Non-strict don't care Algebras and Specifications

Egidio Astesiano and Maura Cerioli

Dipartimento di Matematica - Università di Genova
Via L.B. Alberti 4 - 16132 Genova Italy
e-mail: {astes, cerioli}@igecuniv.bitnet

Abstract. *Non-strict don't care functions, whose foremost representative is the ubiquitous if_then_else, play an essential role in computer science. As for what concerns semantics, they can be modelled by their totalizations with the appropriate use of elements representing undefinedness, as D. Scott has shown in his denotational approach. The situation is not so straightforward when we consider non-strict functions in the context of an algebraic framework. In this paper, after presenting the basic properties of the category of non-strict algebras, we explore the relationship between non-strict don't care and total algebras. Then the conditional algebraic specifications are investigated; it is shown that non-strict conditional specifications are equivalent to disjunctive specifications and necessary and sufficient conditions for the existence of initial models are given. Since non-strict don't care specifications generalize both the total and the partial case, it is shown how the results about initiality can be obtained as specializations.*

Introduction

Functions like the well known *if_then_else* are called non-strict since they can be defined even over tuples where some argument is undefined. For example *if true then a else b* = *a* no matter what is *b*. Note the difference with partial functions, which satisfy the strictness condition $D(f(x_1,\ldots,x_n)) \supset D(x_i)$ for i=1,…,n. What is important to stress in the case of the *if_then_else* operation is the don't care feature; if the first argument is *true* then the third argument can be whatever and even missing (corresponding for example to a non-terminating computation). Non-strict don't care functions are common feature of programming languages; indeed a non-defined function built over non-strict predefined functions like *if_then_else* may be non-strict; moreover in many languages, notably Ada, non-strict built-in functions abound.

We can model this situation defining partial tuples and an ordering over them. For example the tuple (*true*,*a*,?), where the symbol ? denotes a missing argument, is less than or equal (*true*,*a*,*b*) for any *b*. Thus a don't care function *f* can be characterized by a monotonicity condition (denoting tuples by an underbar)

$\underline{d}_1 \leq \underline{d}_2$ and $f(\underline{d}_1)$ defined $\supset f(\underline{d}_1) = f(\underline{d}_2)$.

This work has been partially supported by Esprit-BRA-WG COMPASS and by MURST 40%

All this is well known in denotational semantics where undefinedness is denoted by $\perp$, domains are cpo's with least element $\perp$ and functions are monotonic.

However, if we consider algebras with non-strict (don't care) functions, it is not true that we can just identify them with a subclass of total algebras, if we want algebraic properties to be preserved. Indeed we can think of associating with a non-strict algebra A its totalization $A_\perp$; however this association does not define a functor from non-strict into total algebras.

After defining in section 1 the category of non-strict algebras and presenting its basic properties, in section 2 we analyze the relationship between non-strict and total algebras. It is shown that the correct relationship is provided by a free construction, similar to the one indicated by Poigné in [P] in the case of a partial to total translation. A deeper characterization of that correspondence can be given in terms of the recently introduced concepts of simulation of institution [AC2] and of map of institutions [M] and we refer to the full paper [AC3] for this.

Having settled the model theoretical aspects, we turn our attention to algebraic specifications with non-strict functions, henceforth non-strict specifications. Many problems arise: first of all not even simple equational specifications are always consistent (ie admit models), the usual conditional specifications do not always admit initial models and moreover no simple equational deduction system, generalizing those for the partial and the total case, seems at hand.

We try to clarify the matter as follows.
It is shown that the monotonicity condition necessarily introduces hidden disjunctions. Hence we pass to consider disjunctive specifications giving necessary and sufficient conditions for the existence of initial models, from which, by specialization, necessary and sufficient conditions, for the non-strict conditional case are obtained. Since the non-strict paradigm encompasses both the partial and the total one, we can show that the well known results about initiality of those cases, including recent results on non-positive partial conditional specifications, are reobtained by rather simple specializations.

Note that in our framework we are not dealing with error handling. It seems to us that error handling is in a sense an orthogonal problem. We think that it should be interesting and possible to include in an overall non-strict framework the treatment of error handling presented by Poigné in [P].

Problems connected with non-strictness have been addressed by Broy and Wirsing in [BW2] in the context of generalized algebras, where total algebras are enriched by definedness predicates and special morphisms. However, since the aim of that paper is much broader as its title indicates, the treatment there of non-strictness is rather indirect and the issues of initiality and of the relationship with total algebras are not addressed either. Some stimulating considerations about non-strictness, but in a context with different aims, can also be found in [Br].

We have omitted the proofs, which can be found in the full version of the paper [AC3].

1 Non-Strict Algebras

1.1 Partial products and non-strictness

In the following we will deal with *partial objects*, ie with meta-terms not necessarily denoting a concrete element. Thus, as usual in partial frames, two kinds of equalities may be defined between them: the *existential* equality, which holds iff both sides are defined and equal, and the *strong* equality, which holds iff either both sides are undefined or both sides are defined and equal. We will denote the *existential* equality by $=_e$ and the *strong* equality by $=$.

The basic idea we start from is the one of *partial product*.
Usually the product $A_1\times\ldots\times A_n$ is the set of all (total) functions g from $\{1\ldots n\}$ into $A_1\cup\ldots\cup A_n$ s.t. $g(i)\in A_i$. We generalize this concept by allowing *partial functions*. It is worth to note that, while the usual product coincides with the categorical product in the category of sets with total functions as arrows, the *partial* product is not the categorical product in the category of sets with *partial* functions as arrows, because the uniqueness of the factorization through the partial product fails.
In order to keep the notation as similar as possible to the usual one, we use the symbol ? to denote the "undefined" elements.

Def. 1.1.1. Let $A_1,\ldots,A_n$ be sets. The *partial product* of $A_1,\ldots,A_n$, denoted by $\times_p\{A_1,\ldots,A_n\}$, consists of all partial functions from $\{1,\ldots,n\}$ into $A_1\cup\ldots\cup A_n$ s.t. if $g(i)$ is defined, then $g(i)\in A_i$. In order to keep the notation as similar as possible to the usual one, instead of $\times_p\{A_1,\ldots,A_n\}$ we use the infix notation $A_1\times_p\ldots\times_p A_n$, if $n\geq 2$.
Over $\times_p\{A_1,\ldots,A_n\}$ is naturally defined a partial order $\leq$ by
$\underline{a}\leq\underline{b}$ iff $\underline{a}(i)\in A_i$ implies $\underline{b}(i)=_e\underline{a}(i)$ for all $i=1\ldots n$.
A partial function g from $\times_p\{A_1,\ldots,A_n\}$ into a set A is called *strict* iff $g(\underline{a})\in A$ implies $\underline{a}(i)\in A_i$ for all $i=1,\ldots,n$ and is called *monotonic* if $\underline{a}\leq\underline{b}$ and $g(\underline{a})\in A$ implies $g(\underline{a})=_e g(\underline{b})$.
In the following we often denote an element $\underline{a}\in\times_p\{A_1,\ldots,A_n\}$ by $(a_1,\ldots,a_n)$, where $a_i=\underline{a}(i)$ if $\underline{a}(i)\in A_i$ and $a_i=?$ otherwise.

Note that in the particular case $n=1$, usually A and the unary product of A are isomorphic; on the contrary A and $\times_p\{A\}$ are not in general isomorphic: for example, if A has finite cardinality k, then $\times_p\{A\}$ has cardinality $k+1$.

Let us recall the definition of signature, just in order to fix the notation and then define the non-strict algebras, where function symbols of arity $(s_1\ldots s_n,s)$ are interpreted by partial monotonic functions from $s_1{}^A\times_p\ldots\times_p s_n{}^A$ into s^A.

Def. 1.1.2. A *signature* consists of a set S of *sorts* and of a family $F=\{F_{w,s}\}_{w\in S^*,s\in S}$ of sets of *function symbols*. If $f\in F_{w,s}$, then (w,s) is called the *arity* of f.
We will denote a generic signature by Σ, and use $f\in F$ or $f: w\rightarrow s$ instead of $f\in F_{w,s}$ if no ambiguity arises.

Let $\Sigma=(S,F)$ be a signature; a *non-strict* Σ*-algebra* consists of a family $\{s^A\}_{s\in S}$ of sets, the *carriers*, and of a family $\{f^A\}_{f\in F_{w,s},w\in S^*,s\in S}$ of partial functions, the *interpretations of operation symbols*, s.t. if $f\in F_{\Lambda,s}$, then either f^A is undefined or $f^A\in s^A$, otherwise $f\in F_{s_1\ldots s_n,s}$ with $n\geq 1$ and

f^A: $s_1{}^A \times_p \ldots \times_p s_n{}^A \to s^A$ is a monotonic function.
Often we will denote the non-strict algebra A by the couple $(\{s^A\},\{f^A\})$, omitting the quantifications about s and f which are associated with the signature.

An algebra A is called *strict* if f^A is strict for any $f \in F$; moreover a strict algebra is called *total* if $\underline{a}(i) \in s_i{}^A$ for all i=1,…,n implies $f^A(\underline{a}) \in s^A$ for all $f \in F_{s_1 \ldots s_n,s}$.

The class of all non-strict Σ-algebras will be denoted by NSAlg(Σ).

Then, by definition, *strict algebras* are exactly the partial algebras and *total algebras* are the usual ones.

Note that in non-strict algebras no extra-elements are in the carriers to define non-strict functions. For example we can define the boolean algebras with non-strict functions ∧ and ∨ as follows.
Example 1
sort bool; functions ¬: bool → bool; ∧, ∨: bool,bool → bool.
The algebra $\mathbb{B}$ consists of:
$bool^{\mathbb{B}} = \{t,f\}$
$\neg^{\mathbb{B}}$ is the strict function defined by $\neg^{\mathbb{B}}(t) = f$ and $\neg^{\mathbb{B}}(f) = t$;
$\wedge^{\mathbb{B}}$ is defined by
if $\underline{b}(1) = t$ then $\wedge^{\mathbb{B}}(\underline{b}) = \underline{b}(2)$,
if either $\underline{b}(1) = f$ or $\underline{b}(2) = f$ then $\wedge^{\mathbb{B}}(\underline{b}) = f$,
if $\underline{b}(2) = t$ then $\wedge^{\mathbb{B}}(\underline{b}) = \underline{b}(1)$,
otherwise $\wedge^{\mathbb{B}}(\underline{b})$ is undefined;
$\vee^{\mathbb{B}}$ is defined by
if $\underline{b}(1) = f$ then $\vee^{\mathbb{B}}(\underline{b}) = \underline{b}(2)$,
if either $\underline{b}(1) = t$ or $\underline{b}(2) = t$ then $\vee^{\mathbb{B}}(\underline{b}) = t$,
if $\underline{b}(2) = f$ then $\vee^{\mathbb{B}}(\underline{b}) = \underline{b}(1)$,
otherwise $\vee^{\mathbb{B}}(\underline{b})$ is undefined.

Depending on the partiality of the functions, there are several possibilities to define homomorphisms, each one of them is useful for a different purpose (see eg. [BW1, B, R]), in a way that the algebraic structure is preserved. Our choice follows the tradition of partial algebras (see eg. [B, BW1, T, AC1]), where they are used in order to get a no-junk&no-confusion initial object (see [MG]).

Def. 1.1.3. Let $\Sigma = (S,F)$ be a signature, A and B be non-strict algebras over Σ.
Then a *homomorphism* $h: A \to B$ is a family $\{h_s: s^A \to s^B\}_{s \in S}$ of total functions s.t.
$f^A(\underline{a}) \in s^A$ implies $h_s(f^A(\underline{a})) = f^B(h \cdot \underline{a})$, where $h \cdot \underline{a}$ is defined by $h \cdot \underline{a}(i) = h_{s_i}(\underline{a}(i))$ for i=1,…,n,
for all $f \in F_{s_1 \ldots s_n,s}$ and all $\underline{a} \in s_1{}^A \times_p \ldots \times_p s_n{}^A$.

The category **NSAlg**(Σ) is defined by:

- the objects of **NSAlg**(Σ) are NSAlg(Σ);
- the arrows in **NSAlg**(Σ) are all the homomorphisms;
- composition is done component by component;
- the identity on A is $\{id_{s^A}\}_{s \in S}$.

Note that each homomorphism between strict algebras is a total homomorphism of partial algebras and each homomorphism between total algebras is a usual total homomorphism; thus the category both of total algebras and of partial algebras with total homomorphisms are full sub-categories of **NSAlg(Σ)**. Therefore each result in the non-strict frame applies to the usual ones too.

1.2 Term algebras

Let us introduce the term algebras and state some basic results which will be used in the following sections.

Term algebras are defined like in the total frame, so that the concept of substitution as total homomorphism on $T_\Sigma(X)$ is well defined.

Def. 1.2.1. Let $\Sigma = (S,F)$ be a signature and $X = \{X_s\}_{s\in S}$ be a family of S-sorted variables; the *term sets* $T_\Sigma(X)_s$ on Σ and X are inductively defined by:

- $X_s \subseteq T_\Sigma(X)_s$ for all $s\in S$;
- $F_{\Lambda,s} \subseteq T_\Sigma(X)_s$ for all $s\in S$;
- $f\in F_{s_1\dots s_n,s}$ and $t_i\in T_\Sigma(X)_{s_i}$ for $i=1\dots n$ imply $f(t_1,\dots,t_n)\in T_\Sigma(X)_s$.

If X is the empty set, then $T_\Sigma(X)$ is denoted by T_Σ and its elements are called *closed* or *ground* terms.

For all $f\in F_{s_1\dots s_n,s}$ the function $f^{T_\Sigma(X)}: T_\Sigma(X)_{s_1}\times\dots\times T_\Sigma(X)_{s_n} \to T_\Sigma(X)_s$ is the strict total function defined by $f^{T_\Sigma(X)}(t_1,\dots,t_n) = f(t_1,\dots,t_n)$ for all $t_i\in T_\Sigma(X)_{s_i}$.

The *term algebra* $T_\Sigma(X)$ consists of $(\{T_\Sigma(X)_s\}_{s\in S},\{f^{T_\Sigma(X)} \mid f \in F_{w,s}\}_{w,s\in S^*\times S})$.

Let Y be an S-sorted family of variables. A *substitution* $U: T_\Sigma(X) \to T_\Sigma(Y)$ is a total homomorphism from $T_\Sigma(X)$ into $T_\Sigma(Y)$, ie a family of total functions $\{U_s: T_\Sigma(X)_s \to T_\Sigma(Y)_s\}_{s\in S}$ s.t.

$U_s(f(t_1,\dots,t_n)) = f(U_{s_1}(t_1),\dots,U_{s_n}(t_n))$ $\forall$ $f\in F_{s_1\dots s_n,s}$, $t_i\in T_\Sigma(X)_{s_i}$ $i=1\dots n$.

It is well known that some conditions about the signatures are needed in order to have the unique decomposition of terms in the functional notation defined above, when overloading is allowed. We simply assume that in case of ambiguity another notation for terms has been chosen, which makes unique the decomposition.

The evaluation is defined like in the partial strict frame, but also partial valuations for variables have to be allowed. Valuations being partial functions, it is possible to define in a canonical way an order on them, which has a minimal element: the empty valuation.

Def. 1.2.2. Let $\Sigma = (S,F)$ be a signature, $X = \{X_s\}_{s\in S}$ be a family of S-sorted variables. For all algebras $A\in NSAlg(\Sigma)$ and all *valuations* $V = \{V_s: X_s \to s^A\}_{s\in S}$ for X in A, where V_s is a partial function, the *evaluation* $eval^{A,V}: T_\Sigma(X) \to A$ is inductively defined by:

- $eval^{A,V}(x) = V(x)$ for all $x\in X$;
- $eval^{A,V}(f) = f^A$ for all $f\in F_{\Lambda,s}$;
- $eval^{A,V}(f(t_1,\dots,t_n)) = f^A(\underline{a})$, where $\underline{a}(i) = eval^{A,V}(t_i)$ with $i=1\dots n$, for all $f\in F_{s_1\dots s_n,s}$ and $t_i\in T_\Sigma(X)_{s_i}$ for $i=1,\dots,n$.

For all valuations $V, V': X \to A$

$V \leq V'$ iff $V(x) \in s^A$ implies $V(x) =_e V'(x)$ for all $x \in X$.

The valuation $V_?$ for X in A is the empty map, ie $V_?(x)$ is undefined for all $x \in X_s$ and all $s \in S$.

In the following we will denote $eval^{A,V}(t)$ by $t^{A,V}$; moreover if X is the empty set (so that there exists a unique valuation V for X in A), we will denote $eval^{A,V}$ simply by $eval^A$ and $eval^{A,V}(t)$ by t^A. Finally we denote $\underline{a}$, defined by $\underline{a}(i) = t_i^{A,V}$ for $i=1,\ldots,n$, by $(t_1^{A,V},\ldots,t_n^{A,V})$.

It is worth to note that the order on the valuations is preserved by the evaluation, ie $V \leq V'$ implies $eval^{A,V} \leq eval^{A,V'}$.

Prop. 1.2.3. Let $\Sigma = (S,F)$ be a signature, A a non-strict algebra over Σ, X an S-sorted family of variables, V and V' valuations for X in A s.t. $V \leq V'$. For all terms $t \in T_\Sigma(X)_s$ if $t^{A,V} \in s^A$ then $t^{A,V} =_e t^{A,V'}$.

In the total frame, the term-algebras are the free objects in the class of all total algebras, because of the uniqueness of the evaluation w.r.t. a valuation. Here, as in the partial case, term-algebras are not free, because the evaluations are not homomorphisms, not being total. However a derived property holds also in this case, although a bit relaxed. Indeed in the total frame the freedom of the term algebra implies that if the upper triangle of the diagram 1 commutes, then the triangle below commutes too. Analogously in our frame if $V' \leq h \cdot V$, then $eval^{A',V'} \leq h \cdot eval^{A,V}$, so that we have the diagram 2.

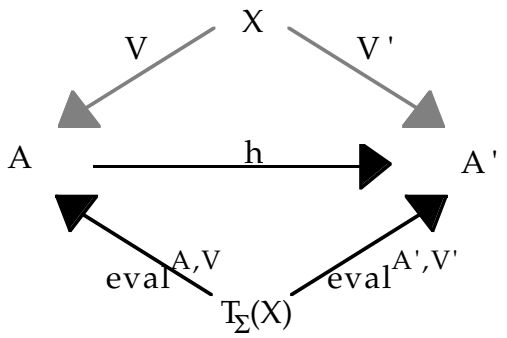


Diagram 1

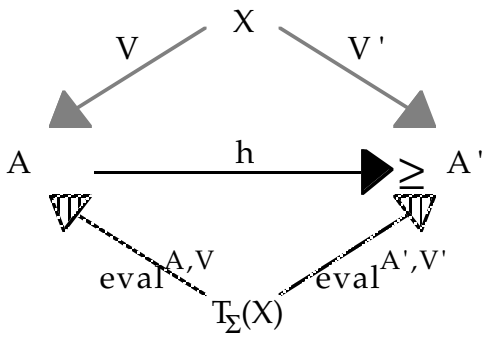


Diagram 2

This result also holds in the partial frame and is crucial in order to get that the initial object in a class, if any, satisfies the *no-junk* and *no-confusion* properties (see [MG]). Indeed if we consider the diagram 2 where A is the initial object and h the unique homomorphism from the initial object into A', then we have that each term defined in A has to be defined in A' (no-junk) and that two terms existentially equal in A have to be existentially equal in A' (no-confusion).

Prop. 1.2.4. Let $\Sigma = (S,F)$ be a signature and $X = \{X_s\}_{s \in S}$ be a family of S-sorted variables. For all non-strict algebras $A,A' \in NSAlg(\Sigma)$, all valuations V for X in A and V' for X in A' and all homomorphisms $h: A \to A'$ s.t. $V' \leq h \cdot V$ we have that

1 $t^{A,V} \in s^A$ implies $h(t^{A,V}) =_e t^{A',V'}$ for all $t \in T_\Sigma(X)$;

2 $t^{A,V} =_e t'^{A,V}$ implies $t^{A',V'} =_e t'^{A',V'}$ for all $t,t' \in T_\Sigma(X)$.

1.3 Inductive and initial algebras.

We first introduce the concept of *inductive* algebra, ie algebra satisfying the no-junk condition, and relate it with the idea of *term-generated*. Then we show that in every class of algebras closed w.r.t. inductive subalgebras the initial object, if any, is characterized by the no-junk and no-confusion properties.

Def. 1.3.1. Let A be a non-strict algebra; its *inductive part* $<A>$ is a family $\{s^{<A>}\}_{s\in S}$ of its carrier sub-sets inductively defined by:

$$\frac{f^A \in s^A}{f^A \in s^{<A>}} \qquad f\in F_{\Lambda,s}$$

$$\frac{\underline{a}\in s_1^{<A>}\times_p\ldots\times_p s_n^{<A>},\ f^A(\underline{a})\in s^A}{f^A(\underline{a})\in s^{<A>}} \qquad f\in F_{s_1\ldots s_n,s}$$

The *inductive subalgebra* B of A consists of:

- $s^B = s^{<A>}$ for all $s\in S$;
- $f^B(\underline{b}) = f^A(\underline{b})$ for all $f\in F_{s_1\ldots s_n,s}$ and all $\underline{b}\in \times_P \{s_1^{<A>},\ldots,s_n^{<A>}\}$.

In the following we will denote by $<A>$ the inductive subalgebra of an algebra A.

The *merge* of $<A>$ into A is the homomorphism $m = \{m_s\}_{s\in S}$ defined by $m_s(a) =_e a$ for all $a\in s^{<A>}$.

A non-strict algebra A is *inductive* iff $A = <A>$.

Let C be a class of non-strict algebras on a signature Σ; the subclass $\mathrm{Ind}(C)$ of C consists of $\{A \mid A\in C, A \text{ is inductive}\}$.

Note that the definition of $s^{<A>}$ guarantees both the well definedness of $f^{<A>}$ and the merge being a homomorphism.

The usual equivalence between inductive and term-generated algebras has to be a little relaxed, because functions over terms are total, while in inductive algebras may also be non-strict. Thus some syntactic elements are needed to play the role of the "undefined" elements which cooperate to build the carriers.

Prop. 1.3.2. Let $\Sigma = (S,F)$ be a signature, X be any family $\{X_s\}_{s\in S}$ of variables s.t. $|X_s| \geq 1$ for all $s\in S$ and A a non-strict algebra.
Then A is inductive iff $eval^{A,V?}: T_\Sigma(X) \to A$ is surjective.

Prop. 1.3.3. Let A and B be non-strict algebras, $h: A \to B$ be a homomorphism; if A is inductive then h is unique.

Def. 1.3.4. Let Σ be a signature and C be a class of non-strict algebras over Σ. An algebra $I\in C$ is *initial* in C iff for each $A\in C$ there exists exactly one homomorphism from I into A.

Prop. 1.3.5. Let $\Sigma = (S,F)$ be a signature, X any family $\{X_s\}_{s\in S}$ of variables s.t. $|X_s| \geq 1$ for all $s\in S$ and C be a class of non-strict algebras over Σ s.t. $A\in C$ implies $<A>\in C$.
A non-strict algebra $I\in C$ is initial in C iff it satisfies the following two conditions

1 I is inductive;
2 $t^{I,V?} =_e t'^{I,V?}$ implies $t^{A,V?} =_e t'^{A,V?}$ for all $A\in C$ and all $t,t'\in T_\Sigma(X)$.

Moreover I is initial in C iff it is initial in $\mathrm{Ind}(C)$.

Prop. 1.3.5 suggests a syntactical characterization of the initial model, very close to the ones, for example, of the total and partial cases. In order to exploit the similarity between this case and the usual ones we start with a definition *ad hoc* of congruence and of quotient.
Since the use of these notions is limited to the study of the existence of the initial model, we restrict the definition just to the case of the term algebra; there are different ways of generalizing these notions to a generic algebra, but none of them corresponds to the quotient in the categorical sense; thus we prefer to avoid this kind of useless generality.

From now on let X denote a family $X = \{X_s\}_{s\in S}$ of variables s.t. $|X_s| \geq 1$.

Def. 1.3.6. Let $\Sigma = (S,F)$ be a signature; a *congruence* $\equiv$ is a family $\equiv = \{\equiv_s\}_{s\in S}$ s.t.

0 $\equiv_s \subseteq T_\Sigma(X)_s \times T_\Sigma(X)_s$ for all $s\in S$; if $(a,b)\in \equiv_s$, then we write $a \equiv_s b$;
1 $\equiv_s$ is symmetric and transitive, ie $t \equiv_s t'$ implies $t \equiv_s t'$ and $t \equiv_s t'$, $t' \equiv_s t''$ imply $t \equiv_s t''$ for all $t,t',t''\in T_\Sigma(X)$.

Let us denote by $Dom(\equiv_s)$ the set $\{t \mid t \equiv_s t\}$ and define $t \equiv^D_s t'$ iff either $t \equiv_s t'$ or $t,t'\notin Dom(\equiv_s)$

2 $t_i \equiv^D_{s_i} t'_i$ for $i=1\ldots n$ and $f\in F_{s_1\ldots s_n,s}$ imply $f(t_1,\ldots,t_n) \equiv^D_s f(t'_1,\ldots,t'_n)$;
3 $f(t_1,\ldots,t_{i-1},x,t_{i+1},\ldots,t_n)\in Dom(\equiv)$ and $x\in X_{s_i}$ imply
$f(t_1,\ldots,t_{i-1},x,t_{i+1},\ldots,t_n) \equiv_s f(t_1,\ldots,t_{i-1},t,t_{i+1},\ldots,t_n)$ for all $t\in T_\Sigma(X)_s$;
4 $f(t_1,\ldots,t_n)\in Dom(\equiv)$, $t_i\notin Dom(\equiv_{s_i})$ imply
$f(t_1,\ldots,t_{i-1},x,t_{i+1},\ldots,t_n) \equiv_s f(t_1,\ldots,t_{i-1},t,t_{i+1},\ldots,t_n)$ for all $t\in T_\Sigma(X)_s$ and all $x\in X_{s_i}$;
5 $x\notin Dom(\equiv_s)$ for all $x\in X_s$.

The *quotient* $T_\Sigma(X)/\equiv$ is the non-strict algebra defined by:

- $s^{T_\Sigma(X)/\equiv}$ is $T_\Sigma(X)/\equiv_s$ for all $s\in S$; in the following we denote the class of t in $s^{T_\Sigma(X)/\equiv}$ by $[t]$;
- $f^{T_\Sigma(X)/\equiv}(\underline{t}) = [f(t_1,\ldots,t_n)]$, where if $\underline{t}(i)\in s^{T_\Sigma(X)/\equiv}$ then $t_i\in \underline{t}(i)$ else $t_i\in X_{s_i}$ for all $f\in F_{s_1\ldots s_n,s}$.

For any non-strict algebra A let $\equiv_A$ be the congruence defined by $t \equiv_A t'$ iff $t^{A,V?} =_e t'^{A,V?}$.

Note that $f^{T_\Sigma(X)/\equiv}$ is well defined. Indeed let t_i, t'_i belong to $\underline{t}(i)$ for all i s.t. $\underline{t}(i)\in s^{T_\Sigma(X)/\equiv}$, otherwise t_i, t'_i belong to X_{s_i} and hence $t_i,t'_i\notin Dom(\equiv_{s_i})$ because of 5; then $t_i \equiv^D t'_i$ for $i=1\ldots n$ and hence, because of 2, $f(t_1,\ldots,t_n) \equiv^D f(t'_1,\ldots,t'_n)$ so that $[f(t_1,\ldots,t_n)] = [f(t_1,\ldots,t_n)]$.

As in more familiar frames, also in this case the evaluation of a term is the equivalence class of the terms where variables have been replaced by their valuation.

Def. 1.3.7. Let $\Sigma = (S,F)$ be a signature and D be a non-empty set of non-strict algebras over Σ. The *product* $\prod_{A\in D} A$ is the non-strict algebra over Σ defined by:
for all $s\in S$ let $s^{\prod_{A\in D} A}$ be $\prod_{A\in D} s^A = \{g: D \to \cup_{A\in D} s^A \mid g(A)\in s^A \ \forall\ A\in D\}$;
for all $f\in F_{s_1\ldots s_n,s}$ let $f^{\prod_{A\in D} A}$ be the function defined by:

for all $\underline{p}\in \times_P \{s_1^{\prod_{A\in D} A},\ldots,s_n^{\prod_{A\in D} A}\}$ $f^{\prod_{A\in D} A}(\underline{p})$ is defined iff $f^A(\underline{a})$ is defined for all $A\in D$, where $\underline{a}$ is defined by $\underline{a}(i) = \underline{p}(i)(A)$ for $i=1\ldots n$, and in this case $f^{\prod_{A\in D} A}(\underline{p})$ is defined by $f^{\prod_{A\in D} A}(\underline{p})(A) = f^A(\underline{a}(A))$ for all $A\in D$.

The *projection* of $\prod_{A\in D} A$ into A, denoted by π_A, is the homomorphism defined by:
$\pi_A(g) = g(A)$ for all $g\in s^{\prod_{A\in D} A}$ and all $s\in S$.
In the following if D is the set $\{A_1,\ldots,A_n\}$, then we denote $\prod_{A\in D} A$ also by $A_1 \times\ldots\times A_n$.

Let C be a non-empty class of non-strict algebras over Σ and D be the set of non-strict algebras defined by $D = \{T_\Sigma(X)/\equiv_A \mid A \in D\}$. Then $I(C)$ denotes the inductive sub-algebra of the product $\prod_{B \in D} B$.

Theorem 1.3.8. Let $\Sigma = (S,F)$ be a signature and C be a class of non-strict algebras over Σ closed under isomorphisms and s.t. $A \in C$ implies $<A> \in C$.

Then there exists an initial object in C iff there exists an initial object in $Ind(C)$ iff $I(C)$ belongs to C iff $I(C)$ is initial in C.

2 Non-Strict Specifications

Usually in both the total and the partial frame, logical formulas (equations and positive Horn clauses) are considered s.t. their model classes are closed w.r.t. non-empty products and sub-objects, so that the model classes satisfy *a fortiori* the closure w.r.t. $I(C)$ which is necessary and sufficient for the existence of an initial model, by theorem 1.3.8, for classes closed under subobjects and isomorphisms. In the non-strict frame, the same way cannot be followed, because there are finite sets of equations whose model classes are neither closed w.r.t. $I(C)$, nor w.r.t. non-empty products. Let us informally show this claim by a simple example.

Example T_2.

Sort s Operations $a, b: \rightarrow s$ $f: s \rightarrow s$ Axiom $f(a) =_e b$.

The following two algebras are obviously models of T_2:

A $s^A = \{\bullet\}$ $a^A = b^A = \bullet$; f^A is the total strict function defined by $f^A(\bullet) = \bullet$;

B $s^B = \{\bullet\}$ a^B is undefined; $b^B = \bullet$; $f^B(\underline{b}) = \bullet$ for all $\underline{b} \in \times_P \{s^B\}$.

Let C be the model class of T_2. By definition of product, both $a^{A\times B}$ and $a^{I(C)}$ are undefined, because a^B is undefined, and analogously both $f^{A\times B}(g)$ and $f^{I(C)}(g)$, where g is the totally undefined function, are undefined too, because f^A is strict. Therefore both $f(a)^{A\times B}$ and $f(a)^{I(C)}$ are undefined and hence both $A\times B$ and $I(C)$ are not models of T_2.

In the example above the problem arises because of the monotonicity of the interpretation of the function symbols; indeed from $f(a) =_e b$ we have that in each model A either $a =_e a$ or $f(x) =_e b$ holds. Thus equations implicitly introduce disjunctions. Moreover, using conditional axioms, it is possible to code each disjunction and hence in the non-strict frame it is equivalent to deal with equations and to deal with disjunctions. Let us support this claim by an informal proof.

Let $\varepsilon_1 \vee \ldots \vee \varepsilon_n \vee \neg\eta_1 \vee \ldots \vee \neg\eta_m$, where $\varepsilon_1,\ldots,\varepsilon_n,\eta_1,\ldots,\eta_m$ are all existential equalities; then $\varepsilon_1 \vee \ldots \vee \varepsilon_n \vee \neg\eta_1 \vee \ldots \vee \neg\eta_m$ may be coded by the set:

α_i $D(f_i(x)) \wedge \eta_1 \wedge \ldots \wedge \eta_m \supset \varepsilon_i$, for $i=1\ldots n$

α $D(f_n(f_{n-1}(\ldots(x)\ldots)$

where $f_1, \ldots, f_n$ are auxiliary unary functions and $D(t)$ stands for $t =_e t$. Indeed each algebra A satisfying α satisfies also at least one $D(f_i(x))$ and hence if A satisfies also α_i either there exists an η_j s.t. A does not satisfy η_j or A satisfies ε_i, so that A satisfies $\varepsilon_1 \vee \ldots \vee \varepsilon_n \vee \neg\eta_1 \vee \ldots \vee \neg\eta_m$. Vice versa if A satisfies $\varepsilon_1 \vee \ldots \vee \varepsilon_n \vee \neg\eta_1 \vee \ldots \vee \neg\eta_m$, then A may generalize to a model of $\alpha_1,\ldots,\alpha_n,\alpha$, suitably defining the interpretation of $f_1,\ldots,f_n$.

Therefore in the following we focus our attention on *disjunctive* types.

2.1 Formulas, Validity and Types

In the following we use formulas within an *infinitary* logic (for reference see eg [K]), with infinitary conjunctions and disjunctions and families of denumerable sets of variables.

Def. 2.1.1. Let $\Sigma = (S,F)$ be a signature and X be a family of S-sorted variables.

- The set $Eq(\Sigma,X)$ of *equalities* on Σ and X consists of $t =_e t'$ for all $t,t' \in T_\Sigma(X)_s$, $s \in S$; the set $At(\Sigma,X)$ of *atomic* formulas on Σ and X is $Eq(\Sigma,X) \cup \{\neg\varepsilon \mid \varepsilon \in Eq(\Sigma,X)\}$.
- The set $Form(\Sigma,X)$ of all *well-formed* formulas is inductively defined by:
  $Eq(\Sigma,X) \subseteq Form(\Sigma,X)$.
  $\Phi \cup \{\vartheta,\vartheta'\} \subseteq Form(\Sigma,X)$ implies $\wedge\Phi, \vee\Phi, \neg\vartheta, \vartheta \supset \vartheta' \in Form(\Sigma,X)$.
- The set $CForm(\Sigma,X)$ of *conditional formulas* on Σ and X is the set $\{\wedge\Delta \supset \varepsilon \mid \Delta \cup \{\varepsilon\} \subseteq Eq(\Sigma,X)\}$. If Δ is the empty set, then $\wedge\Delta \supset \varepsilon$ is an equivalent notation for ε and hence $Eq(\Sigma,X) \subseteq CForm(\Sigma,X)$.
- The set $DForm(\Sigma,X)$ of *disjunctive formulas* on Σ and X is the set $\{\vee\Delta \mid \Delta \subseteq At(\Sigma,X)\}$. If Δ consists of one atomic formulas ε, then $\vee\Delta$ is an equivalent notation for ε and hence $At(\Sigma,X) \subseteq DForm(\Sigma,X)$. ☆

In the following a generic equality will be denoted by ε or η, a generic atomic formula by γ or δ and a generic formula by φ or ϑ or ψ. Moreover we will denote the empty conjunction $\wedge\emptyset$ by True and the empty disjunction $\vee\emptyset$ by False.

Def. 2.1.2. Let $\Sigma = (S,F)$ be a signature, X be a family of S-sorted variables and A be a non-strict Σ-algebra.
If φ is a formula and V is a valuation for $Var(\varphi)$ in A, then we say that φ *holds for* V *in* A (equivalently: *is satisfied for* V *by* A) and write $A \text{ ‘}_V \varphi$ accordingly to the following definitions.

- $A \text{ ‘}_V t =_e t'$ iff $t^{A,V}, t'^{A,V} \in s^A$ and $t^{A,V} = t'^{A,V}$;
- $A \text{ ‘}_V \wedge\Phi$ iff $A \text{ ‘}_V \varphi$ for all $\varphi \in \Phi$;
- $A \text{ ‘}_V \vee\Phi$ iff there exists $\varphi \in \Phi$ s.t. $A \text{ ‘}_V \varphi$.
- $A \text{ ‘}_V \neg\vartheta$ iff $A \text{ ’}_V \vartheta$.
- $A \text{ ‘}_V \vartheta \supset \vartheta'$ iff either $A \text{ ‘}_V \vartheta'$ or $A \text{ ’}_V \vartheta$.

We write $A \text{ ‘ } \varphi$ for a formula φ and say that φ *holds in* (equivalently: *is satisfied by, is valid in*) A iff $A \text{ ‘}_V \varphi$ for all valuations V for $Var(\varphi)$ in A. Let us shortly denote by $D(t)$ the equality $t =_e t$, where both sides are the same term, because $t =_e t$ simply state the definedness of t.

The definition of validity justifies the notations introduced for the empty conjunction and disjunction; indeed $A \text{ ‘}_V \wedge\emptyset$ for all non-strict algebras A and all valuations V, because obviously $A \text{ ‘}_V \varphi$ for all $\varphi \in \emptyset$, so that $\wedge\emptyset$ plays the role of the constant True, and $A \text{ ’}_V \vee\emptyset$ for all non-strict algebras A and all valuations V, because obviously there does not exist $\varphi \in \emptyset$ s.t. $A \text{ ‘}_V \varphi$, so that $\vee\emptyset$ plays the role of the constant False.

Remark. In both the total and the partial frame, since valuations are *total* functions, the relation $\equiv$ on $T_\Sigma(X)$, defined by $t \equiv t'$ iff $A \text{ ‘ } t =_e t'$, may be not an equivalence relation if empty carriers are allowed, because it may be not transitive. Indeed consider the following example, which is a simplified version of a well known example presented in [MG].

Example 3

Let Σ be the signature defined by:

sorts: s_1, s_2 operations: $a,b: \to s_1$ $f: s_2 \to s_1$

Then $T_\Sigma \text{ ‘ } a = f(x)$ and $T_\Sigma \text{ ‘ } f(x) = b$, because $T_{\Sigma s_2} = \varnothing$ and hence there does not exist a (total) valuation for $\{x\}$ in T_Σ, but $T_\Sigma \text{ ’ } a = b$ so that $\equiv$ is not transitive.

This fact has consequences in the case of inference systems, which have to deal very carefully with the elimination of the variables.

On the contrary, in the non-strict frame these problems do not arise, because valuations are *partial* functions, so that there exists at least the totally undefined valuation for all families of variables and all non-strict algebras. For instance, in the above example $T_\Sigma \text{ ’}_{V?} a = f(x)$ and hence $\equiv$ is transitive. Moreover the following prop. 2.1.3 shows that $\equiv$ coincides with the relation $\equiv_?$, defined by $t \equiv_? t'$ iff $A \text{ ‘}_{V?} t =_e t'$, and hence is an equivalence relation for all non-strict Σ-algebra A. Manca and Salibra in [MS] introduced first the partial valuations to solve the empty-carriers problem and maintain the original Birkhoff equational calculus, by changing the concept of validity; however in our frame the introduction of partial valuations has a completely different motivation; indeed it is not a choice and a technical device as in [M], but it arises naturally from the setting, since functions are non-strict and variables have to represent *all* the possible arguments.

Prop. 2.1.3. Let $\Sigma = (S,F)$ be a signature, X be a family of S-sorted variables, Δ be a set of equalities over Σ and X. For any non-strict Σ-algebra A $A \text{ ‘ } \vee \Delta$ iff $A \text{ ‘}_{V?} \vee \Delta$ iff there exists $\delta \in \Delta$ s.t. $A \text{ ‘ } \delta$ and in particular $A \text{ ‘ } t =_e t'$ iff $A \text{ ‘}_{V?} t =_e t'$ for all terms t and t'.

Note that if Δ is a set of atoms, ie of equalities and negated equalities, then $A \text{ ‘}_{V?} \vee \Delta$ does not imply $A \text{ ‘ } \vee \Delta$. For example if Δ is $\{\neg D(x)\}$, then it is satisfied only for the totally undefined valuation, while each non-empty algebra does not satisfy $\vee \Delta$.

Def. 2.1.4. A *type* or *specification* consists of a signature Σ and of a set of well-formed formulas over Σ, called *axioms* of T. If all the axioms of a type are disjunctions, then the type is called *disjunctive*; if all the axioms of a type are conditional formulas, then the type is called *conditional* and if all the axioms of a type are equalities, then the type is called *equational.*

Let $T = (\Sigma,Ax)$ be a type; the class Mod(T) of *models* of T is the set

$\{A \mid A \in NSAlg(\Sigma), A \text{ ‘ } \alpha \ \forall \alpha \in Ax\}$.

A model of T is *initial* for T iff it is initial in Mod(T).

Remark. Note that disjunctive types are sufficient to define any class of models definable using well-formed formulas. Indeed each well-formed formula over the usual logical connectives may be expressed in conjunctive normal form. Since a conjunction of formulas is logically equivalent to the set of the formulas in the conjunction, each well-formed formula is logically equivalent to a set of disjunctive formulas.

Logic in a non-strict frame is quite different from the usual one; for instance Mod(T) may be empty also for equational types. For example if D(x) is an axiom of the specification, then no algebra can satisfy this axiom w.r.t. the valuation completely undefined, so that the specification has no models.

Def. 2.1.5. A type T is *consistent* iff Mod(T) is not empty.

2.2 Relating total and non-strict algebras

Let us now relate the non-strict frame with the more usual total one, first at a naive level and then formalizing the results by the concept of institution [GB1, GB2], simulation of institutions [AC2] and map of institutions [M].

Since the partial product of $s_1^A,\ldots,s_n^A$ is isomorphic, from a set-theoretical point of view, to the (usual) product of $s_1^A\cup\{\perp_{s_1}\},\ldots,s_n^A\cup\{\perp_{s_n}\}$, where the symbol $\cup$ denotes the disjoint union, any non-strict algebra A is in some sense equivalent to the total algebra $A_\perp$, defined by:

$s^{A\perp} = s^A\cup\{\perp_s\}$ for all $s\in S$;
if $f^A(\underline{a})$ is defined, where $\underline{a}(i) = a_i$ if $a_i\in s_i^A$ otherwise $\underline{a}(i)$ is undefined, then $f^{A\perp}(a_1,\ldots,a_n) = f^A(\underline{a})$, else $f^{A\perp}(a_1,\ldots,a_n) = \perp_s$ for all $f: s_1\times\ldots\times s_n \to s$, all $a_i\in s_i^{A\perp}$ for $i=1\ldots n$.

However this equivalence is not categorical, ie there does not exist a functor from non-strict to total algebras sending A into $A_\perp$. Indeed the introduction of *one* element to represent *all* the undefined terms may cause the lack of the existence of homomorphisms. Consider the following example.

Example
Let Σ be the one-sorted signature consisting of just three constant symbols $\{a,b,c\}$ and A, B be the non-strict algebras over Σ, defined by:

A: $s^A = \{1\}$; $a^A = 1$; both b^A and c^A are undefined;
B: $s^B = \{1\}$; $a^B = 1$; $b^B = 1$; c^B is undefined;

then there is a non-strict homomorphism $h: A \to B$, defined by $h(1) = 1$.

Consider now the trivial totalizations of A and B.

$A_\perp$: $s^A = \{1,\perp\}$; $a^{A\perp} = 1$; $b^{A\perp} = \perp$; $c^{A\perp} = \perp$;
$B_\perp$: $s^B = \{1,\perp\}$; $a^{B\perp} = 1$; $b^{B\perp} = 1$; $c^{B\perp} = \perp$;

then there does not exist any total homomorphism from $A_\perp$ into $B_\perp$, because $b^{A\perp} = c^{A\perp}$, while $b^{B\perp} \neq c^{B\perp}$.

Since the undefined elements are *don't care* elements, it is quite unreasonable that the equality between $b^{A\perp}$ and $c^{A\perp}$, which are "undefined", be essential for the non-existence of homomorphisms from $A_\perp$ into $B_\perp$. Thus in order to have a more natural representation of non-strict algebras, we need a definition of (total) homomorphism, which does not involve the "undefined" part. To do this it is useful, not to say necessary, having a tool to individuate the "undefined" elements, for example a family of unary predicates, one for each sort, dividing the carriers in "defined" and "undefined". Following a similar idea both [BW2] and [P] define homomorphisms which are partial functions, having as domain the "defined" part; this approach can be generalized in order to include non-strictness. To fix the notation let us introduce the *total algebras with predicates*, or *first-order structures*.

Def. 2.2.1. A *signature with predicates* consists of a set S of *sorts*, an $S^*\times S$-indexed family F of *operation symbols*, and an S^+-indexed family P of *predicate symbols*.
Let $\Sigma = (S,F,P)$ be a signature with predicates; a *total algebra with predicates* or *first-order structure* consists of a triple $(\{s^A\}_{s\in S},\{f^A\}_{f\in F},\{p^A\}_{p\in P})$, where s^A is an arbitrary set for all $s \in S$, f^A is a

total function from $s_1^A \times \ldots \times s_n^A$ into s^A for all $f \in F_{s_1 \ldots s_n, s}$ and finally $p^A \subseteq s_1^A \times \ldots \times s_n^A$ for all $p \in P_{s_1 \ldots s_n}$.

Let A, B be total algebras with predicates over Σ; a *total homomorphism* is a family $h = \{h_s\}_{s \in S}$ of total functions $h_s: s^A \to s^B$ s.t.

- $h_s(f^A(a_1,\ldots,a_n)) = f^B(h_{s_1}(a_1),\ldots,h_{s_n}(a_n))$ for all $f \in F_{s_1 \ldots s_n, s}$;
- $(a_1,\ldots,a_n) \in p^A$ implies $(h_{s_1}(a_1),\ldots,h_{s_n}(a_n)) \in p^B$ for all $p \in P_{s_1 \ldots s_n}$.

Total algebras with predicates over a signature Σ and total homomorphisms form a category, from now on denoted by **TAlg**(Σ), where composition and identity are composition and identity as maps.

Let $\Sigma = (S,F)$ be a non-strict signature and Σ' denote the following signature with predicates $(S, F \cup \{\bot_s: \to s\}_{s \in S}, \{D_s: s, =_e: s\ s\}_{s \in S})$; let us investigate the relations between **NSAlg**(Σ) and **TAlg**(Σ').

Of course not any total algebra over Σ' represents a non-strict algebra, because either the monotonicity condition may not hold or $\bot$, which has been introduced to represent the "undefined" elements, may be defined, or else $=_e$ may fail to represent the existential equality.

Def. 2.2.2. Let $\Sigma = (S,F)$ be a non-strict signature and Σ' denote the following signature with predicates $(S, F \cup \{\bot_s: \to s\}_{s \in S}, \{D_s: s, =_e: s\ s\}_{s \in S})$; a total algebra A over Σ' is called *don't care* iff

1 $\bot_s \notin D_s^A$ for all $s \in S$;
2 for all $f \in F_{s_1,\ldots,s_n,s}$
$f^A(a_1,\ldots,a_n) \in D_s^A$ and $a_i \notin D_{s_i}^A$ imply $f^A(a_1,\ldots,a_n) = f^A(a_1,\ldots,a_{i-1},a,a_{i+1},\ldots,a_n)\ \forall\ a \in s_i^A$;
3 $(a,a') \in =_e^A$ iff $a,a' \in D_s^A$ and $a = a'$.

Let **DCAlg**(Σ') denote the full subcategory of **TAlg**(Σ') whose objects are all the don't care algebras.

Prop. 2.2.3. Let $\Sigma = (S,F)$ be a non-strict signature and Σ' denote the following signature with predicates $(S, F \cup \{\bot_s: \to s\}_{s \in S}, \{D_s: s, =_e: s\ s\}_{s \in S})$. A total algebra with predicates A over Σ' is don't care iff it satisfies the following set $Ax_{DC}(\Sigma)$ of axioms.

- $\neg D_s(\bot_s)$ for all $s \in S$;
- $t =_e t' \Leftrightarrow D(t) \wedge D(t') \wedge t = t'$,
  ie $\neg D(t) \vee \neg D(t') \vee \neg t = t' \vee t =_e t'$, $\neg t =_e t' \vee D(t)$, $\neg t =_e t' \vee D(t')$, $\neg t =_e t' \vee t = t'$;
- $D_s(f(x_1,\ldots,x_n)) \wedge \neg D_{s_i}(x_i) \supset f(x_1,\ldots,x_n) =_e f(x_1,\ldots,x_{i-1},y,x_{i+1},\ldots,x_n)$,
  ie $\neg D_s(f(x_1,\ldots,x_n)) \vee D_{s_i}(x_i) \vee f(x_1,\ldots,x_n) =_e f(x_1,\ldots,x_{i-1},y,x_{i+1},\ldots,x_n)$.

Remark. There does not exist a total conditional specification whose model class is Obj(**DCAlg**(Σ')). Indeed the trivial total algebra Tr over Σ', having singleton sets as carriers, the unique obvious interpretation of function symbols and the totally true predicates, ie $D_s^{Tr} = s^{Tr}$ and $=_e^{Tr} = s^{Tr} \times s^{Tr}$, is a model of each conditional type while it is not a don't care algebra. Therefore in general it is impossible to translate a conditional (equational) non-strict specification into a conditional total one, because it is impossible at least for the specification without axioms $(\Sigma,\varnothing)$. Thus the relation we are investigating is

between the categories of total and non-strict *algebras* and cannot be lifted to a relation between *conditional specifications*. However using disjunctive formulas it is possible to express the don't care conditions and hence non-strict conditional specifications are translated into disjunctive total specifications. The deep sense of this translation and its formalization involve the concepts of simulations and of map of institutions so that they are discussed at the end of this section.

It is easy to define a *forgetful* functor F from **DCAlg**(Σ') into **NSAlg**(Σ).

Prop. 2.2.4. Let $\Sigma = (S,F)$ be a non-strict signature and Σ' denote the following signature with predicates $(S,F\cup\{\bot_s: \to s\}_{s\in S},\{D_s: s, =_e: s\ s\}_{s\in S})$. Then F, defined below, is a functor from **DCAlg**(Σ') into **NSAlg**(Σ).

For any object A in **DCAlg**(Σ'), $F(A)$ is the non-strict algebra defined by:

- $s^{F(A)} = D_s{}^A$ for all $s\in S$;
- for any $\underline{a}\in s_1{}^{F(A)}\times_P\ldots\times_P s_n{}^{F(A)}$ if $\underline{a}(i)\in D_{s_i}{}^A$ then let a_i be $\underline{a}(i)$, else let a_i be $\bot_{s_i}$;
  for all $f\in F_{s_1,\ldots,s_n,s}$ then define $f^{F(A)}(\underline{a}) = f^A(a_1,\ldots,a_n)$ if $f^A(a_1,\ldots,a_n)\in D_s{}^A$, undefined otherwise.

For any total homomorphism $h = \{h_s\}_{s\in S}: A \to B$, $F(h): F(A) \to F(B)$ is the family $\{h_{s|D_s{}^A}\}_{s\in S}$.

Thus in general many different total algebras with predicates represent the same non-strict algebra, because they differ on the undefined part. Now we show that there exists a privileged representation, given by the left adjoint of the forgetful functor, from now on denoted by *Tot*, corresponding, as usual, to a free construction. *Tot*(A) is a privileged representation of A in the sense that each non-strict homomorphism $h: A \to B$ may be generalized to a total homomorphism *Tot*(h): *Tot*(A) $\to$ *Tot*(B). Thus the lack of existence of homomorphisms seen in the case of the trivial totalization does not apply to this *clever* totalization. However *Tot*(A), being a free construction, has as many "undefined" elements as possible and hence its carriers are full of junk. To build such *Tot* we need some preliminary technical results.

Prop. 2.2.5. Let $\Sigma = (S,F)$ be a non-strict signature and Σ' denote the following signature with predicates $(S,F\cup\{\bot_s: \to s\}_{s\in S},\{D_s: s, =_e: s\ s\}_{s\in S})$; let A be a non-strict algebra over Σ, X_A be the S-sorted family defined by $X_s = s^A \cup \{\bot\}$ for all $s\in S$ and $V_A: X_A \to A$ be the valuation defined by $V_A(a) = a$ if $a\in s^A$, $V_A(\bot)$ undefined.

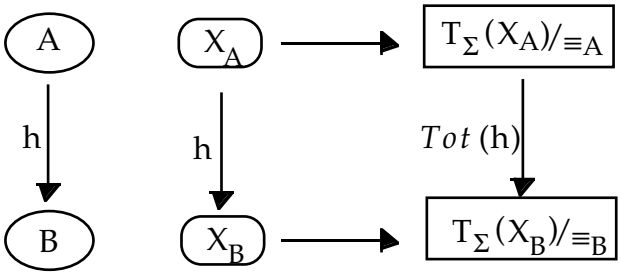
Let $\equiv_A$ denote the total congruence over $T_\Sigma(X_A)$ generated by:

$\{(t,t') \mid t\in T_\Sigma(X_A),\ A\ `_{V_A}\ t =_e t'\}$.

Then the algebra $Tot(A) = (T_\Sigma(X_A)/\equiv_A, \{\bot_s{}^{Tot(A)}\}_{s\in S}, \{D_s{}^{Tot(A)}, =_e{}^{Tot(A)}\}_{s\in S})$, where $\bot_s{}^{Tot(A)} = [\bot]_{\equiv_A}$, $D_s{}^{Tot(A)} = \{[t]_{\equiv_A} \mid t^{A,V_A}\in s^A\}$ and $([t]_{\equiv_A},[t']_{\equiv_A})\in =_e{}^{Tot(A)}$ iff $A\ `_{V_A} t =_e t'$, belongs to **DCAlg**(Σ').

Prop. 2.2.6. Using the notation of prop. 2.2.5, $F(Tot(A))$ is isomorphic to A.

Let us consider a non-strict homomorphism $h: A \to B$. In order to define its image along *Tot*, we use h as a valuation from X_A into X_B and then show that $t \equiv_A t'$ implies $h(t) \equiv_B h(t')$, so that $Tot(h)([t]_{\equiv_A}) = [h(t)]_{\equiv_B}$ is well defined.



Prop. 2.2.7. Let h: A → B be a non-strict homomorphism; using the notation of prop. 2.2.5, *Tot*(h): *Tot*(A) → *Tot*(B), defined by *Tot*(h)($[t]_{\equiv_A}$) = $[h(t)]_{\equiv_B}$, is a homomorphism of total algebras with predicates.

And finally we can put together prop. 2.2.5 and prop. 2.2.7 to define the functor *Tot*.

Theorem 2.2.8. Let Σ = (S,F) be a non-strict signature and Σ' denote the following signature with predicates $(S,F\cup\{\perp_s: \rightarrow s\}_{s\in S},\{D_s: s, =_e: s\ s\}_{s\in S})$. Using the notation of prop. 2.2.5 and prop. 2.2.7, *Tot* is a functor.

Let us now summarize our previous investigation about the relations between total and non-strict algebras.

Each non-strict algebra may be represented by a total algebra, where a special element ⊥ has been added to the carriers to denote the "undefined" elements, but this trivial totalization cannot be lifted to a categorical correspondence, since it cannot be expressed by a functor.

For a right categorical translation of non-strict into total algebras we need some more algebraic tools, in particular definedness predicates, and get that each non-strict algebra may be represented by many different first-order structures (total algebras with predicates) we can think of as its *implementations*, satisfying the same formulas. However the class of all first-order structures (total algebras with predicates) representing some non-strict algebras cannot be described by conditional axioms, so that there is not a correspondence between (equational) conditional non-strict specifications and conditional total specifications.

There exists a disjunctive specification having as model class the class of all don't care total algebras with predicates; thus each disjunctive non-strict specifications is represented by a disjunctive total specifications.

Algebras	**Categories**	**Specifications**
Trivial totalization is sufficient	First-order structures (Algebras with predicates); sentences remain the same	First-order structures (Algebras with predicates) and disjunctive formulas

2.3 Disjunctive types and initiality

Although the class of models of a disjunctive type is not a variety, because it is not closed under products nor quotients, it is at least closed under inductive subobjects and isomorphisms and these closures are sufficient to instantiate prop. 1.3.5 and theorem 1.3.8.

Prop. 2.3.1. The class of models of a disjunctive type is closed w.r.t. inductive subobjects and isomorphisms.

Recall the definition of I(*C*), for *C* class of non-strict Σ-algebras, given in def. 1.3.7.

Theorem 2.3.2. Let $\Sigma = (S,F)$ be a signature and $T = (\Sigma,Ax)$ be a disjunctive type.
A non-strict algebra $I \in Mod(T)$ is initial in $Mod(T)$ iff it is isomorphic to $I(Mod(T))$ iff it satisfies the following two conditions

1 I is inductive;
2 $I \text{ ‘ } t =_e t'$ implies $A \text{ ‘ } t =_e t'$ for all $A \in Mod(T)$ and all $t,t' \in T_\Sigma(X)$, where X is an S-sorted family of variables s.t. $|X_s| \geq 1$ for all $s \in S$.

Moreover I is initial in *C* iff it is initial in Ind(Mod(T)).

Theorem 2.3.3. Let $\Sigma = (S,F)$ be a signature and $T = (\Sigma,Ax)$ be a disjunctive type. The following conditions are equivalent:

1 there exists an initial model in Mod(T);
2 there exists an initial model in Ind(Mod(T));
3 $I(Mod(T)) \in Mod(T)$;
4 I(Mod(T)) is initial in Mod(T).

In general the initial model may not exist even for equational consistent types. Consider the following example.

Example T_4.
Sort s Operations f, f': $s \to s$ Axiom $D(f'(f(x)))$.
Then T_4 is a consistent equational type, because the non-strict algebra A, defined by $s^A = \{\bullet\}$ and $f^A(\underline{a}) = f'^A(\underline{a}) = \bullet$ for all $\underline{a} \in \times_P \{s^A\}$, is a model of T_4.
Now we show that there are two models A and B of T_4 s.t. respectively $f^A(?) \notin s^A$ and $f'^B(?) \notin s^B$ and hence for any algebra I satisfying the condition 2 of prop. 1.3.5 $f^I(f^I(?)) \notin s^I$ so that it is not a model of T_4; thus T_4 has no initial model, because of prop. 1.3.5.

A	$s^A = \{\bullet\}$	$f^A(\bullet)$ and $f^A(?)$ undefined	$f'^A(?) = f'^A(\bullet) = \bullet$
B	$s^B = \{\bullet\}$	$f^B(?) = f^B(\bullet) = \bullet$	$f'^B(\bullet)$ and $f'^B(?)$ undefined.

And finally we can give our first main result: necessary and sufficient conditions for the existence of the initial model.

Theorem 2.3.4. Let $T = (\Sigma,Ax)$ be a consistent disjunctive type. The following conditions are equivalent:

1 $I(Mod(T)) \in Mod(T)$.
2 There exists I initial in Ind(Mod(T)).
3 There exists I initial in Mod(T).
4 $[A \text{ ‘ } \vee \Delta$ for all $A \in Mod(T)]$ implies $[\exists\, \delta \in \Delta$ s.t. $A \text{ ‘ } \delta$ for all $A \in Mod(T)]$ for all sets Δ of equalities.
5 a for all $f \in F_{s_1 \dots s_n,s}$ $[A \text{ ‘ } D(f(t_1,\dots,t_n))$ for all $A \in Mod(T)$ and $\exists\, B \in Mod(T)$ s.t. $B \text{ ’ } D(t_i)]$ imply $[C \text{ ‘ } f(t_1,\dots,t_n) =_e f(t_1,\dots,t_{i-1},x,t_{i+1},\dots,t_n)$ for all $C \in Mod(T)]$;
b for all $\vee \Delta \in Ax$ and all substitutions $U: T_\Sigma(Var(\vee \Delta)) \to T_\Sigma(X)$
$[A \text{ ‘ } t =_e t'$ for all $A \in Mod(T)$ and all $\neg t =_e t' \in U(\Delta)]$ implies $[\exists\, \delta \in U(\Delta) \cap EEq(\Sigma,X)$ s.t. $A \text{ ‘ } \delta$ for all $A \in Mod(T)]$.

6 The relation $\equiv$ over $T_\Sigma(X)$, defined by $t \equiv t'$ iff $A \text{ ‘ } t =_e t'$ for all $A \in Mod(T)$, is a congruence and $T_\Sigma(X)/\equiv$ is the initial model.

2.4 Non-strict conditional types

The conditional types are a particular case of disjunctive types; indeed the conditional formula $\wedge \Delta \supset \varepsilon$ is logically equivalent to $\vee \{\neg\delta \mid \delta \in \Delta\} \cup \{\varepsilon\}$; in other words conditional formulas are disjunctions where exactly one non-negated equality appear, ie are positive Horn clauses. In this case the necessary and sufficient conditions for the existence of an initial object are partially simplified.

Theorem 2.4.1. Let $T = (\Sigma, Ax)$ be a consistent conditional type. The following conditions are equivalent:

1 The inductive sub-algebra of the product $\prod_{A \in Mod(T)} A$ belongs to $Mod(T)$.
2 There exists I initial in $Ind(Mod(T))$.
3 There exists I initial in $Mod(T)$.
4 $[A \text{ ‘ } \vee \Delta \text{ for all } A \in Mod(T)]$ implies $[\exists\, \delta \in \Delta \text{ s.t. } A \text{ ‘ } \delta \text{ for all } A \in Mod(T)]$ for all sets Δ of equalities.
5 $[A \text{ ‘ } D(f(t_1,\ldots,t_n)) \text{ for all } A \in Mod(T) \text{ and } \exists\, B \in Mod(T) \text{ s.t. } B \text{ ’ } D(t_i)]$ imply $[C \text{ ‘ } f(t_1,\ldots,t_n) =_e f(t_1,\ldots,t_{i-1},x,t_{i+1},\ldots,t_n) \text{ for all } C \in Mod(T)]$ for all $f \in F_{s_1\ldots s_n,s}$.
6 The relation $\equiv$ over $T_\Sigma(X)$, defined by $t \equiv t'$ iff $A \text{ ‘ } t =_e t'$ for all $A \in Mod(T)$, is a congruence and $T_\Sigma(X)/\equiv$ is the initial model.

2.5 Total and partial conditional types

In this section we show how, reducing total conditional and partial positive conditional to non-strict conditional types, the well-known results about the existence of an initial object in those cases may be obtained as corollaries of theorem 2.4.1. Moreover we show that partial non-positive conditional specifications, ie partial specifications whose axioms are in conditional form, but involve both *strong* and existential equalities, reduce to non-strict disjunctive types and that the necessary and sufficient conditions for the existence of an initial model, given in [AC1, C] can be deduced from the theorem 2.3.4.

Properly speaking, the connection between total conditional types and conditional non-strict types and the connection between partial conditional types and conditional non-strict types are simulations (see [AC2]). To keep the presentation as concrete as possible, we simply sketch the translation of the theories in the two cases and do not define formally the simulations.

Let us recall the definition of validity both in total and partial algebras, just in order to fix the notation.

Def. 2.5.1. Let $\Sigma = (S,F)$ be a signature, A be a strict algebra over Σ and φ be a well formed formula over Σ and an S-sorted family X of variables;

$A \text{ ‘}^{Tot} \varphi$ iff $A \text{ ‘}_V \varphi$ for all total valuations V for the variables of φ in A.

Let $T = (\Sigma, Ax)$ be a type; $PMod(T)$ denotes the class of all partial models of T, ie $PMod(T) = \{A \mid A \in NSAlg(\Sigma), A \text{ strict}, A \text{ ‘}^{Tot} \alpha, \forall \alpha \in Ax\}$, and $TMod(T)$ denotes the class of all total models of T, ie $TMod(T) = \{A \mid A \in NSAlg(\Sigma), A \text{ total}, A \text{ ‘}^{Tot} \alpha, \forall \alpha \in Ax\}$.

Since partial algebras coincide with strict algebras, $PMod(T)$ is the class of partial models of T in the usual sense and analogously $TMod(T)$ is the class of total models of T in the usual sense, too.

Prop. 2.5.2. Let $\Sigma = (S,F)$ be a signature, $T = (\Sigma,Ax)$ be a conditional type.

1 Let $Par(T)$ be the conditional type $(\Sigma,Ax_{Str}\cup tot(Ax))$, where $tot(Ax) = \{tot(\alpha) \mid \alpha\in Ax\}$, $tot(\wedge\Delta\supset\varepsilon) = \wedge(\Delta\cup\{D(y) \mid y\in Var(\wedge\Delta\supset\varepsilon)\})\supset\varepsilon$ and Ax_{Str} consists of $D(f(y_1,\ldots,y_n))\supset D(y_i)$ for all $f\in F_{s1\ldots sn,s}$.
The partial model class of T coincides with the class of all non-strict models of $Par(T)$, ie $PMod(T) = Mod(Par(T))$.

2 Let $Tot(T)$ be the conditional type $(\Sigma,Ax_{Tot}\cup tot(Ax))$, where Ax_{Tot} consists of all the axioms of Ax_{Str} and of $\wedge\{D(y_i) \mid i=1\ldots n\}\supset D(f(y_1,\ldots,y_n))$ for all $f\in F_{s1\ldots sn,s}$.
The total model class of T coincides with the class of all non-strict models of $Tot(T)$, ie $TMod(T) = Mod(Tot(T))$.

Now we can get the well known results of existence of an initial model for partial positive conditional (see e.g. [BW1,B]) and for total conditional (see e.g. [MG]) types just as a corollary of theorem 2.4.1.

Theorem 2.5.3. Let $\Sigma = (S,F)$ be a signature, $T = (\Sigma,Ax)$ be a conditional type; using the notation of prop. 2.5.2 both $TMod(T)$ and $PMod(T)$ have an initial model.

Let us finally consider the partial conditional case, ie partial models of axioms of the form $\wedge\Delta\supset\varepsilon$, where $\Delta\cup\{\varepsilon\}$ is a set of possibly *strong* equalities. Let us recall that $t = t'$ holds iff $(\neg D(t)\wedge\neg D(t'))\vee t =_e t'$ holds, so that strong equality is only a short notation for a particular kind of disjunction.

Def. 2.5.4. Let $\Sigma = (S,F)$ be a signature and X be an S-sorted family of variables. The set of *non-positive conditional* formulas over Σ and X consists of
$\{\wedge\Delta\supset\varepsilon \mid \Delta\cup\{\varepsilon\}\subseteq EEq(\Sigma,X)\cup SEq(\Sigma,X)\}$, where $SEq(\Sigma,X) = \{(t = t') \mid t,t'\in T_\Sigma(X)\}$.

Let A be a non-strict algebra over Σ; if φ is a non-positive conditional formula and V is a valuation for $Var(\varphi)$ in A, then we say that φ *holds for* V *in* A (equivalently: *is satisfied for* V *by* A) and we write $A\,{}'_V\,\varphi$ accordingly with the following definition:
$A\,{}'_V\,t = t'$ iff $A\,{}'_V\,(\neg D(t)\wedge\neg D(t'))\vee t =_e t'$ and $A\,{}'_V\,\wedge\Delta\supset\varepsilon$ iff either $A\,{}'_V\,\varepsilon$ or there exists $\delta\in\Delta$ s.t. $A\,{}'_V\,\delta$.

We write $A\,{}'\,\varphi$ for a non-positive conditional formula φ and say that φ *holds in* (equivalently: *is satisfied by, is valid in*) A iff $A\,{}'_V\,\varphi$ for all partial valuations V for $Var(\varphi)$ in A. Moreover we write $A\,{}'^{Tot}\,\varphi$ iff $A\,{}'_V\,\varphi$ for all total valuations V for the variables of φ in A.

Remark. Let us consider a non-positive conditional formula $\wedge\Delta\supset\varepsilon$; if ε is the strong equality $t = t'$, then $\wedge\Delta\supset\varepsilon$ is logically equivalent to the couple of axioms $\wedge\Delta\cup\{D(t)\}\supset t =_e t'$ and $\wedge\Delta\cup\{D(t')\}\supset t =_e t'$. Thus from now on we assume that the consequences of the non-positive conditional formulas are always existential equalities.

A non-positive conditional axiom is logically equivalent to an implication involving only existential equalities, but with negated existential equalities in its premises, so that it is not a conditional axiom.
Let us consider the simplest example: a conditional axiom having just one strong equality in its premises; let α be $t = t'\supset\varepsilon$. Then α is logically equivalent to $(\neg D(t)\wedge\neg D(t'))\vee t =_e t'\supset\varepsilon$ and hence to the set consisting of $\neg D(t)\wedge\neg D(t')\supset\varepsilon$ and of $t =_e t'\supset\varepsilon$. If more than one strong equalities appear in the premises a little more machinery is needed.

For example let us consider the formula $\varphi = \wedge \{D(t), t_1 = t'_1, t_2 = t'_2\} \supset \varepsilon$; then φ is equivalent to $D(t) \wedge [(\neg D(t_1) \wedge \neg D(t'_1)) \vee t_1 =_e t'_1] \wedge [(\neg D(t_2) \wedge \neg D(t'_2)) \vee t_2 =_e t'_2] \supset \varepsilon$, which becomes the set consisting of the following four formulas:

- $D(t) \wedge (\neg D(t_1) \wedge \neg D(t'_1)) \wedge (\neg D(t_2) \wedge \neg D(t'_2)) \supset \varepsilon$
- $D(t) \wedge t_1 =_e t'_1 \wedge (\neg D(t_2) \wedge \neg D(t'_2)) \supset \varepsilon$
- $D(t) \wedge (\neg D(t_1) \wedge \neg D(t'_1)) \wedge t_2 =_e t'_2 \supset \varepsilon$
- $D(t) \wedge t_1 =_e t'_1 \wedge t_2 =_e t'_2 \supset \varepsilon$.

Generalizing this procedure, we have that one non-positive conditional axiom $\wedge \Delta \supset \varepsilon$ is logically equivalent to the set of all implications having as consequence ε and as premises the conjunction of all the existential equality in Δ with, for all $t = t' \in \Delta$, either $t =_e t'$ or both $\neg D(t)$ and $\neg D(t')$.

Prop 2.5.5. Let Σ be a signature, $\Gamma \cup \{\varepsilon\}$ be a set of both strong and existential open equalities over Σ, A be a non-strict algebra over Σ and V be a valuation for the variables of $\Gamma \cup \{\varepsilon\}$ in A.
Then $A \text{ ‘}_V \wedge \Gamma \supset \varepsilon$ iff for all Δ_1, Δ_2 s.t. $\Delta_1 \cup \Delta_2 = \Gamma\text{-EEq}(\Sigma,X)$

$A \text{ ‘}_V \wedge [(\Gamma \cap EEq(\Sigma,X)) \cup (\{t =_e t' \mid t=t' \in \Delta_1\} \cup \{\neg D(t), \neg D(t') \mid t=t' \in \Delta_2\})] \supset \varepsilon$.

Then, using the notation of prop. 2.5.5, each non-positive conditional axiom $\wedge \Gamma \supset \varepsilon$ is logically equivalent to the set

$\{\wedge [(\Gamma \cap EEq(\Sigma,X)) \cup (\{t=_e t' \mid t=t' \in \Delta_1\} \cup \{\neg D(t), \neg D(t') \mid t=t' \in \Delta_2\})] \supset \varepsilon \mid \Delta_1 \cup \Delta_2 = \Gamma\text{-EEq}(\Sigma,X)\}$

and hence, putting the above formulas in disjunctive form to the set

$\{\vee (\{\varepsilon\} \cup \{D(t), D(t') \mid t=t' \in \Delta_2\} \cup \{\neg t =_e t' \mid t = t' \in \Delta_1\} \cup \{\neg\gamma \mid \gamma \in \Gamma \cap EEq(\Sigma,X)\}) \mid \Delta_1 \cup \Delta_2 = \Gamma\text{-EEq}(\Sigma,X)\}$

Prop. 2.5.6. Let $\Sigma = (S,F)$ be a signature, $\varphi = \wedge \Gamma \supset \varepsilon$ be an open non-positive conditional formula and Y denote the set of the variables of φ.
Let us denote by $tot(\varphi)$ the set of all disjunctive formulas of the form:
$\vee (\{\varepsilon\} \cup \{D(t), D(t') \mid t=t' \in \Delta_2\} \cup \{\neg t =_e t' \mid t = t' \in \Delta_1\} \cup \{\neg\gamma \mid \gamma \in \Gamma \cap EEq(\Sigma,Y)\} \cup \{\neg D(y) \mid y \in Y\})$
for all Δ_1 and Δ_2 s.t. $\Delta_1 \cup \Delta_2 = \Gamma\text{-EEq}(\Sigma,Y)$.
For any strict algebra A we have that $A \text{ ‘}^{Tot} \varphi$ iff $A \text{ ‘ } \vartheta$ for all $\vartheta \in tot(\varphi)$.
The above propositions gives us sufficient tools to show that each non-positive partial conditional may be reduced to a disjunctive non-strict type.

Prop. 2.5.7. Let $\Sigma = (S,F)$ be a signature, $T = (\Sigma, Ax)$ be a non-positive partial type.
Using the notation of prop. 2.5.6, let T' be the conditional type $(\Sigma, Ax_{Str} \cup \cup_{\alpha \in Ax} tot(\alpha))$, where Ax_{Str} consists of $D(f(y_1,\ldots,y_n)) \supset D(y_i)$ for all $f \in F_{s_1 \ldots s_n, s}$.
The partial model class of T coincides with the class of all non-strict models of T', ie $PMod(T) = Mod(T')$.
Let us show now that the necessary and sufficient conditions for the existence of an initial model in the case of (non-positive) partial conditional types, given in [AC1,C], can be deduced by condition 5 of theorem 2.3.4.

Theorem 2.5.8. Let T be a non-positive conditional type. There exists an initial model in $PMod(T)$ iff for any instantiation $\varphi = \alpha[t_y/y \mid y \in Y]$ of an axiom α by defined closed terms t_y, ie $A \text{ ‘ } D(t_y)$ for all $A \in PMod(T)$ and all $y \in Var(\alpha)$,

1 either for all $A \in PMod(T)$ $A \Leftarrow cons(\varphi)$;

2 or there exists $t =_e t' \in prem(\varphi)$ s.t. $A \Rightarrow t =_e t'$ for some $A \in PMod(T)$;

3 or there exists $t = t' \in prem(\varphi)$ s.t. either for all $A \in PMod(T)$ $A \Leftarrow D(t)$, or for all $A \in PMod(T)$ $A \Leftarrow D(t')$ and $B \Rightarrow t =_e t'$ for some $B \in PMod(T)$.

Conclusion

The paper has presented two main results, clarifying, we believe, two basic issues.

First *the relationship between non-strict and totalized algebras* is analyzed, emphasing that the relationship has been dealt with at three different levels: models, categories, specifications. Only at the first level, which is the only considered in basic denotational semantics, the correspondence is trivial. An even deeper analysis can be given by using the concepts of *institution* (see e.g. [GB1]), *simulation* (see e.g. [AC2]) and *map of institution* (see e.g. [M]). The concept of simulation lifts the idea of implementation to the level of formal systems and formalizes the idea that elements living in different frameworks are in some sense equivalent, w.r.t. the problem faced at that moment. The existence of a simulation from an institution into another one means that signatures and sentences are translated from the first into the second institution in a way that each model in the first institution is represented by a (generally more than one) model in the second institution with satisfies exactly (in some sense) the same sentences. Thus each model in the first institution may be seen as an abstraction of its *implementations* in the second institution. This generalizes to classes of models, but in general not to types (ie theories), ie in general for a given theory (specification) in the first institution there does not exists a theory in the second institution whose models are exactly all the implementations of its models. This becomes true whenever the class of all models of the second institution representing some model of the first one are the model class of some theory, ie whenever the simulation is also a map of institutions; in this case also the logical aspect is preserved.

On the basis of the above concepts the correspondence between non-strict and total algebras is completely characterized by giving three simulations.

- non-strict algebras with equalities by total algebras with atomic formulas;
- non-strict algebras with conditional formulas by total algebras with conditional formulas;
- non-strict algebras with disjuntive formulas by total algebras with disjunctive formulas;

only the third one is also a map of institutions. We refer to the full paper [AC3] for this analysis.

The second main result is a *theory of initial models of non-strict disjunctive types* which encompasses not only the well-known initial theories of total conditional specifications and partial positive-conditional specifications, but also of the recently explored partial nonpositive conditional specifications.

We see a main direction for further research, i.e. allowing in the same paradigm don't care conditions and error-handling; a promising approach seems to merge the present theory with the development of error-handling in [P].

Acknowledgement.

We thank some referees for their careful and competent comments which, together with suggesting improvements, have pointed out a misprint in the definition of homomorphism (the preliminary condition

"$f^A(\underline{a}) \in s^A$ implies" in Def.1.1.3 was unfortunately missing), clearly causing ambiguities and inconsistencies with the counterexamples.

References

[AC1] Astesiano E.; Cerioli M. "On the Existence of Initial Models for Partial (Higher-Order) Conditional Specifications", *Proc. TAPSOFT'89*, vol.1, Lecture Notes in Computer Science n. 351, 1989 .

[AC2] Astesiano E.; Cerioli M. "Commuting between Institutions via Simulation" submitted for acceptance, also available as Technical Report n. 2, 1990, Formal Methods Group, University of Genoa (Italy).

[AC3] Astesiano E.; Cerioli M. "Non-strict don't care algebras and specifications", Technical Report n. 5, 1990, Formal Methods Group, University of Genoa (Italy).

[B] Burmeister P. *A Model Theoretic Oriented Approach to Partial Algebras*, Akademie-Verlag Berlin, 1986.

[Br] Broy M. Partial Interpretation of Higher-Order Algebraic Types, in *Proc. of Mathematical foundations of Computer Science 1986*, Lecture Notes in Computer Science n.233.

[BW1] Broy M.; Wirsing M. "Partial abstract types", *Acta Informatica* 18 (1982).

[BW2] Broy M.; Wirsing M., "Generalized Heterogeneous Algebras and Partial Interpretations" *Proc. CAAP'84,* Lecture Notes in Computer Science n.159, 1984.

[C] Cerioli M. "A sound and equationally-complete deduction system for partial conditional (higher order) types", *Proc 3rd Italian Conference of Theoretical Computer Science*, World Scientific,1989.

[GB1] Goguen J.A.; Burstall R.M. "Introducing Institutions". *Logic of Programs*, Lecture Notes in Computer Science n. 164, 1984.

[GB2] Goguen J.A.; Burstall R.M. "Institutions: Abstract Model Theory for Specification and Programming". Technical Report of Computer Science Laboratory, SRI International, 1990.

[K] Keisler H.J. *Model Theory for Infinitary Logic*, North-Holland, 1971.

[M] Meseguer J. "General logic", in *Proc. Logic Colloquium '87,* H.-D. Ebbinghaus et al. (eds.), North-Holland, 1989

[MG] Meseguer J.; Goguen J.A. "Initiality, Induction and Computability", in *Algebraic Methods in Semantics*, Cambridge University Press, 1985.

[MS] Manca V.; Salibra A "Equational Calculi for Many-sorted Algebras with Empty Carriers", *Proc. Mathematical Foundations of Computer Science 1990*, Lecture Notes in Computer Science n.452, 1990.

[P] Poigné A. “Partial Algebras, Subsorting, and Dependent Types” , *Proc. 5th Workshop on Specification of Abstract Data Types, Gullane, Scotland September ’87*, Lecture Notes in Computer Science n. 332, 1988.

[R] Reichel H. *Initial Computability, Algebraic Specifications, and Partial Algebra*s, Akademie-Verlag, Berlin, 1986.

[T] Tarlecki A. “Quasi-varieties in Abstract Algebraic Institutions”, *Journal of Computer and System Science*, n. 33, 1986.

[W] Wirsing M. “Algebraic Specification” in *Handbook of Theoretical Computer Science,* vol.B, Elsevier, 1990.