

On Using Conditional Definitions in Formal Theories

Jean-Raymond Abrial¹ and Louis Mussat²

¹ Consultant, Marseille France jr@abrial.org

² DCSSI, Paris France louis.mussat@sgdn.pm.gouv.fr

1 Introduction

In this paper, our intention is to explore the notion of definition in formal theories and, in particular, that of *conditional definitions*. We are also interested in analyzing the consequences of the latter on the structure of corresponding *proof systems*. Finally, we shall investigate the various ways such proof systems can be *simplified*.

In formal texts, conditional definitions lead to such oddities as division by zero, the minimum of an empty set, or, more generally, the application of a function to an argument lying outside its domain. In the presence of such *ill-defined expressions*, people usually divide in two groups. A first group considers that such pathologies are totally uninteresting, that professional mathematicians never write things of that kind, and that it would correspond, in every day life, to people not mastering their mother tongue (so that the problem really is that of having people first learn how to correctly express themselves). A second group, especially among the formal developers, considers that this is a serious question, that it may lead to crashes, and that it is thus a problem that one must face and somehow “solve” (we shall see below that a quite a number of “solutions” have been proposed). In this paper, we again study the problem and try to give our view and contribution to it. Our “solution” is certainly not entirely novel but it opens, we think, a number of interesting ways to be explored in the domain of *mechanized proof strategy*. As our work is done in relation with B, we shall make our investigation on a specific formal theory, namely set theory as it has been re-constructed in the B-Book [1].

Before developing our main subject, it is certainly worthwhile recalling what we understand by a proper concept of *definition* in a formal theory. For this, we shall follow the way this notion was presented in an introductory book of logic written by P. Suppes [9] in the fifties, where an entire chapter is devoted to that question. The present paper is to be read *with that very understanding of the concept of definition in mind*.

1.1 On Definitions

Suppes first introduces two criteria characterizing the definition of new symbols in a formal theory: (1) the *Criterion of Eliminability*, and (2) the *Criterion*

of *Non-creativity*. The former requires that “any definition introducing a new symbol may be used to eliminate all subsequent meaningful occurrences (of that new symbol)”. The latter requires that any definition of a new symbol “(does not) make possible the derivation of some previously unprovable theorem stated wholly in terms of primitive and previously defined symbols”. In other words, a proper definition must not add any extra “power” to a theory, it is just a useful, but not indispensable, *extension* of it.

Then Suppes proposes some precise rules by which new symbols can be introduced while guaranteeing both previous criteria. He makes the distinction between rules defining either new *predicate* symbols or new *term* symbols. In both cases, this is done by means of equivalences. We shall continue to follow Suppes in the presentation of these rules.

1.2 Defining a New Predicate Symbol by an Equivalence

A new predicate symbol P involving a single argument \mathbf{E} , thus forming the new predicate $P(\mathbf{E})$, is introduced by an axiomatic equivalence of the following form:

$$\vdash P(\mathbf{E}) \Leftrightarrow \mathbf{D}_{\mathbf{E}}$$

where (1) $\mathbf{D}_{\mathbf{E}}$ is a formula only depending on the variable \mathbf{E} , and (2) $\mathbf{D}_{\mathbf{E}}$ is a formula only containing previously defined symbols of the theory. Traditionally, the new construct, called the *definiendum*, is situated on the left hand side of the equivalence sign while the other side contains the proposed definition, called the *definiens*. For example, in a theory of natural numbers, the binary infix operator “ $<$ ” is defined as follows in terms of “ \leq ”:

$$\vdash \mathbf{E}_1 < \mathbf{E}_2 \Leftrightarrow \mathbf{E}_1 \leq \mathbf{E}_2 \wedge \mathbf{E}_1 \neq \mathbf{E}_2$$

1.3 Defining a New Term Symbol by an Equivalence

A new term symbol T involving a single argument \mathbf{E} is introduced by an axiomatic equivalence of the form:

$$\vdash \mathbf{F} = T(\mathbf{E}) \Leftrightarrow \mathbf{D}_{\mathbf{E},\mathbf{F}}$$

where (1) $\mathbf{D}_{\mathbf{E},\mathbf{F}}$ is a formula only depending on the variables \mathbf{E} and \mathbf{F} , (2) $\mathbf{D}_{\mathbf{E},\mathbf{F}}$ is a formula only containing previously defined symbols of the theory, and finally (3) the following *justifying theorems*

$$\begin{aligned} &\vdash \forall (\mathbf{F}, \mathbf{G}) \cdot (\mathbf{D}_{\mathbf{E},\mathbf{F}} \wedge \mathbf{D}_{\mathbf{E},\mathbf{G}} \Rightarrow \mathbf{F} = \mathbf{G}) \\ &\vdash \exists \mathbf{F} \cdot \mathbf{D}_{\mathbf{E},\mathbf{F}} \end{aligned}$$

must be both provable under the axioms and the previously introduced definitions of the theory. In other words the new term denotes a *unique* entity, which indeed *exists*. We notice that the justifying theorem concerning uniqueness guarantees the criterion of non-creativity since otherwise one could derive an obvious contradiction, and the justifying theorem concerning the existence guarantees that the definition is not void. For example, in a theory of integers, the binary infix symbol of subtraction can be defined as follows:

$$\vdash \mathbf{F} = \mathbf{E}_1 - \mathbf{E}_2 \Leftrightarrow \mathbf{E}_1 = \mathbf{F} + \mathbf{E}_2$$

And the two justifying theorems, which are valid, are then the following:

$$\begin{aligned} &\vdash \forall (\mathbf{F}, \mathbf{G}) \cdot (\mathbf{E}_1 = \mathbf{F} + \mathbf{E}_2 \wedge \mathbf{E}_1 = \mathbf{G} + \mathbf{E}_2 \Rightarrow \mathbf{F} = \mathbf{G}) \\ &\vdash \exists \mathbf{F} \cdot (\mathbf{E}_1 = \mathbf{F} + \mathbf{E}_2) \end{aligned}$$

1.4 Defining a New Term Symbol by an Equality

There is another way of introducing a new term symbol T with an argument \mathbf{E} : this consists in using a simple axiomatic *equality* of the form

$$\vdash \mathsf{T}(\mathbf{E}) = \mathbf{D}_{\mathbf{E}}$$

where (1) $\mathbf{D}_{\mathbf{E}}$ is a formula only depending on the variable \mathbf{E} and (2) $\mathbf{D}_{\mathbf{E}}$ is a formula only containing previously defined symbols of the theory. The advantage of using that kind of definition is that it does not require proving the two justifying theorems since such a definition is the same as the following:

$$\vdash \mathbf{F} = \mathsf{T}(\mathbf{E}) \Leftrightarrow \mathbf{F} = \mathbf{D}_{\mathbf{E}}$$

and thus the justifying theorems become the following, which are logically valid:

$$\begin{aligned} &\vdash \forall (\mathbf{F}, \mathbf{G}) \cdot (\mathbf{F} = \mathbf{D}_{\mathbf{E}} \wedge \mathbf{G} = \mathbf{D}_{\mathbf{E}} \Rightarrow \mathbf{F} = \mathbf{G}) \\ &\vdash \exists \mathbf{F} \cdot (\mathbf{F} = \mathbf{D}_{\mathbf{E}}) \end{aligned}$$

One might wonder then why we are not using that second form systematically in order to introduce new term symbols. The answer is that it is not always possible simply because there are sometimes no explicit term, which can be made easily equal to the new introduced construct (as in the case of subtraction above). But clearly, we shall try to use that second form as often as we can.

1.5 About Recursive “Definitions”

At this point, it is worth departing a little from Suppes presentation of definitions and consider the, so-called, *recursive definitions*, in order to see whether they agree with the concept of definitions we have presented. The typical (and over-used) example of a recursive “definition” is that of the symbol **factorial** introducing the factorial function in a theory of natural numbers:

$$\vdash \mathbf{factorial}(0) = 1 \wedge \mathbf{factorial}(\mathbf{n} + 1) = (\mathbf{n} + 1) \times \mathbf{factorial}(\mathbf{n})$$

Clearly this formulation is not a proper definition (according to the concept of definition presented here) of the unary construct **factorial**(\mathbf{n}) as it does not comply with the above way of defining new term symbols. Nor does it represent a definition of the constant symbol **factorial** for the same reasons. One has however the feeling that it is a correct definition. It is indeed, but not on that form. We have to use an equivalence and thus define the *constant* symbol **factorial** as follows

$$\vdash \mathbf{F} = \text{factorial} \Leftrightarrow \left(\begin{array}{l} \mathbf{F} \in \mathbb{N} \rightarrow \mathbb{N} \quad \wedge \\ \mathbf{F}(0) = 1 \quad \wedge \\ \forall \mathbf{n} \cdot (\mathbf{n} \in \mathbb{N} \Rightarrow \mathbf{F}(\mathbf{n} + 1) = (\mathbf{n} + 1) \times \mathbf{F}(\mathbf{n})) \end{array} \right)$$

This is now a correct definition from a “syntactic” point of view. To make it a genuine definition, one has, of course, to prove both justifying theorems asserting that the definiens introduces a unique function \mathbf{F} . As one knows, it is possible to do so by using an adequate mathematical theory (uniqueness requiring however some more precision).

Because definitions of that form are very useful but rather heavy to formulate as such in practice, people *abbreviate* them by just introducing the essence of them, corresponding to some parts only of the above definiens, hence the “definition” we have given initially. So far so good because, in this example, the proper definition can always be (virtually) reconstructed and the justifying theorems proved. But notice that it is only possible because of the very *shape* of the “definition”. Should this shape be different (not following the inductive construction of the set of natural numbers) then the justification is not possible. The moral of the story is that recursive “definitions” are perfectly correct abbreviations of genuine definitions *provided they obey certain specific rules*. When it is not the case, then, again, such abbreviations are not definitions according to the way this concept is presented here.

1.6 Conditional Definition

Coming back to our main development, we follow again Suppes for the concept of conditional definitions. Sometimes it is necessary to define new symbols that could only be used *provided certain conditions are met*. In the case of a new term symbol introduced by an equivalence (the only case we consider here, the other ones being very close to it), the corresponding rule is as follows: a new term symbol \mathbf{T} involving an argument \mathbf{E} is *conditionally defined* when it is introduced by an axiomatic equivalence of the form:

$$\mathbf{C}_{\mathbf{E}} \vdash \mathbf{F} = \mathbf{T}(\mathbf{E}) \Leftrightarrow \mathbf{D}_{\mathbf{E},\mathbf{F}}$$

where (1) \mathbf{F} is not free in the condition $\mathbf{C}_{\mathbf{E}}$, (2) $\mathbf{D}_{\mathbf{E},\mathbf{F}}$ is a formula only depending on the variables \mathbf{E} and \mathbf{F} , (2) $\mathbf{C}_{\mathbf{E}}$ and $\mathbf{D}_{\mathbf{E},\mathbf{F}}$ are formulae only containing previously defined symbols of the theory, and finally (3) the two justifying theorems

$$\begin{array}{l} \mathbf{C}_{\mathbf{E}} \vdash \forall (\mathbf{F}, \mathbf{G}) \cdot (\mathbf{D}_{\mathbf{E},\mathbf{F}} \wedge \mathbf{D}_{\mathbf{E},\mathbf{G}} \Rightarrow \mathbf{F} = \mathbf{G}) \\ \mathbf{C}_{\mathbf{E}} \vdash \exists \mathbf{F} \cdot \mathbf{D}_{\mathbf{E},\mathbf{F}} \end{array}$$

must be both provable under the axioms and the previously introduced definitions of the theory. As can be seen, the usage of conditional definitions is rather delicate because one must always be careful that they are indeed used under the required conditions. Moreover, they clearly *limit the criterion of eliminability* to the situations where the condition holds. For example, in a theory of real numbers, the binary infix symbol of division is defined conditionally as follows in terms of multiplication:

$$\mathbf{E}_2 \neq 0 \vdash \mathbf{F} = \mathbf{E}_1/\mathbf{E}_2 \Leftrightarrow \mathbf{E}_1 = \mathbf{F} \times \mathbf{E}_2$$

And the two justifying theorems, which are valid, are then the following:

$$\begin{aligned} \mathbf{E}_2 \neq 0 \vdash \forall (\mathbf{F}, \mathbf{G}) \cdot (\mathbf{E}_1 = \mathbf{F} \times \mathbf{E}_2 \wedge \mathbf{E}_1 = \mathbf{G} \times \mathbf{E}_2 \Rightarrow \mathbf{F} = \mathbf{G}) \\ \mathbf{E}_2 \neq 0 \vdash \exists \mathbf{F} \cdot (\mathbf{E}_1 = \mathbf{F} \times \mathbf{E}_2) \end{aligned}$$

One could perhaps argue that conditional definitions are not needed at all, as it would suffice to insert the conditions in the definiens in order to get rid of it. In the case of the division of the reals, this yields

$$\vdash \mathbf{F} = \mathbf{E}_1/\mathbf{E}_2 \Leftrightarrow \mathbf{E}_2 \neq 0 \wedge \mathbf{E}_1 = \mathbf{F} \times \mathbf{E}_2$$

In other words, the very fact of writing $\mathbf{E}_1/\mathbf{E}_2$ would “automatically” imply that \mathbf{E}_2 is different from 0. Let us see however what are now the two justifying theorems. After some obvious transformations, we obtain the following

$$\begin{aligned} \vdash \mathbf{E}_2 \neq 0 \Rightarrow \forall (\mathbf{F}, \mathbf{G}) \cdot (\mathbf{E}_1 = \mathbf{F} \times \mathbf{E}_2 \wedge \mathbf{E}_1 = \mathbf{G} \times \mathbf{E}_2 \Rightarrow \mathbf{F} = \mathbf{G}) \\ \vdash \mathbf{E}_2 \neq 0 \wedge \exists \mathbf{F} \cdot (\mathbf{E}_1 = \mathbf{F} \times \mathbf{E}_2) \end{aligned}$$

The uniqueness theorem clearly holds (it is in fact the same as the previous one). But unfortunately, the existence theorem does not, precisely when \mathbf{E}_2 is equal to 0. An implication is clearly needed here, *not a conjunction*.

The example of the conditional definition of the division of the reals we have just recalled, shows that one must be extremely careful in not asserting things too quickly concerning $\mathbf{E}_1/\mathbf{E}_2$. This could be the case, should we blindly apply the definition of the division without being aware that \mathbf{E}_2 might be equal to 0. As a matter of fact, every pupil quickly learns (his teacher is particularly attentive to that) that “simplification by \mathbf{E}_2 ” in the expression $(\mathbf{E}_1/\mathbf{E}_2) \times \mathbf{E}_2$ can only be performed *provided $\mathbf{E}_2 \neq 0$ holds*.

What is particularly irritating about such matters is that it pollutes our mind while we are trying to solve some “interesting” problem. In an ideal world, would not it be nice to have the possibility to eliminate *once and for all* such special cases and consider only a problem that has thus been simplified? That is, one where we have the *guarantee* that such pathologies can never occur *by the very construction of the statement of the problem*. Notice that this approach is that taken implicitly by the working mathematician: before engaging in a problem, he is very careful of totally eliminating such odd things as division by zero, divergent series, and the like by means of certain *preliminary* treatments.

This problem raised by formal texts containing potentially “ill-defined” expressions such as $3/0$ is, of course, not new. People have proposed many different solution to that question. Reviews of such solutions can be found in [6] and [8]. In what follows, we shall quickly overview some of them before introducing ours.

1.7 Various Approaches

(1) One of the most famous approaches is that advocated by C.B. Jones and his colleagues [2], [5], [7]. It consists in considering a, so-called, three-valued logic,

within which the *valuation* of predicates can be true, false or else undefined. It is a solution aiming at completely integrating the problem of ill-definedness within the formal logic, in contrast to that proposed by others, which tends, on the contrary, to eliminate the problem as much as possible using various artifacts. It has received a considerable interest in various community and in particular in that of VDM. For instance, within that approach, the following predicate is said to be undefined (as well, by the way, as the equality $3/0 = 3/0$, which thus necessitates to redefine equality):

$$2 + 3/0 = 2$$

(2) Another solution [9] consists in totally eliminating conditional definitions by *forcing* them to be always unconditional. Within that approach, one would, for example, systematically give the “value” 0 to a term of the form $\mathbf{E}/0$. In this context, the equality $2 + 3/0 = 2$ is *perfectly valid*, although rather hard to swallow. With that approach however, classical equality is saved, since clearly $3/0 = 3/0$ holds.

(3) Yet another solution [6] consists in claiming that a term of the form $\mathbf{E}/0$ denotes a genuine real number, but that this number is *unknown*. This approach is called *under-specification*. Here the definition is unconditional, as in the previous approach, but also *incomplete*, in contrast with the previous one. What is hard to accept, however, in this approach is the very fact that $3/0$ denotes a real number. This claim is not supported by any classical mathematical construction of such numbers (Dedekind, Cauchy). Notice that there is no means of proving nor refuting the equality $2 + 3/0 = 2$, but, as in the previous one, equality is saved (a real number is indeed equal to itself, thus $3/0=3/0$ holds).

(4) A more drastic approach [9] is one by which the definition of new term symbols is banished. The introduction of new predicate symbols is the only kind of allowed definition. With that approach, an “equality” is redefined for each implicit new term. For instance, division is defined by the predicate symbol Div, where $\text{Div}(\mathbf{F}, \mathbf{E}_1, \mathbf{E}_2)$ stands for $\mathbf{F} = \mathbf{E}_1/\mathbf{E}_2$. It is thus defined as follows:

$$\vdash \text{Div}(\mathbf{F}, \mathbf{E}_1, \mathbf{E}_2) \Leftrightarrow \neg \text{Zero}(\mathbf{E}_2) \wedge \text{Mult}(\mathbf{E}_1, \mathbf{F}, \mathbf{E}_2)$$

where $\text{Zero}(\mathbf{E}_2)$ stands for $\mathbf{E}_2 = 0$ and $\text{Mult}(\mathbf{E}_1, \mathbf{F}, \mathbf{E}_2)$ stands for $\mathbf{E}_1 = \mathbf{F} \times \mathbf{E}_2$. The equality $2 + 3/0 = 2$ becomes the following monster:

$$\text{Zero}(x) \wedge \text{Two}(y) \wedge \text{Three}(z) \wedge \text{Div}(t, z, x) \wedge \text{Plus}(u, y, t) \Rightarrow u = y$$

reducing to the following, which is thus true (since the antecedent of the implication is false):

$$\text{Zero}(x) \wedge \text{Two}(y) \wedge \text{Three}(z) \wedge \neg \text{Zero}(x) \wedge \text{Mult}(z, t, x) \wedge \text{Plus}(u, y, t) \Rightarrow u = y$$

As can be seen, the problem of ill-definedness is now completely evacuated (since there are no terms except variables). Are we, however, ready to accept the rather heavy price?

(5) For the sake of completeness, we must mention the approach [10] taken in IMPS where an undefined predicate is false. It is advocated that this approach is that “commonly used by mathematicians” and “taught to American students in high school and college”.

(6) A “final” solution [3], [4] is based on the idea that a formal language containing potentially ill-defined expressions can be given a semantic interpretation within a three-valued logical domain. So far, it is thus very close to the approach of C.B. Jones. Besides a number of technicalities, the main difference, however, lies in the practical treatment of terms and predicates that are interpreted by an undefined value. Rather than being fully integrated within the logic as in C.B. Jones work, they are only “marginally” integrated as is explained in what follows. In this approach, all terms and predicates are associated with a two-valued logical pseudo-operator, \mathcal{D} , which is given the following interpretation: $\mathcal{D}(\mathbf{F})$ is given the value **false** when the formula (predicate or term) \mathbf{F} is interpreted as **undefined**, and the value **true** otherwise. Notice that $\mathcal{D}(\mathbf{F})$ can never be interpreted as **undefined**. Given a predicate \mathbf{P} , it is argued that $\mathcal{D}(\mathbf{P})$ can always be constructed. If $\mathcal{D}(\mathbf{P})$ is then proved successfully then \mathbf{P} is guaranteed to only contain sub-predicates and sub-terms that are not **undefined**. $\mathcal{D}(\mathbf{P})$ has acted as a *filter*. As a consequence, the interpretation (and proof) of \mathbf{P} can proceed from there within a *pseudo-two-valued logic*: in fact, a three-valued one where **undefined** is now guaranteed to be never encountered. Notice that, in this approach, the equality $2 + 3/0 = 2$ does not pass the filter and is thus simply not considered at all as a formal text to be proved. Interestingly enough, the status of the equality $2 + 3/0 = 2$ as well as that of its negation $2 + 3/0 \neq 2$ can be summarized in the following table for the various approaches we have considered:

	$2 + 3/0 = 2$	$2 + 3/0 \neq 2$
1	undefined	undefined
2	true	false
3	unprovable	unprovable
4	true	true
5	false	false
6	rejected	rejected

1.8 Our Approach

Our approach is very much in the spirit of the last one. It is very close to it, all our results are the same. Only the construction (justification) is *completely different*. We do not make any detour through a three-valued logic. In fact, we do not make any detour through any semantical interpretation at all. We entirely remain within the *syntactic manipulation of proofs*. Our approach just reduces, in a sense, to the development of a mere *proof strategy*.

Our work has been motivated by the very observation, made by the second author of this article, that in the previous approach, when given the task of proving a predicate \mathbf{P} , one proves in fact *more* than just it, one rather proves $\mathcal{D}(\mathbf{P}) \wedge \mathbf{P}$. Let’s call this predicate $\mathcal{N}(\mathbf{P})$. It was then found that the syntactic

transformation $\mathcal{N}(\mathbf{P})$ can be given quite a simple (inductive) definition for the different kind of syntactic constructs of our formal language. It was then easy to reconstruct the filter $\mathcal{D}(\mathbf{P})$ and prove that a successful proof of that filter implies that the proof of \mathbf{P} can then be done safely with our “standard” prover, only slightly modified (simplified) by just *unprotecting* all the conditional definitions it may contain so as to transform them into mere unconditional definitions. This allows one then to possibly fully *eliminate* all the symbols introduced by any kind of definition: we have indeed fulfilled again the criterion of eliminability.

The purpose of this paper is thus to study under which circumstances it is possible to introduce new term symbols by means of conditional definitions in a formal theory as if the definitions in question were unconditional, thus recovering completely the right to subsequently *eliminate these symbols* without bothering about the validity of such an elimination. Taking again the example of $(\mathbf{E}_1/\mathbf{E}_2) \times \mathbf{E}_2$, we want to be in situations where the possibility to simplify by \mathbf{E}_2 is *always* guaranteed.

Although our approach seems to be close to the previous ones, one must be aware that it is, in essence, *quite distinct*. This is so because we do not think in terms of any *valuation*. We do not reason in terms of any semantics. For that reason, a predicate, in our framework, can never be said to be *undefined*, in exactly the same way as it cannot be said to be *true* or *false*. A predicate, at best, is proved or refuted (its negation being proved), or else not (yet) proved nor refuted. What is fundamental then is to determine what the rules of the “proving” game are. In our view, the predicate $0/\mathbf{E} = 0$ is not stamped with any “bad mark”: it is something that we can write. However, it is something that we cannot prove nor refute because we do not know whether \mathbf{E} is equal to 0 or not. What our method tells us is exactly this: as $0/\mathbf{E} = 0$ does not pass the filter (meaning for us that no proof can successfully be conducted), you better give more information (i.e. $\mathbf{E} \neq 0$) so that a proof can be successfully performed. By doing this, we do not modify “classical” logic, we do not introduce any special logic at all. We do not modify equality in any way: $\vdash \mathbf{E} = \mathbf{E}$ is still an axiom, and the Leibniz Law still a rule of inference. The only price we pay is to sometimes reject perfectly provable formulae such as $\mathbf{E}/\mathbf{F} = \mathbf{E}/\mathbf{F}$. Rejecting means, again, that we are asking for more hypotheses.

2 Application to Set Theory

In what follows, we shall illustrate our approach on a specific (although general enough) formal theory, namely Set Theory. In this short section, we recall the way it was introduced in [1] as an *extension* of Predicate Calculus with Equality. We shall also present the proving system we shall use in order to discharge theorems within Set Theory.

2.1 First Order Predicate Calculus with Equality

The language of the classical First Order Predicate Calculus with Equality (and pairs) is defined by the following syntax where the syntactic category *prd* stands

for predicates, vrb for variables, trm for terms, and idt for identifiers. This syntax can be disambiguated by means of the usual ingredients, namely the usage of parentheses and the assignment of precedences in decreasing orders to \neg , \wedge , and \Rightarrow .

$ \begin{aligned} prd & ::= prd \wedge prd \\ & \quad prd \Rightarrow prd \\ & \quad \neg prd \\ & \quad \forall vrb \cdot prd \\ trm & = trm \end{aligned} $	$ \begin{aligned} trm & ::= vrb \\ & \quad trm, trm \\ vrb & ::= idt \\ & \quad vrb, vrb \end{aligned} $
---	---

We suppose that we have at our disposal a Proof System for Predicate Calculus with Equality. Let's call it **PSPC**. This might be, for example, the classical corresponding Sequent Calculus. We shall not present here the axioms and inference rules of this calculus.

2.2 Extending Predicate Calculus to Handle Set Theory

The basic syntax of Set Theory is an extension of the previous one. Now appear the new predicate construct of *set membership* and the new syntactic category of *set*. The three set constructs are the classical cartesian product, comprehension and power set. Notice that, in this syntax, sets are terms whereas some terms are not sets (for instance, pairs).

$ \begin{aligned} prd & ::= \dots \\ & \quad trm \in set \\ trm & ::= \dots \\ & \quad set \end{aligned} $	$ \begin{aligned} set & ::= set \times set \\ & \quad \{ vrb \mid vrb \in set \wedge prd \} \\ & \quad \mathbb{P}(set) \\ & \quad idt \end{aligned} $
---	--

2.3 A Simplified Axiomatization of Set Theory

The following axioms are mere *linguistic* axioms expressing the way set membership can be defined in a straightforward way for the three basic constructs we have introduced (note that in axiom **A2**, \mathbf{x} must not be free in \mathbf{S} , and in axiom **A3**, \mathbf{x} must not be free in \mathbf{S} and \mathbf{T}):

$ \begin{array}{lll} \vdash \mathbf{E}, \mathbf{F} \in \mathbf{S} \times \mathbf{T} & \Leftrightarrow \mathbf{E} \in \mathbf{S} \wedge \mathbf{E} \in \mathbf{T} & \mathbf{A1} \\ \vdash \mathbf{E} \in \{ \mathbf{x} \mid \mathbf{x} \in \mathbf{S} \wedge \mathbf{P}_{\mathbf{x}} \} & \Leftrightarrow \mathbf{E} \in \mathbf{S} \wedge \mathbf{P}_{\mathbf{E}} & \mathbf{A2} \\ \vdash \mathbf{S} \in \mathbb{P}(\mathbf{T}) & \Leftrightarrow \forall \mathbf{x} \cdot (\mathbf{x} \in \mathbf{S} \Rightarrow \mathbf{x} \in \mathbf{T}) & \mathbf{A3} \end{array} $
--

In this axiomatization we have used boldface characters to indicate that the corresponding identifiers denote some *meta-variables*. More precisely, **E** and **F** are meta-variables standing for *trm*, **S** and **T** are meta-variables standing for *set*, **P** is a meta-variable standing for *prd*, and finally **x** is a meta-variable standing for *vrbl*.

Notice that the above axioms, although they may appear to, do *not* constitute proper definitions of the symbol \in . Simply because that symbol does appear on both sides of the equivalence sign. They constitute however three genuine definitions of some specializations of set membership, namely “belonging to a cartesian product”, “belonging to a set comprehension”, and “belonging to a power set”. Such definitions clearly obey our requirements for unconditional definitions and as such *can be eliminated*. Another axiom, which is slightly different from the previous ones, is that of *extensionality* relating set equality and set membership

$$\vdash \mathbf{S} \in \mathbb{P}(\mathbf{T}) \wedge \mathbf{T} \in \mathbb{P}(\mathbf{S}) \Rightarrow \mathbf{S} = \mathbf{T} \quad \mathbf{A4}$$

2.4 Extending the Formal Language by Means of Pure Definitions

Next is a sample of *pure* definitions allowing us to extend our language by introducing respectively disjunction, existential quantification, and set inclusion.

$$\begin{array}{lll} \vdash \mathbf{P} \vee \mathbf{Q} \Leftrightarrow \neg \mathbf{P} \Rightarrow \mathbf{Q} & \mathbf{D1} \\ \vdash \exists \mathbf{x} \cdot \mathbf{P} \Leftrightarrow \neg \forall \mathbf{x} \cdot \neg \mathbf{P} & \mathbf{D2} \\ \vdash \mathbf{S} \subseteq \mathbf{T} \Leftrightarrow \mathbf{S} \in \mathbb{P}(\mathbf{T}) & \mathbf{D3} \end{array}$$

These are indeed pure definitions because: (1) the introduced symbols or constructs are *new*, (2) they appear in the definiendum only, and (3) they are defined in terms of *previous symbols*. As a consequence, they just represent some (very) *useful abbreviations* that can be eliminated *in all circumstances*. From definition **D3** and axiom **A3**, we immediately obtain the following *derived* axiom:

$$\vdash \mathbf{S} \subseteq \mathbf{T} \Leftrightarrow \forall \mathbf{x} \cdot (\mathbf{x} \in \mathbf{S} \Rightarrow \mathbf{x} \in \mathbf{T}) \quad \mathbf{A5}$$

2.5 A Proof System for Set Theory

Our proof system for Predicate Calculus extended with Set Theory is a *straight-forward extension* of **PSPC**. Since any predicate **P** involving sets can obviously be reduced to set memberships, we can use the four axioms **A1** to **A5** to gradually equivalently *eliminate* the various set constructs until there only remains some irreducible “meaningless” set memberships. At this point, we can use our

standard **PSPC** on the resulting predicate **Q**, which, again, is equivalent to the original predicate **P**. An example of such a procedure is given by the proof of the following predicate **P**: $S \subseteq T \wedge T \subseteq U \Rightarrow S \subseteq U$ leading to the proof of the following equivalent predicate **Q**:

$$\forall x \cdot (x \in S \Rightarrow x \in T) \wedge \forall x \cdot (x \in T \Rightarrow x \in U) \Rightarrow \forall x \cdot (x \in S \Rightarrow x \in U)$$

As can be seen, the three predicate $x \in S$, $x \in T$, and $x \in U$ are now “meaningless”. They could be replaced by the three predicates U_x , V_x , and W_x without changing the resulting proof, which, in this case, is completely trivial.

The previous proof technique, which can be used for proving all set-theoretic statements, is not claimed to be “the” proof technique to be used in such a case. We shall use it, however, in this paper because it is very simple. On the other hand, our thesis is that other techniques, using ad-hoc proof rules acting directly at the set-theoretic level, can certainly be deduced from that one (in that the proof rules in question can certainly be proved using it).

What should be clear is that this technique of proof (using a preliminary translation of a set-theoretic predicate into a “pure” predicate) requires that the set-theoretic constructs can *always be eliminated*. This is where the usage of *conditional* definitions might be very problematic since, as we know, such definitions limit the criterion of eliminability. Our main problem thus will be to somehow succeed in using conditional definitions *as if they were not conditional*.

We now enter in the heart of our subject. In the next section, we shall see how “simple” conditional definitions can be eliminated from a formal text to be proved, provided it passes the filter of the standard decision procedure of *type-checking*. In the section to follow after that one, we shall consider more elaborate extensions necessitating a less primitive filter, one that will involve making some *genuine proofs*, which will not be trivially discharged by a decision procedure (although quite simple in general).

3 Filtering the Formal Text with Type-checking

In this section, we shall extend our set-theoretic language by means of a number of definitions aimed at introducing all the classical constructs, namely union, intersection, difference and the like. In doing so, we shall see that such definitions need to be conditional. Fortunately, the conditions are not very “strong”, so that it will be possible to eliminate them after filtering successfully the formal text by means of a simple *decision procedure*: type-checking.

3.1 Extending the Formal Language by Means of Simple Conditional Definitions

Next is another series of extensions of our formal language, extensions corresponding to the classical set-theoretic symbols of union, intersection and difference. These symbols are introduced by means of some *conditional* definitions.

$\mathbf{A} \subseteq \mathbf{S} ; \mathbf{B} \subseteq \mathbf{S} \vdash \mathbf{A} \cup \mathbf{B} = \{x \mid x \in \mathbf{S} \wedge (x \in \mathbf{A} \vee x \in \mathbf{B})\}$	D6
$\mathbf{A} \subseteq \mathbf{S} ; \mathbf{B} \subseteq \mathbf{S} \vdash \mathbf{A} \cap \mathbf{B} = \{x \mid x \in \mathbf{S} \wedge (x \in \mathbf{A} \wedge x \in \mathbf{B})\}$	D7
$\mathbf{A} \subseteq \mathbf{S} ; \mathbf{B} \subseteq \mathbf{S} \vdash \mathbf{A} - \mathbf{B} = \{x \mid x \in \mathbf{S} \wedge (x \in \mathbf{A} \wedge x \notin \mathbf{B})\}$	D8

As previously, the symbols are new, they appear in one side only of the equality sign, and they are all defined in terms of previous symbols. The essential difference with the previous definitions lies in the hypotheses that can be seen on the left hand side of the \vdash symbol. The rôle of such hypotheses is to exhibit a set \mathbf{S} within which \mathbf{A} and \mathbf{B} are included so that union, intersection and difference can be defined by means of *set comprehension* (i.e. as subsets of \mathbf{S}) in an obvious manner. Notice finally that since the introduced symbols indirectly involve the extra variable \mathbf{S} , it has to be proved that these definitions lead to the same results whatever the set in question (unicity).

A consequence of the presence of such hypotheses is that the introduced symbols *cannot always be eliminated*: this can arise only when the hypotheses in question are present. Next is an example where this is the case: this is a formal proof of $A \subseteq S ; B \subseteq S \vdash A \subseteq A \cup B$

(1) $A \subseteq S ; B \subseteq S$	$\vdash A \subseteq A \cup B$	
(2) $A \subseteq S ; B \subseteq S ; x \in A$	$\vdash x \in A \cup B$	A5
(3) $A \subseteq S ; B \subseteq S ; x \in A$	$\vdash x \in \{x \mid x \in S \wedge (x \in A \vee x \in B)\}$	D6
(4) $A \subseteq S ; B \subseteq S ; x \in A$	$\vdash x \in S \wedge (x \in A \vee x \in B)$	A2
(5) $A \subseteq S ; B \subseteq S ; x \in A$	$\vdash x \in A \vee x \in B$	A5

As can be seen, definition **D6** can be applied on line (2) since the proper hypotheses (i.e. $A \subseteq S$ and $B \subseteq S$) are present. By looking more carefully at this proof, we can observe that the transformation of $x \in A \cup B$ at line (2) into $x \in A \vee x \in B$ at line (5) takes three steps, which seems rather tedious for such a triviality. We also notice that, on line (4), we have to discharge the little predicate $x \in S$, which seems rather irrelevant considering what our main problem is. In fact, it is possible to drastically shorten this (and similar) proof by means of the following three rules, which are easily *derivable* from the previous definitions:

$\mathbf{A} \subseteq \mathbf{S} ; \mathbf{B} \subseteq \mathbf{S} \vdash \mathbf{E} \in \mathbf{A} \cup \mathbf{B} \Leftrightarrow \mathbf{E} \in \mathbf{A} \vee \mathbf{E} \in \mathbf{B}$	R1
$\mathbf{A} \subseteq \mathbf{S} ; \mathbf{B} \subseteq \mathbf{S} \vdash \mathbf{E} \in \mathbf{A} \cap \mathbf{B} \Leftrightarrow \mathbf{E} \in \mathbf{A} \wedge \mathbf{E} \in \mathbf{B}$	R2
$\mathbf{A} \subseteq \mathbf{S} ; \mathbf{B} \subseteq \mathbf{S} \vdash \mathbf{E} \in \mathbf{A} - \mathbf{B} \Leftrightarrow \mathbf{E} \in \mathbf{A} \wedge \mathbf{E} \notin \mathbf{B}$	R3

In these rules, the union, intersection and difference of two sets are not defined directly as sets, they are rather defined indirectly by giving an equivalence to the corresponding memberships. What is very interesting about these rules is that the predicate $\mathbf{E} \in \mathbf{S}$ has completely disappeared. In fact, \mathbf{S} is now only

mentioned in the hypotheses, where the predicates $\mathbf{A} \subseteq \mathbf{S}$ and $\mathbf{B} \subseteq \mathbf{S}$ just appear as *witnesses* showing that \mathbf{A} and \mathbf{B} are both included in the same set \mathbf{S} . These rules lead to the following shorter proof, where it can now be observed that \mathbf{S} does not appear on the right hand side of \vdash as was the case in previous proof. As can be seen, the proof is now reduced almost to the essential.

(1) $A \subseteq S ; B \subseteq S$	$\vdash A \subseteq A \cup B$	
(2) $A \subseteq S ; B \subseteq S ; x \in A$	$\vdash x \in A \cup B$	A5
(3) $A \subseteq S ; B \subseteq S ; x \in A$	$\vdash x \in A \vee x \in B$	R1

On this proof, we can observe that Rule **R1** can easily be applied at line (2) because we exactly have the proper hypotheses at hand. Unfortunately, the situation is not always that convenient. For example, suppose we want to prove part of the distributivity of intersection over union, namely $A \subseteq S ; B \subseteq S ; C \subseteq S \vdash (A \cup B) \cap C \subseteq (A \cap C) \cup (B \cap C)$. Since we shall certainly apply rules **R1** and **R2** in order to decompose union and intersection, we shall need the missing hypotheses $A \cup B \subseteq S$, $A \cap C \subseteq S$ and $B \cap C \subseteq S$. In the absence of specific rules, the generation of such hypotheses might be tedious. Fortunately, this can be shortened thanks to the following derived *closure* rules:

$\mathbf{A} \subseteq \mathbf{S} ; \mathbf{B} \subseteq \mathbf{S}$	$\vdash \mathbf{A} \cup \mathbf{B} \subseteq \mathbf{S}$	R4
$\mathbf{A} \subseteq \mathbf{S} ; \mathbf{B} \subseteq \mathbf{S}$	$\vdash \mathbf{A} \cap \mathbf{B} \subseteq \mathbf{S}$	R5
$\mathbf{A} \subseteq \mathbf{S} ; \mathbf{B} \subseteq \mathbf{S}$	$\vdash \mathbf{A} - \mathbf{B} \subseteq \mathbf{S}$	R6

On this occasion, we also notice that similar closure rules can be proved for the basic operators (pairing, cartesian product, set comprehension, and power sets):

$\mathbf{E} \in \mathbf{S} ; \mathbf{F} \in \mathbf{T}$	$\vdash \mathbf{E}, \mathbf{F} \in \mathbf{S} \times \mathbf{T}$	R7
$\mathbf{A} \subseteq \mathbf{S} ; \mathbf{B} \subseteq \mathbf{T}$	$\vdash \mathbf{A} \times \mathbf{B} \subseteq \mathbf{S} \times \mathbf{T}$	R8
$\mathbf{A} \subseteq \mathbf{S}$	$\vdash \{ \mathbf{x} \mid \mathbf{x} \in \mathbf{A} \wedge \mathbf{P} \} \subseteq \mathbf{S}$	R9
$\mathbf{A} \subseteq \mathbf{S}$	$\vdash \mathbb{P}(\mathbf{A}) \subseteq \mathbb{P}(\mathbf{S})$	R10

The proof now proceeds as follows :

(1) $A \subseteq S ; B \subseteq S ; C \subseteq S$	$\vdash (A \cup B) \cap C \subseteq (A \cap C) \cup (B \cap C)$	
(2) $\dots ; A \cup B \subseteq S$	$\vdash (A \cup B) \cap C \subseteq (A \cap C) \cup (B \cap C)$	R4
(3) $\dots ; A \cap C \subseteq S$	$\vdash (A \cup B) \cap C \subseteq (A \cap C) \cup (B \cap C)$	R5
(4) $\dots ; B \cap C \subseteq S$	$\vdash (A \cup B) \cap C \subseteq (A \cap C) \cup (B \cap C)$	R5
(5) $\dots ; x \in (A \cup B) \cap C$	$\vdash x \in (A \cap C) \cup (B \cap C)$	A5
...		

We have organized the proof in such a way that all the needed hypotheses are generated at the beginning, so that the proof can then proceed with everything at hand.

3.2 Simple Type-checking

In the previous proof, the steps (2), (3), and (4) seem to be extremely general. They consist in generating the proper hypotheses corresponding to some of the sub-formulae of the main statement to prove. We notice that the *goal*, situated on the right hand side of \vdash , remains the same during these steps. In the case where more complicated statements have to be proved, it seems that the number of such initial steps might be rather large. On the other hand, these steps, which are clearly very mechanical and systematic, are the consequence of applying the closure rules **R4**, **R5**, and **R6**. Would it be possible to have, in a certain way, these steps generated automatically ?

The answer is yes and it is given by *type-checking*. It is outside the scope of this paper to re-formulate the theory of type-checking as applied to set theory [1]. What will be said here is only the following: provided the statement to prove *does type-check* then each of its sub-terms has a *type*, which, *for the moment*, is a certain *set* to which it belongs. In other words, provided type-checking is successful, then the required hypotheses (and more) are all guaranteed even *without being generated at all*.

3.3 Simplifying the Proof System

The last statement of previous section has an extremely important consequence. Provided, again, type-checking is performed systematically and successfully (we remind the reader that it is performed by an automatic decision procedure), then it is possible to *remove the conditions* in rules **R1**, **R2** and **R3** since we know that the corresponding hypotheses are always (virtually) there. We can thus rephrase and simplify our proof system for set theory (as considered so far) as follows:

$\vdash \mathbf{E}, \mathbf{F} \in \mathbf{S} \times \mathbf{T}$	$\Leftrightarrow \mathbf{E} \in \mathbf{S} \wedge \mathbf{E} \in \mathbf{T}$	A1
$\vdash \mathbf{E} \in \{ \mathbf{x} \mid \mathbf{x} \in \mathbf{S} \wedge \mathbf{P}_{\mathbf{x}} \}$	$\Leftrightarrow \mathbf{E} \in \mathbf{S} \wedge \mathbf{P}_{\mathbf{E}}$	A2
$\vdash \mathbf{S} \in \mathbb{P}(\mathbf{T})$	$\Leftrightarrow \forall \mathbf{x} \cdot (\mathbf{x} \in \mathbf{S} \Rightarrow \mathbf{x} \in \mathbf{T})$	A3
$\vdash \mathbf{S} \subseteq \mathbf{T}$	$\Leftrightarrow \forall \mathbf{x} \cdot (\mathbf{x} \in \mathbf{S} \Rightarrow \mathbf{x} \in \mathbf{T})$	A5
$\vdash \mathbf{S} \subseteq \mathbf{T} \wedge \mathbf{T} \subseteq \mathbf{S}$	$\Leftrightarrow \mathbf{S} = \mathbf{T}$	A4
$\vdash \mathbf{E} \in \mathbf{A} \cup \mathbf{B}$	$\Leftrightarrow \mathbf{E} \in \mathbf{A} \vee \mathbf{E} \in \mathbf{B}$	R1
$\vdash \mathbf{E} \in \mathbf{A} \cap \mathbf{B}$	$\Leftrightarrow \mathbf{E} \in \mathbf{A} \wedge \mathbf{E} \in \mathbf{B}$	R2
$\vdash \mathbf{E} \in \mathbf{A} - \mathbf{B}$	$\Leftrightarrow \mathbf{E} \in \mathbf{A} \wedge \mathbf{E} \notin \mathbf{B}$	R3

Given a set-theoretic predicate to prove, what should be completely clear from these rules is that the set-theoretic constructs contained in it (at least

those envisaged so far) can always be eliminated by a straightforward translation process. The net result is an equivalent predicate containing basic set membership only. The proof can then proceed from there *within pure predicate calculus* (the remaining set membership being “uninterpreted”).

3.4 Constructing the Typing System

Similarly, we can put together the derivable closure rules as follows:

$\mathbf{E} \in \mathbf{S} ; \mathbf{F} \in \mathbf{T} \vdash \mathbf{E}, \mathbf{F} \in \mathbf{S} \times \mathbf{T}$	R7
$\mathbf{A} \subseteq \mathbf{S} ; \mathbf{B} \subseteq \mathbf{T} \vdash \mathbf{A} \times \mathbf{B} \subseteq \mathbf{S} \times \mathbf{T}$	R8
$\mathbf{A} \subseteq \mathbf{S} \vdash \{ \mathbf{x} \mid \mathbf{x} \in \mathbf{A} \wedge \mathbf{P} \} \subseteq \mathbf{S}$	R9
$\mathbf{A} \subseteq \mathbf{S} \vdash \mathbb{P}(\mathbf{A}) \subseteq \mathbb{P}(\mathbf{S})$	R10
$\mathbf{A} \subseteq \mathbf{S} ; \mathbf{B} \subseteq \mathbf{S} \vdash \mathbf{A} \cup \mathbf{B} \subseteq \mathbf{S}$	R4
$\mathbf{A} \subseteq \mathbf{S} ; \mathbf{B} \subseteq \mathbf{S} \vdash \mathbf{A} \cap \mathbf{B} \subseteq \mathbf{S}$	R5
$\mathbf{A} \subseteq \mathbf{S} ; \mathbf{B} \subseteq \mathbf{S} \vdash \mathbf{A} - \mathbf{B} \subseteq \mathbf{S}$	R6

We notice that our closure rules all have the same shape. On the right hand-side of \vdash , we have a predicate expressing that a certain term \mathbf{T} , depending on a number of meta-variables (i.e. $\mathbf{E}, \mathbf{F}, \mathbf{A}, \mathbf{B}$), belongs (since set inclusion can always be transformed in power set membership) to a certain set \mathbf{S} made of *other* meta-variables (i.e. \mathbf{S}, \mathbf{T}). The set \mathbf{S} is either a power set or a cartesian product. On the lefthand side of \vdash , we have a number of hypotheses where each meta-variable of \mathbf{T} is supposed to belong to a certain set involving only the meta-variables of \mathbf{S} , the set in question being made of a simple meta-variable, a cartesian product, or a power set.

From this very shape, these closure rules can be read as follows: if each of the components of a certain construct has a well-defined type then the construct in question also has a well-defined type. As can be seen, a type is either simple, or a cartesian product of types or else a power set of types. Typing gives the *dimensionality* of terms. As in physics where you do not think of adding a distance to a mass, here we do not envisage taking the union of a simple set with, say, the cartesian product of two sets.

3.5 Inference Type-checking

The first proof which we used when starting this discussion was that of $A \subseteq S ; B \subseteq S \vdash A \subseteq A \cup B$. It is now quite clear that, taking account of the simplified proof system we have presented in the previous section, the presence of the hypotheses $A \subseteq S$ and $B \subseteq S$ does not seem to be necessary any more since from now on the rules have no conditions. What are these hypotheses needed for thus ? Well, they are *just* there to ensure the possibility of a correct type-checking of the predicate $A \subseteq A \cup B$.

This seems to be a bit superfluous. We wonder whether we could type-check *without* them. The answer is *inference type-checking*. It gives us exactly what we need, that is, the possibility, when successful, to *infer* from the very shape of a formula that a type *does exist* for each of its sub-terms. This is, in fact, the only thing we need, since in our proofs (as already noticed), we never use the types in question. The ultimate simplification of the proof is thus the following:

$ \begin{array}{ll} \text{(1)} & \vdash A \subseteq A \cup B \\ \text{(2)} & x \in A \vdash x \in A \cup B \quad \mathbf{A5} \\ \text{(3)} & x \in A \vdash x \in A \vee x \in B \quad \mathbf{R1} \end{array} $
--

3.6 Practical Conclusion

At this point, we can formulate our proof strategy as follows for proving a set-theoretic statement **P**:

- | |
|---|
| <ol style="list-style-type: none"> (1) Type-check P (2) Eliminate set membership predicates as much as possible in P (3) Prove the resulting predicate using PSPC |
|---|

4 Filtering the Formal Text with Extended Type-checking

In this section, we now extend our set-theoretic language by means of a number of more elaborate definitions aimed at introducing constructs such as $\mathbf{f}(\mathbf{E})$ (the application of a partial function \mathbf{f} to an argument \mathbf{E}). Other similar extensions could be introduced at this point, we shall not do so however in order to simplify the presentation since the development concerning these other extensions is of the same nature as that presented in what follows.

The corresponding definitions will appear to be more complicated than the ones we have considered so far. And we shall discover that a simple decision procedure such as type-checking *is not sufficient any more*.

4.1 More Extensions

Encouraged by the previous results, we can envisage more extensions such as those corresponding to the set of *binary relations* from one set to another, the *domain* and *range* of a relation, and the set of *partial functions* from one set to another. For this, we shall use our new approach, consisting in defining these symbols by means of pure definitions and then deriving the corresponding closure (typing) rules.

$\vdash \mathbf{S} \leftrightarrow \mathbf{T}$	$= \mathbb{P}(\mathbf{S} \times \mathbf{T})$	D9
$\vdash \mathbf{E} \in \text{dom}(\mathbf{r})$	$\Leftrightarrow \exists \mathbf{y} \cdot (\mathbf{E}, \mathbf{y} \in \mathbf{r})$	D10
$\vdash \mathbf{F} \in \text{ran}(\mathbf{r})$	$\Leftrightarrow \exists \mathbf{x} \cdot (\mathbf{x}, \mathbf{F} \in \mathbf{r})$	D11
$\vdash \text{fnc}(\mathbf{f})$	$\Leftrightarrow \forall (\mathbf{x}, \mathbf{y}, \mathbf{z}) \cdot (\mathbf{x}, \mathbf{y} \in \mathbf{f} \wedge \mathbf{x}, \mathbf{z} \in \mathbf{f} \Rightarrow \mathbf{y} = \mathbf{z})$	D12
$\vdash \mathbf{f} \in \mathbf{S} \rightarrow \mathbf{T}$	$\Leftrightarrow \mathbf{f} \in \mathbf{S} \leftrightarrow \mathbf{T} \wedge \text{fnc}(\mathbf{f})$	D13

The corresponding closure rules are as follows

$\mathbf{A} \subseteq \mathbf{S} ; \mathbf{B} \subseteq \mathbf{T}$	$\vdash \mathbf{A} \leftrightarrow \mathbf{B} \subseteq \mathbf{S} \times \mathbf{T}$	R11
$\mathbf{r} \subseteq \mathbf{S} \times \mathbf{T}$	$\vdash \text{dom}(\mathbf{r}) \subseteq \mathbf{S}$	R12
$\mathbf{r} \subseteq \mathbf{S} \times \mathbf{T}$	$\vdash \text{ran}(\mathbf{r}) \subseteq \mathbf{T}$	R13
$\mathbf{A} \subseteq \mathbf{S} ; \mathbf{B} \subseteq \mathbf{T}$	$\vdash \mathbf{A} \rightarrow \mathbf{B} \subseteq \mathbf{S} \times \mathbf{T}$	R14

4.2 Where Things are Getting Odd Again

Our next extension consists in introducing the classical construct $\mathbf{f}(\mathbf{E})$ denoting the application of a function \mathbf{f} to its argument \mathbf{E} . Notice that the new symbol that is introduced here is certainly not \mathbf{f} , this is in fact a hidden binary symbol, let's call it *apply*, which together with its arguments yields *apply*(\mathbf{f}, \mathbf{E}). As we have done it before, this notation will be introduced by means of a definition. But as the direct definition of $\mathbf{f}(\mathbf{E})$ by means of an equality is not convenient, we shall use an equivalence, namely that of the equality predicate $\mathbf{F} = \mathbf{f}(\mathbf{E})$. The first idea that comes immediately to mind corresponds to the following equivalence:

$$\vdash \mathbf{F} = \mathbf{f}(\mathbf{E}) \Leftrightarrow \mathbf{E}, \mathbf{F} \in \mathbf{f}$$

But, for this to constitute a proper *definition*, we must prove the following *justifying theorems*

$$\begin{aligned} &\vdash \forall (\mathbf{F}, \mathbf{G}) \cdot (\mathbf{E}, \mathbf{F} \in \mathbf{f} \wedge \mathbf{E}, \mathbf{G} \in \mathbf{f} \Rightarrow \mathbf{F} = \mathbf{G}) \\ &\vdash \exists \mathbf{F} \cdot (\mathbf{E}, \mathbf{F} \in \mathbf{f}) \end{aligned}$$

By generalizing the first one to all \mathbf{E} and using definitions **D12** and **D10**, we obtain the following:

$$\begin{aligned} &\vdash \text{fnc}(\mathbf{f}) \\ &\vdash \mathbf{E} \in \text{dom}(\mathbf{f}) \end{aligned}$$

In order to evacuate the justifying theorems (but not the main problem, unfortunately!), our definition has thus to be made conditional:

$\text{fnc}(\mathbf{f}) ; \mathbf{E} \in \text{dom}(\mathbf{f})$	$\vdash \mathbf{F} = \mathbf{f}(\mathbf{E}) \Leftrightarrow \mathbf{E}, \mathbf{F} \in \mathbf{f}$	D14
--	--	------------

The corresponding closure rule can easily be derived from this:

$$\boxed{\mathbf{f} \subseteq \mathbf{S} \times \mathbf{T} ; \mathbf{E} \in \mathbf{S} ; \text{fnc}(\mathbf{f}) ; \mathbf{E} \in \text{dom}(\mathbf{f}) \vdash \mathbf{f}(\mathbf{E}) \in \mathbf{T} \quad \mathbf{R15}}$$

It seems then that all our previous efforts are now vain. Thanks to type-checking, supposedly performed initially and successfully on a predicate to prove, we had been able to transform all our previous conditional definitions into pure definitions. And, here again, comes a definition with some conditions that are of quite a different nature in comparison to those we had previously. As a result, the present conditions cannot be eliminated by assuming simple type-checking. Moreover, the typing system itself is now *polluted* by some conditions, which do not correspond at all to the typing assumptions we had in all our previous closure rules.

Nevertheless, we still believe that the complete elimination of these conditions in the definition and, consequently, in the closure rules is fundamental in order to drastically simplify matters in the proofs. But it seems that we are now *asking for the impossible*.

4.3 Saving the Typing System

Let us start by the apparently simpler task, namely that of de-polluting the typing system. For this let's rewrite the previous closure rule without the last two assumptions.

$$\boxed{\mathbf{f} \subseteq \mathbf{S} \times \mathbf{T} ; \mathbf{E} \in \mathbf{S} \vdash \mathbf{f}(\mathbf{E}) \in \mathbf{T}}$$

This is clearly mathematically wrong. Nevertheless, we have the feeling that there is a certain “truth” in this rule. In fact, under these limited assumptions, it certainly cannot be said that $\mathbf{f}(\mathbf{E})$ belongs to \mathbf{T} . But something can be said, namely that $\mathbf{f}(\mathbf{E})$ has the *dimension* of \mathbf{T} . We shall thus say that $\mathbf{f}(\mathbf{E})$ *is of type* \mathbf{T} , which from now on is *not the same* any more as saying that $\mathbf{f}(\mathbf{E})$ belongs to \mathbf{T} . In order to make the closure rules correct again, we have to introduce two new symbols: \prec to mean *is of type* (this corresponds to \in) and \preceq to mean *is of super-type* (this corresponds to \subseteq). All our closure rules now have to be rewritten as follows:

$$\boxed{\begin{array}{ll} \mathbf{E} \prec \mathbf{S} ; \mathbf{F} \prec \mathbf{T} \vdash \mathbf{E}, \mathbf{F} \prec \mathbf{S} \times \mathbf{T} & \mathbf{R7} \\ \mathbf{A} \preceq \mathbf{S} ; \mathbf{B} \preceq \mathbf{T} \vdash \mathbf{A} \times \mathbf{B} \preceq \mathbf{S} \times \mathbf{T} & \mathbf{R8} \\ \mathbf{A} \preceq \mathbf{S} \vdash \{ \mathbf{x} \mid \mathbf{x} \in \mathbf{A} \wedge \mathbf{P} \} \preceq \mathbf{S} & \mathbf{R9} \\ \mathbf{A} \preceq \mathbf{S} \vdash \mathbb{P}(\mathbf{A}) \preceq \mathbb{P}(\mathbf{S}) & \mathbf{R10} \\ \mathbf{A} \preceq \mathbf{S} ; \mathbf{B} \preceq \mathbf{S} \vdash \mathbf{A} \cup \mathbf{B} \preceq \mathbf{S} & \mathbf{R4} \end{array}}$$

$\mathbf{A} \Vdash \mathbf{S} ; \mathbf{B} \Vdash \mathbf{S}$	\vdash	$\mathbf{A} \cap \mathbf{B} \Vdash \mathbf{S}$	R5
$\mathbf{A} \Vdash \mathbf{S} ; \mathbf{B} \Vdash \mathbf{S}$	\vdash	$\mathbf{A} - \mathbf{B} \Vdash \mathbf{S}$	R6
$\mathbf{A} \Vdash \mathbf{S} ; \mathbf{B} \Vdash \mathbf{T}$	\vdash	$\mathbf{A} \leftrightarrow \mathbf{B} \Vdash \mathbf{S} \times \mathbf{T}$	R11
$\mathbf{r} \Vdash \mathbf{S} \times \mathbf{T}$	\vdash	$\text{dom}(\mathbf{r}) \Vdash \mathbf{S}$	R12
$\mathbf{r} \Vdash \mathbf{S} \times \mathbf{T}$	\vdash	$\text{ran}(\mathbf{r}) \Vdash \mathbf{T}$	R13
$\mathbf{A} \Vdash \mathbf{S} ; \mathbf{B} \Vdash \mathbf{T}$	\vdash	$\mathbf{A} \rightarrow \mathbf{B} \Vdash \mathbf{S} \times \mathbf{T}$	R14
$\mathbf{f} \Vdash \mathbf{S} \times \mathbf{T} ; \mathbf{E} \Vdash \mathbf{S}$	\vdash	$\mathbf{f}(\mathbf{E}) \Vdash \mathbf{T}$	R15

What is important about this new typing is the following: when the typing of a predicate is successful, then as soon as the proper conditions concerning all occurrences of the construct $\mathbf{f}(\mathbf{E})$ in that predicate are met then it can be said that each of its sub-terms *belongs to its type*.

But, clearly, typing *alone* cannot guarantee this. As a consequence, what we had done previously, namely making all our definitions pure, is not valid any more since it was based on the very fact that each sub-term could be made members of certain sets. Things are not yet saved.

Our problem is thus now the following: how could we guarantee *in advance* (by some process to invent) that all these conditions are met on a predicate so that (1) the definition of $\mathbf{f}(\mathbf{E})$ can be made unconditional, (2) the sub-terms of the predicate can then be guaranteed to belong to their types, and (3) by extension, all other definitions can again be made unconditional? As can be seen, things are very intricately mixed here.

4.4 Facing the Wall

Let's start gently then. Suppose we have an "atomic" predicate to prove: one involving only equality, set membership or set inclusion as its main operator. Let's denote this predicate by $\mathbf{A}_{\mathbf{f}(\mathbf{E})}$, thus indicating that we have a single occurrence of $\mathbf{f}(\mathbf{E})$ in it: this is an obvious simplification, which is presently used to make things more easily tractable, it will be subsequently generalized. This predicate is to be proved without assumptions (this is again a simplification).

$$\vdash \mathbf{A}_{\mathbf{f}(\mathbf{E})}$$

Suppose that $\mathbf{A}_{\mathbf{f}(\mathbf{E})}$ does type-check (in the new sense). In order to proceed further (that is, use our pure definitions to perform the proof), we must have the guarantee that the conditions $\text{fnc}(\mathbf{f})$ and $\mathbf{E} \in \text{dom}(\mathbf{f})$ are met. So, we have no choice but to *prove* them, that is:

$$\vdash \text{fnc}(\mathbf{f}) \wedge \mathbf{E} \in \text{dom}(\mathbf{f})$$

Provided the previous proof is successful, then we could in principle put the two conditions as extra hypotheses (by applying the, so-called, *Cut Rule*) and proceed from there under this umbrella. We shall not do so however because we now feel free to use our definition of $\mathbf{f}(\mathbf{E})$ without the extra conditions. These

extra hypotheses are not needed : this is a technique we have learned from the simpler conditional definitions envisaged earlier.

And then we can obviously proceed as before, that is use *all our pure definitions* in order to gradually eliminate all the set-theoretic constructs. But, to begin with, can we eliminate $\mathbf{f}(\mathbf{E})$? The answer is positive thanks to the so-called “one point rule” allowing us to replace equivalently $\mathbf{A}_{\mathbf{f}(\mathbf{E})}$ as follows (where y is supposed to be a “fresh” variable):

$$\vdash \forall y \cdot (y = \mathbf{f}(\mathbf{E}) \Rightarrow \mathbf{A}_y)$$

This leads to the following by applying the (now pure) definition of $y = \mathbf{f}(\mathbf{E})$:

$$\vdash \forall y \cdot ((\mathbf{E}, y) \in \mathbf{f} \Rightarrow \mathbf{A}_y)$$

yielding eventually

$$(\mathbf{E}, y) \in \mathbf{f} \vdash \mathbf{A}_y$$

As can be seen, the term $\mathbf{f}(\mathbf{E})$ has now completely disappeared.

4.5 A Calculus of Syntactic Transformation

To summarize at this point, what we have done is to prove *a little more* than just $\mathbf{A}_{\mathbf{f}(\mathbf{E})}$. Putting the extra together with the main, we have indeed proved

$$\vdash \text{fnc}(\mathbf{f}) \wedge \mathbf{E} \in \text{dom}(\mathbf{f}) \wedge \mathbf{A}_{\mathbf{f}(\mathbf{E})}$$

In other words, we have *transformed* our original predicate \mathbf{A} into a slightly more complicated one. Let’s give a name, $\mathcal{N}(\mathbf{A})$, to this transformation and also a name, $\mathcal{D}(\mathbf{A})$ to the extension. We have thus

$$\mathcal{N}(\mathbf{A}) = \mathcal{D}(\mathbf{A}) \wedge \mathbf{A}$$

It is now easy to generalize this transformation to predicates involving the logical operators. The syntactic transformation \mathcal{N} clearly distributes over conjunction, so that we have $\mathcal{N}(\mathbf{P} \wedge \mathbf{Q}) = \mathcal{N}(\mathbf{P}) \wedge \mathcal{N}(\mathbf{Q})$. The case of the negation is interesting. Suppose we have to prove $\neg \mathbf{A}$. The condition $\mathcal{D}(\mathbf{A})$ for applying the conditional definition remaining obviously the same, we have to prove $\vdash \mathcal{D}(\mathbf{A}) \wedge \neg \mathbf{A}$, that is $\vdash \neg(\mathcal{D}(\mathbf{A}) \Rightarrow \mathbf{A})$. By defining the following dual transformation: $\mathcal{T}(\mathbf{A}) = \mathcal{D}(\mathbf{A}) \Rightarrow \mathbf{A}$. We obtain $\mathcal{N}(\neg \mathbf{A}) = \neg \mathcal{T}(\mathbf{A})$, which can be subsequently generalized to any predicate \mathbf{P} , yielding $\mathcal{N}(\neg \mathbf{P}) = \neg \mathcal{T}(\mathbf{P})$. All this leads to the following syntactic calculus where \mathbf{P} and \mathbf{Q} are predicates and where \mathbf{A} is an atomic predicate, which does not contain any logical operators:

$\mathcal{N}(\mathbf{A}) = \mathcal{D}(\mathbf{A}) \wedge \mathbf{A}$	$\mathcal{T}(\mathbf{A}) = \mathcal{D}(\mathbf{A}) \Rightarrow \mathbf{A}$
$\mathcal{N}(\mathbf{P} \wedge \mathbf{Q}) = \mathcal{N}(\mathbf{P}) \wedge \mathcal{N}(\mathbf{Q})$	$\mathcal{T}(\mathbf{P} \wedge \mathbf{Q}) = \mathcal{T}(\mathbf{P}) \wedge \mathcal{T}(\mathbf{Q})$
$\mathcal{N}(\neg \mathbf{P}) = \neg \mathcal{T}(\mathbf{P})$	$\mathcal{T}(\neg \mathbf{P}) = \neg \mathcal{N}(\mathbf{P})$
$\mathcal{N}(\mathbf{P} \Rightarrow \mathbf{Q}) = \mathcal{T}(\mathbf{P}) \Rightarrow \mathcal{N}(\mathbf{Q})$	$\mathcal{T}(\mathbf{P} \Rightarrow \mathbf{Q}) = \mathcal{N}(\mathbf{P}) \Rightarrow \mathcal{T}(\mathbf{Q})$
$\mathcal{N}(\forall \mathbf{x} \cdot \mathbf{P}) = \forall \mathbf{x} \cdot \mathcal{N}(\mathbf{P})$	$\mathcal{T}(\forall \mathbf{x} \cdot \mathbf{P}) = \forall \mathbf{x} \cdot \mathcal{T}(\mathbf{P})$

4.6 Extended Typing

Unfortunately, although very simple and appealing, the previous calculus is not exactly what we need. In fact, and as already explained, what we are aiming at is to exhibit a preliminary proof (a filter) allowing us to then use the pure definitions within the main proof. This is exactly what we have done on the atomic predicate \mathbf{A} where we first proved $\mathcal{D}(\mathbf{A})$, then \mathbf{A} . This is really that process that has to be generalized. The \mathcal{D} operator, only used so far for atomic predicates, has thus to be generalized to any predicate. Generalizing first the relationship between $\mathcal{D}(\mathbf{A})$, $\mathcal{N}(\mathbf{A})$, and $\mathcal{T}(\mathbf{A})$, we obtain:

$$\begin{array}{l} \mathcal{N}(\mathbf{P}) = \mathcal{D}(\mathbf{P}) \wedge \mathbf{P} \\ \mathcal{T}(\mathbf{P}) = \mathcal{D}(\mathbf{P}) \Rightarrow \mathbf{P} \end{array}$$

By rewriting $\neg\mathcal{T}(\mathbf{P})$ as $\mathcal{D}(\mathbf{P}) \wedge \neg\mathbf{P}$, we can see that $\neg\mathcal{T}(\mathbf{P}) \vee \mathcal{N}(\mathbf{P})$ is equivalent to $\mathcal{D}(\mathbf{P})$. As a consequence, we have

$$\mathcal{D}(\mathbf{P}) = \mathcal{T}(\mathbf{P}) \Rightarrow \mathcal{N}(\mathbf{P})$$

Notice that $\mathcal{D}(\mathbf{P})$ is thus the same as $\mathcal{N}(\neg\mathbf{P}) \vee \mathcal{N}(\mathbf{P})$, which is probably more intuitive than $\mathcal{T}(\mathbf{P}) \Rightarrow \mathcal{N}(\mathbf{P})$, since we have the strong feeling that $\mathcal{D}(\neg\mathbf{P})$ is the same as $\mathcal{D}(\mathbf{P})$. By applying this and the previous calculus, we obtain:

$$\begin{aligned} & \mathcal{D}(\mathbf{P} \wedge \mathbf{Q}) \\ &= \mathcal{T}(\mathbf{P} \wedge \mathbf{Q}) \Rightarrow \mathcal{N}(\mathbf{P} \wedge \mathbf{Q}) \\ &= \mathcal{T}(\mathbf{P}) \wedge \mathcal{T}(\mathbf{Q}) \Rightarrow \mathcal{N}(\mathbf{P}) \wedge \mathcal{N}(\mathbf{Q}) \\ &= \neg\mathcal{T}(\mathbf{P}) \vee \neg\mathcal{T}(\mathbf{Q}) \vee (\mathcal{N}(\mathbf{P}) \wedge \mathcal{N}(\mathbf{Q})) \\ &= (\mathcal{D}(\mathbf{P}) \wedge \neg\mathbf{P}) \vee (\mathcal{D}(\mathbf{Q}) \wedge \neg\mathbf{Q}) \vee (\mathcal{D}(\mathbf{P}) \wedge \mathcal{D}(\mathbf{Q}) \wedge \mathbf{P} \wedge \mathbf{Q}) \\ &= (\mathcal{D}(\mathbf{P}) \wedge \neg\mathbf{P}) \vee (\mathcal{D}(\mathbf{Q}) \wedge \neg\mathbf{Q}) \vee (\mathcal{D}(\mathbf{P}) \wedge \mathcal{D}(\mathbf{Q})); \end{aligned}$$

Similar results can be obtained in the other cases leading to the following table:

$$\begin{array}{l} \mathcal{D}(\mathbf{P} \wedge \mathbf{Q}) = (\mathcal{D}(\mathbf{P}) \wedge \mathcal{D}(\mathbf{Q})) \vee (\mathcal{D}(\mathbf{P}) \wedge \neg\mathbf{P}) \vee (\mathcal{D}(\mathbf{Q}) \wedge \neg\mathbf{Q}) \\ \mathcal{D}(\neg\mathbf{P}) = \mathcal{D}(\mathbf{P}) \\ \mathcal{D}(\mathbf{P} \Rightarrow \mathbf{Q}) = (\mathcal{D}(\mathbf{P}) \wedge \mathcal{D}(\mathbf{Q})) \vee (\mathcal{D}(\mathbf{P}) \wedge \neg\mathbf{P}) \vee (\mathcal{D}(\mathbf{Q}) \wedge \mathbf{Q}) \\ \mathcal{D}(\forall \mathbf{x} \cdot \mathbf{P}_{\mathbf{x}}) = \forall \mathbf{x} \cdot \mathcal{D}(\mathbf{P}_{\mathbf{x}}) \vee \exists x \cdot (\mathcal{D}(\mathbf{P}_{\mathbf{x}}) \wedge \neg\mathbf{P}_{\mathbf{x}}) \end{array}$$

When the predicate \mathbf{A} is atomic, we have to consider several cases:

1. \mathbf{A} does not contain any occurrences of terms of the shape $\mathbf{f}(\mathbf{E})$ or of the shape $\{\mathbf{x} \mid \mathbf{x} \in \mathbf{S} \wedge \mathbf{P}_{\mathbf{x}}\}$,

2. $\mathbf{A}_{\mathbf{f}(\mathbf{E})}$ contains occurrences of terms of the shape $\mathbf{f}(\mathbf{E})$ but \mathbf{f} and \mathbf{E} do not contain occurrences of the shape $\mathbf{g}(\mathbf{F})$ or of the shape $\{\mathbf{x} \mid \mathbf{x} \in \mathbf{S} \wedge \mathbf{P}_x\}$,
3. $\mathbf{A}_{\{\mathbf{x} \mid \mathbf{x} \in \mathbf{S} \wedge \mathbf{P}_x\}}$ contains occurrences of terms of the shape $\{\mathbf{x} \mid \mathbf{x} \in \mathbf{S} \wedge \mathbf{P}_x\}$ but \mathbf{S} does not contain occurrences of the shape $\mathbf{f}(\mathbf{E})$ or of the shape $\{\mathbf{x} \mid \mathbf{x} \in \mathbf{T} \wedge \mathbf{Q}_x\}$

Under these restrictions, which *dictates a certain order* for performing the syntactic transformation \mathcal{D} (it involves starting the transformation \mathcal{D} on the deepest sub-term first), the following table shows the proposed values when the predicate \mathbf{A} is atomic:

$\mathcal{D}(\mathbf{A})$	$= \text{true}$
$\mathcal{D}(\mathbf{A}_{\mathbf{f}(\mathbf{E})})$	$= \text{fnc}(\mathbf{f}) \wedge \mathbf{E} \in \text{dom}(\mathbf{f}) \wedge \forall \mathbf{y} \cdot (\mathbf{y} = \mathbf{f}(\mathbf{E}) \Rightarrow \mathcal{D}(\mathbf{A}_y))$
$\mathcal{D}(\mathbf{A}_{\{\mathbf{x} \mid \mathbf{x} \in \mathbf{S} \wedge \mathbf{P}_x\}})$	$= \forall \mathbf{x} \cdot (\mathbf{x} \in \mathbf{S} \Rightarrow \mathcal{D}(\mathbf{P}_x)) \wedge \forall \mathbf{y} \cdot (\mathbf{y} = \{\mathbf{x} \mid \mathbf{x} \in \mathbf{S} \wedge \mathbf{P}_x\} \Rightarrow \mathcal{D}(\mathbf{A}_y))$

Conversely, given the above definitions of $\mathcal{D}(\mathbf{P})$, it would have been possible to prove that $\mathcal{N}(\mathbf{P})$ is equal to $\mathcal{D}(\mathbf{P}) \wedge \mathbf{P}$, and also that $\mathcal{T}(\mathbf{P})$ is equal to $\mathcal{D}(\mathbf{P}) \Rightarrow \mathbf{P}$.

4.7 Filtering with $\mathcal{D}(\mathbf{P})$

Given a predicate \mathbf{P} , *nothing guarantees* for the moment that $\mathcal{D}(\mathbf{P})$, as we have proposed it in the previous tables, can act as the *filter* we are aiming at: this has to be proved rigorously. The problem can be stated as follows. Suppose we have proved $\mathcal{D}(\mathbf{P})$, then it must be shown that occurrences of terms of the shape $\mathbf{f}(\mathbf{E})$ in \mathbf{P} can be safely *eliminated* using definition **D14**, where the condition $\text{fnc}(\mathbf{f}) \wedge \mathbf{E} \in \text{dom}(\mathbf{f})$ *has been dropped*. For this, we consider the following syntactic transformation $\mathcal{E}(\mathbf{P})$ aiming at transforming a predicate \mathbf{P} by *unconditionally eliminating* all occurrences of terms of the shape $\mathbf{f}(\mathbf{E})$.

$\mathcal{E}(\mathbf{P} \wedge \mathbf{Q})$	$= \mathcal{E}(\mathbf{P}) \wedge \mathcal{E}(\mathbf{Q})$
$\mathcal{E}(\neg \mathbf{P})$	$= \neg \mathcal{E}(\mathbf{P})$
$\mathcal{E}(\mathbf{P} \Rightarrow \mathbf{Q})$	$= \mathcal{E}(\mathbf{P}) \Rightarrow \mathcal{E}(\mathbf{Q})$
$\mathcal{E}(\forall \mathbf{x} \cdot \mathbf{P}_x)$	$= \forall \mathbf{x} \cdot \mathcal{E}(\mathbf{P}_x)$

When the predicate \mathbf{A} is atomic, we have to consider the same cases as above for \mathcal{D} , which, again, dictate to perform the elimination \mathcal{E} in a certain order (this involves making the elimination from the inside up):

$\mathcal{E}(\mathbf{A})$	$= \mathbf{A}$
$\mathcal{E}(\mathbf{A}_{\mathbf{f}(\mathbf{E})})$	$= \forall \mathbf{y} \cdot ((\mathbf{y}, \mathbf{E}) \in \mathbf{f} \Rightarrow \mathcal{E}(\mathbf{A}_y))$
$\mathcal{E}(\mathbf{A}_{\{\mathbf{x} \mid \mathbf{x} \in \mathbf{S} \wedge \mathbf{P}_x\}})$	$= \forall \mathbf{y} \cdot (\mathbf{y} = \{\mathbf{x} \mid \mathbf{x} \in \mathbf{S} \wedge \mathcal{E}(\mathbf{P}_x)\} \Rightarrow \mathcal{E}(\mathbf{A}_y))$

Notice that the elimination of $\mathbf{f}(\mathbf{E})$ in $\mathbf{A}_{\mathbf{f}(\mathbf{E})}$ has been done *without taking care of the condition* $\text{fnc}(\mathbf{f}) \wedge \mathbf{E} \in \text{dom}(\mathbf{f})$. Now the *main result* is this

$$\mathcal{D}(\mathbf{P}) \Rightarrow (\mathbf{P} \Leftrightarrow \mathcal{E}(\mathbf{P}))$$

In other words, we state exactly what we need: provided we have a proof of $\mathcal{D}(\mathbf{P})$, then the elimination of terms of the shape $\mathbf{f}(\mathbf{E})$ in \mathbf{P} can be done with our definition **D14** *where the required conditions have been dropped*, thus resulting in an equivalent predicate. The proof is done by structural induction in the appendix.

4.8 Strengthening the condition $\mathcal{D}(\mathbf{P})$

The calculation of $\mathcal{D}(\mathbf{P})$ we have done in previous section is not very encouraging (far too complicated). In particular the disjunctive forms we have obtained are rather repulsive. The idea then is to have a stronger but simpler filter called $\mathcal{L}(\mathbf{P})$. For this, we only select the two first disjuncts of $\mathcal{D}(\mathbf{P} \wedge \mathbf{Q})$ and $\mathcal{D}(\mathbf{P} \Rightarrow \mathbf{Q})$, and the first disjunct of $\mathcal{D}(\forall \mathbf{x} \cdot \mathbf{P})$. This leads to the following where we now have some conjunctive forms leading to an easy *decomposition*:

$$\begin{aligned} \mathcal{L}(\mathbf{P} \wedge \mathbf{Q}) &= \mathcal{L}(\mathbf{P}) \wedge (\mathbf{P} \Rightarrow \mathcal{L}(\mathbf{Q})) \\ \mathcal{L}(\neg \mathbf{P}) &= \mathcal{L}(\mathbf{P}) \\ \mathcal{L}(\mathbf{P} \Rightarrow \mathbf{Q}) &= \mathcal{L}(\mathbf{P}) \wedge (\mathbf{P} \Rightarrow \mathcal{L}(\mathbf{Q})) \\ \mathcal{L}(\forall \mathbf{x} \cdot \mathbf{P}) &= \forall \mathbf{x} \cdot \mathcal{L}(\mathbf{P}) \end{aligned}$$

In the atomic cases, we have the usual restrictions and the following straightforward transformation:

$$\begin{aligned} \mathcal{L}(\mathbf{A}) &= \text{true} \\ \mathcal{L}(\mathbf{A}_{\mathbf{f}(\mathbf{E})}) &= \text{fnc}(\mathbf{f}) \wedge \mathbf{E} \in \text{dom}(\mathbf{f}) \wedge \forall \mathbf{y} \cdot (\mathbf{y} = \mathbf{f}(\mathbf{E}) \Rightarrow \mathcal{L}(\mathbf{A}_{\mathbf{y}})) \\ \mathcal{L}(\mathbf{A}_{\{\mathbf{x} \mid \mathbf{x} \in \mathbf{S} \wedge \mathbf{P}_{\mathbf{x}}\}}) &= \forall \mathbf{x} \cdot (\mathbf{x} \in \mathbf{S} \Rightarrow \mathcal{L}(\mathbf{P}_{\mathbf{x}})) \wedge \\ &\quad \forall \mathbf{y} \cdot (\mathbf{y} = \{\mathbf{x} \mid \mathbf{x} \in \mathbf{S} \wedge \mathbf{P}_{\mathbf{x}}\} \Rightarrow \mathcal{L}(\mathbf{A}_{\mathbf{y}})) \end{aligned}$$

The following can then easily be proved by structural induction:

$$\mathcal{L}(\mathbf{P}) \Rightarrow \mathcal{D}(\mathbf{P})$$

4.9 Proving $\mathcal{L}(\mathbf{P})$

The previous result allows us to use $\mathcal{L}(\mathbf{P})$ rather than $\mathcal{D}(\mathbf{P})$ as a filter. The former is stronger but more tractable than the latter. An interesting question is now that concerning the proof of $\mathcal{L}(\mathbf{P})$. Can we prove it by using the simplified system or are we required to be careful in expanding the potential conditional definitions it contains? The question is essentially that of proving

$$\mathcal{L}(\mathcal{L}(\mathbf{P}))$$

since, should it be the case, then we have $\mathcal{D}(\mathcal{L}(\mathbf{P}))$ and we can thus prove $\mathcal{L}(\mathbf{P})$ with the simplified system. The proof is, again, by structural induction.

4.10 Practical Conclusion

At this point, we can re-formulate our proof strategy as follows for proving a set-theoretic statement \mathbf{P} :

- (1) Type-check \mathbf{P}
- (2) Calculate $\mathcal{L}(\mathbf{P})$
- (3) Eliminate set-theoretic formulae as much as possible in $\mathcal{L}(\mathbf{P})$
- (4) Prove the resulting predicate using **PSPC**
- (5) Eliminate set-theoretic formulae as much as possible in \mathbf{P}
- (6) Prove the resulting predicate using **PSPC**

5 An Example

Suppose we want to prove the following predicate \mathbf{P} :

$$f \in S \leftrightarrow T \wedge g \in T \leftrightarrow U \wedge x \in \text{dom}(f) \Rightarrow g(f(x)) \in U$$

The type checking of \mathbf{P} is clearly successful. Let's first compute $\mathcal{L}(\mathbf{P})$:

$$\left(\begin{array}{l} f \in S \leftrightarrow T \wedge \\ g \in T \leftrightarrow U \wedge \\ x \in \text{dom}(f) \end{array} \right) \Rightarrow \left(\begin{array}{l} \text{fnc}(f) \wedge \\ x \in \text{dom}(f) \wedge \\ \text{fnc}(g) \wedge \\ f(x) \in \text{dom}(g) \end{array} \right)$$

Part of the consequent is easily discharged. It just remains $f(x) \in \text{dom}(g)$, which we can expand using the “one point” rule, and the purified definitions of $f(x)$ and $\text{dom}(g)$. After some simplifications, this yields the following, which we obviously *cannot prove*:

$$\exists y \cdot (x, y \in f) \Rightarrow \forall y \cdot (x, y \in f \Rightarrow \exists z \cdot (y, z \in g))$$

In other words, our predicate \mathbf{P} does not pass the filter $\mathcal{L}(\mathbf{P})$. At this point, we figure out that we had forgotten an assumption concerning the relationship between the range of f and the domain of g . Here is the modification of our original predicate \mathbf{P} that becomes the new predicate \mathbf{Q} (we have added the antecedent $\text{ran}(f) \subseteq \text{dom}(g)$):

$$\left(\begin{array}{l} f \in S \leftrightarrow T \wedge \\ g \in T \leftrightarrow U \wedge \\ \text{ran}(f) \subseteq \text{dom}(g) \wedge \\ x \in \text{dom}(f) \end{array} \right) \Rightarrow g(f(x)) \in U$$

The calculation of the corresponding new $\mathcal{L}(\mathbf{Q})$ leads to the proof of the following, which now holds trivially:

$$\begin{aligned} & \forall y \cdot (\exists x \cdot (x, y \in f) \Rightarrow \exists z \cdot (y, z \in g)) \wedge \\ & \exists y \cdot (x, y \in f) \\ \Rightarrow & \\ & \forall y \cdot (x, y \in f \Rightarrow \exists z \cdot (y, z \in g)) \end{aligned}$$

At this point, we proceed with the proof of \mathbf{Q} . And for doing this, we use the simplified system. After some simplifications, this leads to proving the following, which holds trivially:

$$\begin{aligned} & \forall(x, y) \cdot (x, y \in f \Rightarrow x \in S \wedge y \in T) \wedge \\ & \forall(y, z) \cdot (y, z \in g \Rightarrow y \in T \wedge z \in U) \\ \Rightarrow & \\ & \forall(y, z) \cdot (x, y \in f \wedge y, z \in g \Rightarrow z \in U) \end{aligned}$$

Acknowledgements: We thank B. Stoddart for numerous very interesting electronic discussions.

References

1. J.R. Abrial. *The B-Book: Assigning Programs to Meanings*. Cambridge University Press (1996).
2. H. Barringer, J.H. Cheng, C.B. Jones. *A Logic Covering Undefinedness in Program Proofs*. Acta Informatica 21: 251-269 (1984).
3. P. Behm, L. Burdy, J.M. Meynadier. *Well defined B*. Second B International Conference. (Bert editor) LNCS 1393 Springer (1998).
4. L. Burdy *Traitement des expressions dépourvues de sens de la théorie des ensembles*. Thèse de Doctorat (2000).
5. J.H. Cheng, C.B. Jones *On the Usability of Logics which Handle Partial Functions*. Proceedings of Third Refinement Workshop. (1990).
6. D. Gries. *Foundations for Computational Logic in Mathematical Methods in Program Development* (M. Broy and B. Schieder Editors). Springer (1996).

7. C.B. Jones *Partial Functions and Logics: a warning*. Information Processing Letter 54 (1995).
8. B. Stoddart, S. Dunne, A. Galloway. *Undefined Expressions and Logic in Z and B*. Formal Methods in System Design, 15 (1999).
9. P. Suppes. *Introduction to Logic*. Wadsworth International Group (1957).
10. W.M. Farmer and J.D. Guttman. *A Set Theory with Support for Partial Functions in Partiality and Modality*. (E. Thijsse, F. Lepage, and H. Wansing Editors). Special issue of *Logica Studia* (2000).

PROOF of $\mathcal{D}(\mathbf{P}) \Rightarrow (\mathbf{P} \Leftrightarrow \mathcal{E}(\mathbf{P}))$. The proof is by structural induction.

(1) *Base Case*: \mathbf{A} does not contain any occurrences of terms of the shape $\mathbf{f}(\mathbf{E})$ or of the shape $\{\mathbf{x} \mid \mathbf{x} \in \mathbf{S} \wedge \mathbf{P}_{\mathbf{x}}\}$. We have to prove $\mathcal{D}(\mathbf{A}) \Rightarrow (\mathbf{A} \Leftrightarrow \mathcal{E}(\mathbf{A}))$, which is obvious.

(2) *Inductive Case* : $\mathbf{A}_{\mathbf{f}(\mathbf{E})}$. We assume the inductive hypothesis

$$\forall \mathbf{y} \cdot (\mathcal{D}(\mathbf{A}_{\mathbf{y}}) \Rightarrow (\mathbf{A}_{\mathbf{y}} \Leftrightarrow \mathcal{E}(\mathbf{A}_{\mathbf{y}})))$$

and we have to prove $\mathcal{D}(\mathbf{A}_{\mathbf{f}(\mathbf{E})}) \Rightarrow (\mathbf{A}_{\mathbf{f}(\mathbf{E})} \Leftrightarrow \mathcal{E}(\mathbf{A}_{\mathbf{f}(\mathbf{E})}))$. For this, we assume $\mathcal{D}(\mathbf{A}_{\mathbf{f}(\mathbf{E})})$, that is

$$\text{fnc}(\mathbf{f}) \wedge \mathbf{E} \in \text{dom}(\mathbf{f}) \wedge \forall \mathbf{y} \cdot (\mathbf{y} = \mathbf{f}(\mathbf{E}) \Rightarrow \mathcal{D}(\mathbf{A}_{\mathbf{y}}))$$

and we have to prove $\mathbf{A}_{\mathbf{f}(\mathbf{E})} \Leftrightarrow \mathcal{E}(\mathbf{A}_{\mathbf{f}(\mathbf{E})})$, that is

$$\mathbf{A}_{\mathbf{f}(\mathbf{E})} \Leftrightarrow \forall \mathbf{y} \cdot ((\mathbf{y}, \mathbf{E}) \in \mathbf{f} \Rightarrow \mathcal{E}(\mathbf{A}_{\mathbf{y}}))$$

By applying the *conditional* definition **D14** (which we can do since the proper conditions, namely $\text{fnc}(\mathbf{f}) \wedge \mathbf{E} \in \text{dom}(\mathbf{f})$, are assumed), we obtain equivalently:

$$\mathbf{A}_{\mathbf{f}(\mathbf{E})} \Leftrightarrow \forall \mathbf{y} \cdot (\mathbf{y} = \mathbf{f}(\mathbf{E}) \Rightarrow \mathcal{E}(\mathbf{A}_{\mathbf{y}}))$$

But, according to the assumption $\forall \mathbf{y} \cdot (\mathbf{y} = \mathbf{f}(\mathbf{E}) \Rightarrow \mathcal{D}(\mathbf{A}_{\mathbf{y}}))$ and the induction hypothesis $\forall \mathbf{y} \cdot (\mathcal{D}(\mathbf{A}_{\mathbf{y}}) \Rightarrow (\mathbf{A}_{\mathbf{y}} \Leftrightarrow \mathcal{E}(\mathbf{A}_{\mathbf{y}})))$ we have $\forall \mathbf{y} \cdot (\mathbf{y} = \mathbf{f}(\mathbf{E}) \Rightarrow (\mathbf{A}_{\mathbf{y}} \Leftrightarrow \mathcal{E}(\mathbf{A}_{\mathbf{y}})))$, we can thus replace $\mathcal{E}(\mathbf{A}_{\mathbf{y}})$ by $\mathbf{A}_{\mathbf{y}}$ in $\forall \mathbf{y} \cdot (\mathbf{y} = \mathbf{f}(\mathbf{E}) \Rightarrow \mathcal{E}(\mathbf{A}_{\mathbf{y}}))$, leading to the following, which is obvious (one point rule):

$$\mathbf{A}_{\mathbf{f}(\mathbf{E})} \Leftrightarrow \forall \mathbf{y} \cdot (\mathbf{y} = \mathbf{f}(\mathbf{E}) \Rightarrow \mathbf{A}_{\mathbf{y}})$$

(3) *Inductive Case* : $\mathbf{A}_{\{\mathbf{x} \mid \mathbf{x} \in \mathbf{S} \wedge \mathbf{P}_{\mathbf{x}}\}}$. We assume both inductive hypotheses

$$\begin{aligned} \forall \mathbf{y} \cdot (\mathcal{D}(\mathbf{A}_{\mathbf{y}}) \Rightarrow (\mathbf{A}_{\mathbf{y}} \Leftrightarrow \mathcal{E}(\mathbf{A}_{\mathbf{y}}))) \\ \forall \mathbf{x} \cdot (\mathcal{D}(\mathbf{P}_{\mathbf{x}}) \Rightarrow (\mathbf{P}_{\mathbf{x}} \Leftrightarrow \mathcal{E}(\mathbf{P}_{\mathbf{x}}))) \end{aligned}$$

and we have to prove $\mathcal{D}(\mathbf{A}_{\{\mathbf{x} \mid \mathbf{x} \in \mathbf{S} \wedge \mathbf{P}_{\mathbf{x}}\}}) \Rightarrow (\mathbf{A}_{\{\mathbf{x} \mid \mathbf{x} \in \mathbf{S} \wedge \mathbf{P}_{\mathbf{x}}\}} \Leftrightarrow \mathcal{E}(\mathbf{A}_{\{\mathbf{x} \mid \mathbf{x} \in \mathbf{S} \wedge \mathbf{P}_{\mathbf{x}}\}}))$. For this, we assume $\mathcal{D}(\mathbf{A}_{\{\mathbf{x} \mid \mathbf{x} \in \mathbf{S} \wedge \mathbf{P}_{\mathbf{x}}\}})$, that is

$$\forall \mathbf{x} \cdot (\mathbf{x} \in \mathbf{S} \Rightarrow \mathcal{D}(\mathbf{P}_{\mathbf{x}})) \wedge \forall \mathbf{y} \cdot (\mathbf{y} = \{\mathbf{x} \mid \mathbf{x} \in \mathbf{S} \wedge \mathbf{P}_{\mathbf{x}}\} \Rightarrow \mathcal{D}(\mathbf{A}_{\mathbf{y}}))$$

and we have to prove $\mathbf{A}_{\{\mathbf{x} \mid \mathbf{x} \in \mathbf{S} \wedge \mathbf{P}_{\mathbf{x}}\}} \Leftrightarrow \mathcal{E}(\mathbf{A}_{\{\mathbf{x} \mid \mathbf{x} \in \mathbf{S} \wedge \mathbf{P}_{\mathbf{x}}\}})$, that is

$$\mathbf{A}_{\{\mathbf{x}|\mathbf{x}\in\mathbf{S}\wedge\mathbf{P}_x\}} \Leftrightarrow \forall \mathbf{y} \cdot (\mathbf{y} = \{\mathbf{x}|\mathbf{x}\in\mathbf{S}\wedge\mathcal{E}(\mathbf{P}_x)\} \Rightarrow \mathcal{E}(\mathbf{A}_y))$$

According to the first assumption $\forall \mathbf{x} \cdot (\mathbf{x} \in \mathbf{S} \Rightarrow \mathcal{D}(\mathbf{P}_x))$ and the second inductive hypothesis $\forall \mathbf{x} \cdot (\mathcal{D}(\mathbf{P}_x) \Rightarrow (\mathbf{P}_x \Leftrightarrow \mathcal{E}(\mathbf{P}_x)))$, we have $\forall \mathbf{x} \cdot (\mathbf{x} \in \mathbf{S} \Rightarrow (\mathbf{P}_x \Leftrightarrow \mathcal{E}(\mathbf{P}_x)))$, we can thus replace $\mathcal{E}(\mathbf{P}_x)$ by \mathbf{P}_x in $\{\mathbf{x}|\mathbf{x}\in\mathbf{S}\wedge\mathcal{E}(\mathbf{P}_x)\}$ and obtain equivalently:

$$\mathbf{A}_{\{\mathbf{x}|\mathbf{x}\in\mathbf{S}\wedge\mathbf{P}_x\}} \Leftrightarrow \forall \mathbf{y} \cdot (\mathbf{y} = \{\mathbf{x}|\mathbf{x}\in\mathbf{S}\wedge\mathbf{P}_x\} \Rightarrow \mathcal{E}(\mathbf{A}_y))$$

But, according to the second assumption $\forall \mathbf{y} \cdot (\mathbf{y} = \{\mathbf{x}|\mathbf{x}\in\mathbf{S}\wedge\mathbf{P}_x\} \Rightarrow \mathcal{D}(\mathbf{A}_y))$ and the first induction hypothesis $\forall \mathbf{y} \cdot (\mathcal{D}(\mathbf{A}_y) \Rightarrow (\mathbf{A}_y \Leftrightarrow \mathcal{E}(\mathbf{A}_y)))$ we have $\forall \mathbf{y} \cdot (\mathbf{y} = \{\mathbf{x}|\mathbf{x}\in\mathbf{S}\wedge\mathbf{P}_x\} \Rightarrow (\mathbf{A}_y \Leftrightarrow \mathcal{E}(\mathbf{A}_y)))$, we can thus replace $\mathcal{E}(\mathbf{A}_y)$ by \mathbf{A}_y in $\forall \mathbf{y} \cdot (\mathbf{y} = \{\mathbf{x}|\mathbf{x}\in\mathbf{S}\wedge\mathbf{P}_x\} \Rightarrow \mathcal{E}(\mathbf{A}_y))$, leading to the following, which is obvious (one point rule)

$$\mathbf{A}_{\{\mathbf{x}|\mathbf{x}\in\mathbf{S}\wedge\mathbf{P}_x\}} \Leftrightarrow \forall \mathbf{y} \cdot (\mathbf{y} = \{\mathbf{x}|\mathbf{x}\in\mathbf{S}\wedge\mathbf{P}_x\} \Rightarrow \mathbf{A}_y)$$

(4) *Inductive Case* : $\mathbf{P} \wedge \mathbf{Q}$. We assume both inductive hypotheses

$$\begin{aligned} \mathcal{D}(\mathbf{P}) &\Rightarrow (\mathbf{P} \Leftrightarrow \mathcal{E}(\mathbf{P})) \\ \mathcal{D}(\mathbf{Q}) &\Rightarrow (\mathbf{Q} \Leftrightarrow \mathcal{E}(\mathbf{Q})) \end{aligned}$$

and we have to prove

$$\mathcal{D}(\mathbf{P} \wedge \mathbf{Q}) \Rightarrow (\mathbf{P} \wedge \mathbf{Q} \Leftrightarrow \mathcal{E}(\mathbf{P} \wedge \mathbf{Q}))$$

We thus now assume $\mathcal{D}(\mathbf{P} \wedge \mathbf{Q})$, that is

$$(\mathcal{D}(\mathbf{P}) \wedge \mathcal{D}(\mathbf{Q})) \vee (\mathcal{D}(\mathbf{P}) \wedge \neg \mathbf{P}) \vee (\mathcal{D}(\mathbf{Q}) \wedge \neg \mathbf{Q})$$

and we have to prove $\mathbf{P} \wedge \mathbf{Q} \Leftrightarrow \mathcal{E}(\mathbf{P} \wedge \mathbf{Q})$, that is

$$\mathbf{P} \wedge \mathbf{Q} \Leftrightarrow \mathcal{E}(\mathbf{P}) \wedge \mathcal{E}(\mathbf{Q})$$

The disjunctive shape of the assumption suggests a proof by cases. The first case, $\mathcal{D}(\mathbf{P}) \wedge \mathcal{D}(\mathbf{Q})$, is trivial according to the two inductive hypotheses. We assume then the second case:

$$\mathcal{D}(\mathbf{P}) \wedge \neg \mathbf{P}$$

From $\mathcal{D}(\mathbf{P})$ and the first induction hypothesis, we deduce

$$\mathbf{P} \wedge \mathbf{Q} \Leftrightarrow \mathbf{P} \wedge \mathcal{E}(\mathbf{Q})$$

which holds since we have $\neg \mathbf{P}$. The last case, $\mathcal{D}(\mathbf{Q}) \wedge \neg \mathbf{Q}$, is proved in a similar fashion.

(5) *Inductive Case*: $\neg \mathbf{P}$. Proof omitted.

(6) *Inductive Case*: $\mathbf{P} \Rightarrow \mathbf{Q}$. Proof omitted.

(7) *Inductive Case*: $\forall \mathbf{x} \cdot \mathbf{P}_x$. We assume the inductive hypothesis

$$\forall \mathbf{x} \cdot (\mathcal{D}(\mathbf{P}_{\mathbf{x}}) \Rightarrow (\mathbf{P}_{\mathbf{x}} \Leftrightarrow \mathcal{E}(\mathbf{P}_{\mathbf{x}})))$$

and we have to prove

$$\mathcal{D}(\forall \mathbf{x} \cdot \mathbf{P}_{\mathbf{x}}) \Rightarrow (\forall \mathbf{x} \cdot \mathbf{P}_{\mathbf{x}} \Leftrightarrow \mathcal{E}(\forall \mathbf{x} \cdot \mathbf{P}_{\mathbf{x}}))$$

We thus now assume $\mathcal{D}(\forall \mathbf{x} \cdot \mathbf{P}_{\mathbf{x}})$, that is

$$\forall \mathbf{x} \cdot \mathcal{D}(\mathbf{P}_{\mathbf{x}}) \vee \exists \mathbf{x} \cdot (\mathcal{D}(\mathbf{P}_{\mathbf{x}}) \wedge \neg \mathbf{P}_{\mathbf{x}})$$

and we have to prove $\forall \mathbf{x} \cdot \mathbf{P}_{\mathbf{x}} \Leftrightarrow \mathcal{E}(\forall \mathbf{x} \cdot \mathbf{P}_{\mathbf{x}})$, that is

$$\forall \mathbf{x} \cdot \mathbf{P}_{\mathbf{x}} \Leftrightarrow \forall \mathbf{x} \cdot \mathcal{E}(\mathbf{P}_{\mathbf{x}})$$

The disjunctive shape of the assumption suggests a proof by cases. The first case, $\forall \mathbf{x} \cdot \mathcal{D}(\mathbf{P}_{\mathbf{x}})$ is trivial. We assume the second case

$$\exists \mathbf{x} \cdot (\mathcal{D}(\mathbf{P}_{\mathbf{x}}) \wedge \neg \mathbf{P}_{\mathbf{x}})$$

which allows us to prove easily the following, which is equivalent to $\forall \mathbf{x} \cdot \mathbf{P}_{\mathbf{x}} \Leftrightarrow \forall \mathbf{x} \cdot \mathcal{E}(\mathbf{P}_{\mathbf{x}})$

$$\exists \mathbf{x} \cdot \neg \mathbf{P}_{\mathbf{x}} \Leftrightarrow \exists \mathbf{x} \cdot \neg \mathcal{E}(\mathbf{P}_{\mathbf{x}})$$

QED