wedge y) \vee z = x \wedge (y \vee z)$, [Gr78].

Let be, as in example (1) $x = \{f_1, f_2\}$, $z = \{f_1\}$, $y = \{f_3, f_4\}$. Then,

$(x \widehat{\bigcap y}) \bigcup z = \{f_1\} \neq x \bigcap (y \widehat{\bigcup z}) = \{f_1, f_2\}$.

7. $Img(\Sigma)$ is not a lattice of finite length.

If $D = N$ and $R = N$, given a $n \in N$ we always can get a set F of n functions $f_i : N \longrightarrow N$, $1 \leq i \leq n$, and $F \in Img(\Sigma)$ (for example if the n functions f_i are all different each other only in one same point, and equal in the rest).

8. $Img(\Sigma)$ has atoms, because the unitary sets are in the lattice and they cover to $\emptyset$. Thus, $Img(\Sigma)$ is an atomic lattice.

9. $Img(\Sigma)$ is an atomistic lattice: a lattice L is an *atomistic lattice* iff every element of L is a join of atoms, [Gr78].

Let be $F \in Img(\Sigma)$. We have for all $f \in F$ that $\{f\} \in Img(\Sigma)$ and it is an atom of $Img(\Sigma)$, thus $\widehat{\bigcup}_{f \in F}\{f\} = F$.

10. $Img(\Sigma)$ is not a continuous lattice: a lattice L is a *continuous lattice* iff for all

$a \in L$ $a \wedge \bigvee D = \bigvee\{a \wedge x \ / \ x \in D\}$ for any directed subset D of L, [Gr78].

Let be $D = \{G_i\}_{i \in N} - N^N$, where the G_i are as in Proposition 3.18 below.

Thus, $\widehat{\bigcup} D = N^N$.

And, if we take $H = \{id : N \longrightarrow N \ / \ id(x) = x\}$ we have that, $H \bigcap \widehat{\bigcup} D = H$. On the other hand we have that, because the $Img(id)$ is an infinity set, $\forall \ G_i \in D$, $id \notin G_i$, and so $\{id\} \bigcap G_i = \emptyset$, $\forall G_i \in D$. Therefore,

$\widehat{\bigcup}\{\{id\}\bigcap G_i \ , \ \forall G_i \in D\} = \emptyset.$

Proposition 3.18 *Given $F \subseteq R^D$ there is not the greatest $G \subseteq F$ s.t. $G \in Img(\Sigma)$.*

Proof :

Let be $F = \{f : N \longrightarrow N \ / \ Img(f) \ \text{ is a finite set } \} \subset \widehat{F} = N^N$

$G_1 = \{f : N \longrightarrow N \ / \ Img(f) \subseteq \{0,1\} \ \} = \widehat{G_1}$

$G_2 = \{f : N \longrightarrow N \ / \ Img(f) \subseteq \{0,1,2\} \ \} = \widehat{G_2}$

$G_3 = \{f : N \longrightarrow N \ / \ Img(f) \subseteq \{0,1,2,3\} \ \} = \widehat{G_3}$

$G_n = \{f : N \longrightarrow N \ / \ Img(f) \subseteq \{0,1,\ldots,n\} \ \} = \widehat{G_n}$

Then, $G_1 \subset G_2 \subset G_3 \subset \ldots \subset G_n \subset \ldots \subset F \subset \widehat{F}$

Proposition 3.19 *Given $F \subseteq R^D$ there is the smallest $H \in Img(\Sigma)$ s.t. $F \subseteq H$.*

Proof :

Let's define $M_F = \{H \ / \ F \subseteq H \ \text{ and } \ H \in Img(\Sigma)\}$. This set is not empty because $R^D \in M_F$. $(M_F, \subseteq)$ is a partially ordered set , we shall show that it has first element, that is there exists an element $H \in M_F$ s.t. for all other $H' \in M_F$ $H \subseteq H'$.

We shall prove that $\widehat{F}$ is the first element.

- *$\widehat{F} \in Img(\Sigma)$ and $F \subseteq \widehat{F}$ thus, $\widehat{F} \in M_F$.*
- *Let be $H' \in M_F$ that is, $F \subseteq H'$ then, $\widehat{F} \subseteq \widehat{H'} = H'$.* □

Lemma 3.20 *$(M_F, \subseteq)$ is a bounded sublattice of $(Img(\Sigma), \subseteq)$.*

Proof :

Let be $H_1, H_2 \in M_F$ that is, $F \subseteq H_1$ and $F \subseteq H_2$ then $F \subseteq H_1 \bigcap H_2$ and we have already seen that $H_1 \bigcap H_2 \ \in Img(\Sigma)$ when H_1, H_2 are in $Img(\Sigma)$. Then, $H_1 \bigcap H_2 \ \in M_F$ and

$g.l.b(H_1, H_2) = H_1 \bigcap H_2$.

Also, we have $F \subseteq H_1 \widehat{\bigcup} H_2$ *and this element is in* $Img(\Sigma)$. *Then,* $H_1 \widehat{\bigcup} H_2 \in M_F$ *and* $l.u.b(H_1, H_2) = H_1 \widehat{\bigcup} H_2$.

$\widehat{F}$ *is the first element and* R^D *is the last element of* M_F. □

Lemma 3.21 $Img(\Sigma) = \bigcup_{F \in \mathcal{P}(R^D)} M_F$.

Proof :

- *Let be* $F \in Img(\Sigma)$ *then,* $F \in M_F$ *and hence* $F \in \bigcup_{F \in \mathcal{P}(R^D)} M_F$.

- *Let be* $H \in \bigcup_{F \in \mathcal{P}(R^D)} M_F$ *then, there is* $F \in \mathcal{P}(R^D)$ *s.t.* $H \in M_F$*, and then,* $H \in Img(\Sigma)$. □

Finally, as in the case with the function $\dot{\Sigma}$ we have the following result:

Proposition 3.22 *If* q *and* q' *are both viable or empty and* $F = \Sigma(q)$, $F' = \Sigma(q')$ *then we have that* $l.u.b.(F, F') = \Sigma(q \cup q')$ *and* $g.l.b.(F, F') = \Sigma(q \cap q')$.

Proof:

1. $l.u.b.(F, F') = F \widehat{\cup} F' =$

   $\Sigma(\bigcup(F \cup F')) = \Sigma(\bigcup_{f \in F \cup F'} f) =$

   $\Sigma(\bigcup_{f \in F} f \ \cup \ \bigcup_{f \in F'} f) = \Sigma(\bigcup F \ \cup \ \bigcup F') =$

   $\Sigma(q \cup q') = \Sigma(l.u.b.(q, q'))$

2. $g.l.b.(F, F') = F \bigcap F' = \Sigma(\bigcup_{f \in F} f) \ \bigcap \ \Sigma(\bigcup_{f \in F'} f) =$

   $\Sigma(q) \ \bigcap \Sigma(q') = \Sigma(q \cap q')$.

   Let's see that the last equality is true.

Let's assume that $\Sigma(q) \bigcap \Sigma(q') \neq \emptyset$ *then,*

$f \in \Sigma(q) \bigcap \Sigma(q')$ *iff* $f \in \Sigma(q)$ *and* $f \in \Sigma(q')$ *iff* $f \subseteq q$ *and* $f \subseteq q'$ *iff*

$f \subseteq q \cap q'$ *iff* $f \in \Sigma(q \cap q')$.

Now, let's assume that $\Sigma(q) \bigcap \Sigma(q') = \emptyset$ *iff*

$\neg\, ((\exists f)\, (f \in R^D \;\wedge\; f \in \Sigma(q) \;\wedge\; f \in \Sigma(q')))$ *iff*

$\neg\, ((\exists f)\, (f \in R^D \;\wedge\; f \subseteq q \;\wedge\; f \subseteq q'))$ *iff*

$\neg\, ((\exists f)\, (f \in R^D \;\wedge\; f \subseteq q \cap q'))$ *iff* $\Sigma(q \cap q') = \emptyset$ □

4 Relating Lattices

In this section we study the possibility of relating the algebraic structures involved in the previous sections.

First of all, let's notice that $(Img(\Sigma), \subseteq)$ is a sublattice of $(Img(\dot{\Sigma}), \ll)$ (all the elements of $Img(\Sigma)$ are in $Img(\dot{\Sigma})$ because Σ is the restriction of $\dot{\Sigma}$ to viable or empty relations. And, if F, F' are in $Img(\Sigma)$ then the $l.u.b.(F, F')$ and the $g.l.b.(F, F')$ are equal in both lattices).

But, actually, we are interested in knowing what the relationship between the algebraic structure of problems (i.e., the semantics of angelic non-deterministic programs) and the algebraic structure of their functional complete subproblems (i.e., the sets of runs of the programs) is.

Let's define $PS_{D,R} = \{< D, R, q > \;/\; q \subseteq D \times R\}$ the set of all the proper subproblems of $< D, R, D \times R >$. We have that $(PS_{D,R}, \sqsubseteq)$ is isomorphic to $(\mathcal{P}(D \times R), \subseteq)$ and then, we have a distributive, complete, complement lattice with unit element $< D, R, D \times R >$ and zero element $< D, R, \emptyset >$.

The *l.u.b.* of two elements $< D, R, q >, < D, R, q' >$ in $PS_{D,R}$ is $< D, R, q \cup q' >$ and the *g.l.b.* is $< D, R, q \cap q' >$. Thus, $(PS_{D,R} , \sqsubseteq)$ is a sublattice of $(P_U, \sqsubseteq)$. Then, we define the function $\Gamma_{D,R} : (PS_{D,R}, \sqsubseteq) \longrightarrow (Img(\dot{\Sigma}_{D,R}), \ll)$ s.t.

$< D, R, q > \longmapsto \dot{\Sigma}_{D,R}(q) = \{f \in R^D \ / \ f \subseteq q\}$.

Proposition 4.1 *$\Gamma_{D,R}$ is an isomorphism of lattices.*

Proof:

- *$\Gamma_{D,R}$ is 1-1, because $\dot{\Sigma}_{D,R}$ is.*
- *$\Gamma_{D,R}$ is onto. Obvious.*
- $\Gamma_{D,R}(< D, R, \emptyset >) = \dot{\Sigma}_{D,R}(\emptyset) = \{\emptyset_{D,R}\}$.
- $\Gamma_{D,R}(< D, R, D \times R >) = \dot{\Sigma}_{D,R}(D \times R) = R^D$.
- *$\Gamma_{D,R}$ preserves l.u.b. Direct because* $\dot{\Sigma}_{D,R}(q \cup q') = \dot{\Sigma}_{D,R}(\ (\bigcup \dot{\Sigma}_{D,R}(q)) \ \bigcup \ (\bigcup \dot{\Sigma}_{D,R}(q')) \ )$.
- *$\Gamma_{D,R}$ preserves g.l.b. Direct because* $\dot{\Sigma}_{D,R}(q \cap q') = \dot{\Sigma}_{D,R}(\ (\bigcup \dot{\Sigma}_{D,R}(q)) \ \bigcap \ (\bigcup \dot{\Sigma}_{D,R}(q')) \ )$.

  □

Therefore, the lattice $(Img(\dot{\Sigma}_{D,R}) \ , \ \ll)$ has the same algebraic structure and properties than the lattices $(PS_{D,R} \ , \ \sqsubseteq) \cong (\mathcal{P}(D \times R), \subseteq)$.

Now, we want to obtain a relation between the algebraic structure of problems and their sets of total Skolem functions.

Let $V_{D,R} = \{< D, R, q > \ / \ q \subseteq D \times R \ , \ \text{q viable}\}$ be the set of all the *restrictions* of $< D, R, D \times R >$ for a fixed pair D, R.

$(V_{D,R}, \sqsubseteq)$ is an upper semilattice with unit element (the problem $< D, R, D \times R >$). The *l.u.b.* of two problems P and P' in $V_{D,R}$ is the addition problem $P + P' =< D, R, q \cup q' >$. Thus, $(V_{D,R}, \sqsubseteq)$ is an upper subsemilattice of $(P_U, \sqsubseteq)$.

Let's define the function $\Omega_{D,R} : (V_{D,R}\ ,\ \sqsubseteq) \longrightarrow (Img(\Sigma_{D,R})\ ,\ \subseteq)$ s.t. $< D, R, q > \longmapsto \Sigma_{D,R}(q) = \{f \in Rg(q)^D \ / \ f \subseteq q\ ,\ f \text{ total } function\}$.

Proposition 4.2 *$\Omega_{D,R}$ is a one-to-one, meet-homomorphism (or meet-embedding) of upper semilattices. Also, $\Omega_{D,R}$ takes unit into unit.*

Proof:

- *$\Omega_{D,R}$ is not an onto map. Because for all q the problems $< D, R, q >$ are viable ones then, $\Sigma(q) \neq \emptyset$.*

- *$\Omega_{D,R}$ is 1-1, because $\Sigma_{D,R}$ is 1-1.*

- $\Omega_{D,R}(< D, R, D \times R >) = \Sigma_{D,R}(D \times R) = R^D$. *Thus, $\Omega_{D,R}$ preserves the unit of the lattice.*

- $\Omega_{D,R}(l.u.b.(< D, R, q >, < D, R, q' >)) =$

  $\Omega_{D,R}(< D, R, q > \cup < D, R, q' >) =$

  $\Omega_{D,R}(< D, R, q \cup q' >) =$

  $\Sigma_{D,R}(q \cup q') =$

  $\Sigma_{D,R}(q) \widehat{\cup} \Sigma_{D,R}(q') =$

  $l.u.b.(\Sigma_{D,R}(q), \Sigma_{D,R}(q')) =$

  $l.u.b.(\Omega_{D,R}(< D, R, q >), \Omega_{D,R}(< D, R, q' >))$. *Thus, $\Omega_{D,R}$ preserves the l.u.b. of the lattice.*

$\square$

Then, $(Img(\Sigma_{D,R})\ ,\ \subseteq)$ is a meet-homomorphic image of $(V_{D,R}\ ,\ \sqsubseteq)$.

Note that we have not chosen the lattice of proper subproblems of $(D, R, D \times R)$ (for a given pair D, R), because with this definition of $\Sigma_{D,R}$ we can only obtain an onto map from this lattice to $(Img(\Sigma_{D,R})\ ,\ \subseteq)$. $\Sigma_{D,R}$ only preserves *l.u.b.* and *g.l.b.* when we deal with viable relations, therefore we couldn't obtain an homomorphism of lattices.

5 Conclusion

In the preceding sections we have studied and analyzed the algebraic structure determined by the sets of the functional complete subproblems of relations $q \subseteq D \times R$. This kind of subproblem is an extension of the well-known Skolem functions in order to cope with partial relations. We have proved that there is some sets of runs from a set D to a set R that do not correspond to any possible condition in $D \times R$. Also, we have verified some interesting properties of these sets of functions and finally, we related them to the lattice of problems.

References

[Gr78] G. Grätzer. *General Lattice Theory.* Academic Press, New York, 1978.

[HV89] A.M. Haeberer and P.A.S. Veloso. The inevitability of program testing - a theoretical analysis. In *XVII Chilean Conference on Informatics*, 1989.

[HV90a] A.M. Haeberer and P.A.S. Veloso. Partial relations for program derivation : Adequacy, inevitability and expressiveness. Series Monografias em Ciência da Computação 12, Pontificia Universidade Católica, 1990.

[HV90b] A.M. Haeberer and P.A.S. Veloso. Towards a formal analysis of the software development process. In *41st Meeting of the IFIP Working Group 2.1. Programming Languages and Calculi. Chester, England*, 1990.

[HV90c] A.M. Haeberer and P.A.S. Veloso. Towards a relational calculus for software construction. In *41st Meeting of the IFIP Working Group 2.1. Programming Languages and Calculi. Chester, England*, 1990.

[HVB89] A.M. Haeberer, P.A.S. Veloso, and G. Baum. *Formalización del Proceso de Desarrollo de Software.* EBAI. Kapelusz S.A., Buenos Aires, 1989.

[Leh84] M.M. Lehman. A further model of coherent programming process. In *IEEE Proceedings of Software Process Workshop*, UK, Feb 1984.

[Pol57] G. Polya. *How to Solve it: a new Aspect of the Mathematical Method.* Princeton University Press, Princeton, 1957.

[Sup60] P. Suppes. *Axiomatic Set Theory.* Van Nostrand, Princeton, 1960.

[Vel84] P.A.S. Veloso. Aspectos de uma teoria geral de problemas. *Cadernos de Historia e Filosofia da Ciência*, (7):21–42, 1984.

[VH] P.A.S. Veloso and A.M. Haeberer. A new algebra of first-order logic; binary relations resumed. Submitted for Publication to the Bulletin of American Mathematical Society.