Inductive Completion with Retracts

William P.R.Mitchell*
Department of Computer Science
University of Manchester
Manchester M13 9PL

July 24th 1987, Revised Oct 1987

Summary In this paper we give countably infinitely many extensions of Huet and Hullot's inductive completion procedure. Also we try and throw some light on inductive problems when functions are only partially defined by some set of rewrite rules. We also give a procedure which attempts to show that two derived F-algebras are isomorphic when both of the algebras are realised as retracts.

1 Introduction

Let F be a set of function symbols and R be a rewrite system over $\tau(F \cup \nu)$, the universal algebra constructed from F and a countably infinite set of variables ν. How can we decide whether an equation $M = N$ is an inductive theorem (defn 3.4)of R or not ? The technique investigated here comes under the general notion of 'inductionless induction'. This methods works by completing $R \cup \{M = N\}$ in such a way that whilst completing it simultaneously checks to see if $M = N$ is an inductive theorem of R.

Let $\tau(F)/{=_R}$ be the initial algebra of R, $\tau(F)/{=_{R\cup\{M=N\}}}$ be the initial algebra of $R \cup \{M = N\}$ and $P : \tau(F)/{=_R} \longrightarrow \tau(F)/{=_{R\cup\{M=N\}}}$ be given by sending the equivalence class $[x]_R$ to the equivalence class $[x]_{R\cup\{M=N\}}$. It is straightforward to see that $M = N$ is an inductive theorem of R iff

*Supported by Science and Enginering Research Council under grant GR/D/81718. Whilst the paper was under revision the author was supported by S.E.R.C grant GR/D/57942

P is a bijection. Let $p : \tau(F) \longrightarrow \tau(F)/=_R$ be given by sending a term to its equivalence class. Now suppose we have a canonical rewrite system B which is equationally equivalent to $R \cup \{M = N\}$ on ground terms. Let $I(B)$ be the set of irreducible ground terms of B. The map P is a bijection iff $p : I(B) \longrightarrow \tau(F)/=_R$ (which is p restricted to $I(B)$) is a bijection. Thus $M = N$ is an inductive theorem of R iff $p : I(B) \longrightarrow \tau(F)/=_R$ is a bijection. An inductive completion procedure, when given R and $M = N$, uses some mutation of the Knuth Bendix procedure to try and produce a B for which it can decide whether $p : I(B) \longrightarrow \tau(F)/=_R$ is a bijection. In all cases such procedures restrict what R is allowed to be in order to make $\tau(F)/=_R$ easier to handle.

The inductive completion procedure given in [5] is now commonly regarded as the first really significant inductive completion procedure. This procedure only works when R satisfies the 'principle of definition'. This principle of definition is that there must be a set of symbols $C \subseteq F$ such that $p : \tau(C) \longrightarrow \tau(F)/=_R$ is a bijection. That is every term in $\tau(F)$ is congruent mod R to a unique term in $\tau(C)$. The set C is known as the set of constructors or H.H operators, and the inductive completion procedure is often referred to as the H.H procedure. The H.H procedure works by trying to find a B such that $I(B) = \tau(C)$, then if R satisfies the principle of definition it follows that $p : I(B) \longrightarrow \tau(F)/=_R$ is a bijection.

This approach although much more successful than previous methods can't be applied to some very natural rewrite systems because they do not satisfy the principle of definition. One such example is the standard rewrite system which describes the integers in terms of the functions S (successor), P (predecessor), 0 (zero), + (addition) and $\times$(multiplication). This example can't satisfy the principle of definition because the constructors would have to be the set $\{S, P, 0\}$, and this leads to a contradiction because $S(P(0)) = 0$. Many papers have now been published which give inductive completion procedures which can be applied to many more rewrite systems than the H.H procedure can. I do not intend to give a complete bibliography of these papers here, but amongst them are [3], [6], [8], [7], [12] and [14]. A rescent survey paper on inductive completion is [2].

In this paper we define a set of subsets of $\tau(F)$ denoted $M\tau$ which are built out of constructor symbols in some way. Each $\Delta \in M\tau$ has its own inductive completion procedure, in the following sense. We define a pair (R, Δ) to be a retract if every ground term is congruent mod R to a unique term in Δ, that is $p : \Delta \longrightarrow \tau(F)/=_R$ is a bijection. We then give a procedure which, on input of the retract (R, Δ) plus an equation, tries to decide whether the equation is an inductive theorem of R. This idea of retract allows us to have relations amongst the constructors.

In section 2 we define $M\tau$. We show that if we are given a rewrite rule and $\Delta \in M\tau$ then it is decidable whether Δ has an element which can be reduced by this rewrite rule. We also show that it is decidable whether there are two elements in Δ such that the first reduces to the second, in one step by the given rewrite rule. In the third section we develop the general theory which enables us to verify our inductive completion procedure in the fourth section. In section 4 we prove that each element of $M\tau$ has a completion procedure. In the last section we illustrate the use of retracts and inductionless induction. We see that under certain circumstances we have a procedure which attempts to prove that two F-algebras are isomorphic, and we give an example for which the method succeeds.

When we develop the idea of retracts, in section 4, we also develop the idea of a partially inductive theorem. This in some sense allows functions which are only partially defined. We see that even if a rewrite system R is not a full retract we still can use our inductive completion procedure to show that an equation is a partial inductive theorem of R. Unfortunately the notion of partially inductive theorems appears to be to nebulous to be of much practical use, though it may prove to be of theoretical importance.

All of the theorems in this paper are stated and proved for single sorted algebras only. This is to keep the notation as simple as possible. It is easy to see how the results generalise to the multisorted case. Also in this paper we do not consider AC operators because we wish to avoid the superfluous complication this causes, but the results do generalise to allow for them.

Notation

In general the convention is that the notation follows that given in [4]. The set F denotes a set of function symbols which is assumed to be finite for the purposes of this paper. Each function symbol has finite, fixed arity. The set ν denotes a countably infinite set of variables and $\tau(F \cup \nu)$ is the universal algebra which consists of terms made from the function symbols in F and variables in ν. The ground terms of $\tau(F \cup \nu)$ are denoted $\tau(F)$.

A rewrite rule is a pair $l \longrightarrow r$ where $l, r \in \tau(F \cup \nu)$. For a term t, $\nu(t)$ denotes the set of variables occurring in t. A rewrite system R is a set of rewrite rules such that $\nu(r) \subseteq \nu(l)$ for every rule $l \longrightarrow r$ in R.

For a term t and an occurrence u we write t/u for the subterm of t whose root occurs at u (see [4]). The term $t[u \leftarrow q]$ is t with the subterm t/u replaced with q. Let $\mathcal{O}(t)$ be the set of occurrences in t.

We write $t \longrightarrow_R s$ for terms t and s when there is a rule $l \longrightarrow r \in R$, a substitution $\Theta : \nu \longrightarrow \tau(F \cup \nu)$ and an occurrence u of t such that $t/u = \Theta(l)$ and $t[u \leftarrow \Theta(r)] = s$. The congruence relation $\longrightarrow_R^*$ on $\tau(F \cup \nu)$ is generated by the transitive reflexive closure of $\longrightarrow_R$. The relation $=_R$ is the transitive closure of $\longrightarrow_R^* \cup \longleftarrow_R^*$. Two rewrite systems R and S are equationally equivalent if $=_R$ and $=_S$ are the same. An equation $t = s$ is a theorem of R if $t =_R s$.

A canonical rewrite system is one which is confluent and terminating. For any rewrite system R let $I(R)$ denote the set of irreducible ground terms. A term t is irreducible if there is no term s such that $t \longrightarrow_R s$. For a term t we will let $t\!\downarrow$ denote any term which is irreducible and such that $t \longrightarrow_R^* t\!\downarrow$. Note that $t\!\downarrow$ is not unique for a non canonical rewrite system.

If f is a function symbol then $f^{(n)}(x)$ denotes $\underbrace{f(\ldots f}_{n}(x)\ldots)$. If f is a function of arity n defined on a set D then $f(D^n)$ denotes the image of the function, i.e the set $\{f(d_1, \ldots, d_n) \mid d_i \in D \text{ for } i = 1, \ldots, n\}$.

2 Properties of $M\tau$

In this short section we describe an interesting class of subsets of $\tau(F)$ which we will denote $M\tau$. Let $\Delta \in M\tau$. In a later section we will see that if (R, Δ) is a retract (see definition 3.6) then there is an inductive completion procedure which attempts to decide if an equation is an inductive theorem of R. Let $G \subseteq F$ and $A \subseteq \tau(F \cup \nu)$. Define $\tau(G \cup A) = \{\theta(t) \mid t \in \tau(G \cup \nu)$ and θ is a substitution such that $\theta : \nu \longrightarrow A\}$. So for example $\tau(\tau(F)) = \tau(F)$ and $\tau(F \cup \tau(F)) = \tau(F)$. Note though that in general it is NOT the case that $\tau(G_1 \cup \tau(G_2)) = \tau(G_1 \cup G_2)$.

Definition 2.1 *Let $C_1, \ldots, C_n$ be subsets of F such that $C_1 \cup \cdots \cup C_n$ contains at least one constant symbol. Define $M\tau(C_1, \ldots, C_n)$ to be a set whose elements are subsets of $\tau(F)$, described as follows.*

- *For each i $\tau(C_i) \in M\tau(C_1, \ldots, C_n)$ if C_i contains a constant symbol.*
- *If Δ_1 , $\Delta_2 \in M\tau(C_1, \ldots, C_n)$ then $\Delta_1 \cup \Delta_2 \in M\tau(C_1, \ldots, C_n)$.*
- *If $\Delta \in M\tau(C_1, \ldots, C_n)$ then for each i $\tau(C_i \cup \Delta) \in M\tau(C_1, \ldots, C_n)$.*

□

For the time being abbreviate $M\tau(C_1, \ldots, C_n)$ to $M\tau$. We have fixed the definition of $M\tau$ so that it doesn't contain the empty set in order to make some of the proofs below more readable. Notice that any $\Delta \in M\tau$ must contain a constant symbol. Thus for example

$$\tau(C_i) \in M\tau(C_1, \ldots, C_n)$$

$$\tau(C_1 \cup \tau(C_2)) \in M\tau(C_1, \ldots, C_n)$$

$$\tau(C_1 \cup \tau(C_2 \cup \tau(C_1)) \in M\tau(C_1, \ldots, C_n)$$

$$\tau(C_1) \cup \ldots \cup \tau(C_n) \in M\tau(C_1, \ldots, C_n).$$

If C_1 and C_2 are disjoint, non empty sets of symbols (say each contains a constant and non constant symbol) then $\tau(C_1 \cup \tau(C_2 \cup \tau(C_1))$ is a strict superset of $\tau(C_1 \cup \tau(C_2))$. Note that $\tau(C_1 \cup C_2) \notin M\tau(C_1, \ldots, C_n)$, but this is only a technical point since we can define $D_0 = C_1 \cup C_2$ and $D_i = C_i$ for $i > 0$. Then $\tau(C_1 \cup C_2) \in M\tau(D_0, D_1, \ldots, D_n)$.

Elements of $M\tau$ are interesting in their own right because they can be used to represent significant structures. For example we can represent polynomials in variables $X_1, \ldots, X_n$ as follows. Put $C_1 = \{+\}, C_2 = \{*, X_1, \ldots, X_n\}, C_3 = \{+, 1\}, C_4 = \{0\}$. The polynomials in $X_1, \ldots, X_n$ with positive integer coefficients are in 1-1 correspondence with

$$(\ \tau(\ C_1 \cup \tau(C_2) \cup \tau(C_3) \ ) \cup \tau(C_4) \ \ )/AC$$

where AC is the congruence relation generated by the associative and commutative rules for $+$ and $*$. The symbols $X_1, \ldots, X_n, 0, 1$ are constants. Obviously we have that

$$\tau(\ C_1 \cup \tau(C_2) \cup \tau(C_3) \ ) \cup \tau(C_4) \ \in \ M\tau(C_1, C_2, C_3, C_4).$$

Definition 2.2 *Let u be an occurrence of a term P. If $u = \Lambda.u_1.u_2 \ldots u_{n-1}$ where $u_i \in \mathbf{N}$ for each i then u has depth n and we write $d(u) = n$ .If u is at the end of a branch β of P we say β has depth n denoted $d(\beta) = n$. If β is the branch of P with maximum depth then we say the depth of P is n and write $d(P) = n$. If P is the left hand side of a rule in a rewrite system R and has maximum depth of all the left hand sides then n is the depth of R and denoted $d(R)$.*

Let "τ", "$\cup$", "C_1",…,"C_n" be new function symbols where "τ" is unary, "$\cup$" is binary and "C_1", …,"C_n" are constants. Let AC be the congruence relation generated by the associative and commutative rules for "$\cup$". Define $\mathcal{M}$ to be the set of terms
$t \in \tau(\{``\cup", ``\tau", ``C_1", \ldots, ``C_n"\})/AC$ such that any constant "C_i" in t must occur within some subterm of t whose root symbol is "τ". For each element t of $\mathcal{M}$ we can define a subset $\pi(t)$ of $\tau(F \cup \nu)$ as follows. If $t =$ "τ"("C_i") then $\pi(t) = \tau(C_i)$. If $t = t_1$"$\cup$"t_2, where t_1, $t_2 \in \mathcal{M}$ then $\pi(t) = \pi(t_1) \cup \pi(t_2)$. If $t =$ "τ"("C_{i_1}""$\cup$""C_{i_2}""$\cup$" … "C_{i_N}""$\cup$"t_1) where $t_1 \in \mathcal{M}$ then $\pi(t) = \tau(C_{i_1} \cup C_{i_2} \cup \ldots C_{i_N} \cup \pi(t_1))$. Clearly for each element

Δ of $M\tau$ there is at least one element Δ' of $\mathcal{M}$ such that $\pi(\Delta') = \Delta$. Any term $\Delta' \in \mathcal{M}$ such that $\pi(\Delta') = \Delta$ is said to *represent* Δ.

Definition 2.3 *For $\Delta \in M\tau$ define $d(\Delta)$ as follows. Choose a term $\Delta' \in \mathcal{M}$ which has minimum depth as a term and which represents Δ. Define $d(\Delta)$ to be the depth of Δ'.* $\square$

Thus for example $d(\tau(C_i)) = 2$, and $d(\tau(C_i \cup \tau(C_i))) = 2$. When $C_i \not\subseteq C_j$ and $C_j \not\subseteq C_i$ then $d(\tau(C_i \cup \tau(C_j))) = 4$. Let $\Delta \in M\tau$ and let Δ' be an element of $\mathcal{M}$ with minimal depth such that $\pi(\Delta') = \Delta$. We call such a Δ' a *minimal representation* of Δ.

Lemma 2.4 *If $\Delta \in M\tau$ then $\forall t \in \Delta$ and $\forall u \in \mathcal{O}(t)$ we have $t/u \in \Delta$.*

Proof. The proof goes by induction on $d(\Delta)$.
*Base case.*When $d(\Delta) = 2$, Δ can only be of the form $\tau(C_i)$ for some i and the result is trivial.
Induction case When $d(\Delta) > 2$ we have the following possible cases.
Case 1. $\Delta = \Delta_1 \cup \Delta_2 \cup \ldots \cup \Delta_m$ where $\Delta_i \in M\tau$ and $d(\Delta_i) \leq d(\Delta) - 1$. If $t \in \Delta$ then $t \in \Delta_i$ for some i and by the induction hypothesis $\forall u \in \mathcal{O}(t)$ we have $t/u \in \Delta_i \subseteq \Delta$.
Case 2.When $\Delta = \tau(C_i \cup \Delta_1)$. Let $t \in \Delta$. Thus there is a term $s \in \tau(C_i \cup \nu)$ and a substitution $\theta : \nu \longrightarrow \Delta_1$ such that $\theta(s) = t$. Let $V(s) = \{v \in \mathcal{O}(s) \mid s/v \in \nu\}$. Let $u \in \mathcal{O}(t)$. If $u \in \mathcal{O}(s) - V(s)$ then $\theta(s)/u = \theta(s/u)$, since $s/u \in \tau(C_1 \cup \nu)$ it follows that $t/u \in \Delta$. If $u = v \cdot w$ where $v \in V(s)$ then $t/u \in \Delta_1$ and were done, since $\Delta_1 \subseteq \Delta$. $\square$

Consider the following problem. Given $\Delta \in M\tau$ and $\ell \to r$ a rewrite rule, is there a term in Δ which can be reduced by $\ell \to r$? By the above lemma this is equivalent to finding a ground substitution Θ such that $\Theta(\ell) \in \Delta$. The following lemma gives a simple criteria for deciding whether such a substitution exists.

Definition 2.5 *Let Δ be an element of $M\tau$, with a minimal representation Δ'. We define a subset $V\Delta$ of $\tau(F \cup \nu)$ as follows.*

1. *If* $\Delta' =$ "τ"("C_i") *then* $V\Delta = \tau(C_i \cup \nu)$.

2. *If* $\Delta' = \Delta'_1$"$\cup$"Δ'_2 *then* $V\Delta = V\Delta_1 \cup V\Delta_2$*, where* $\pi(\Delta'_i) = \Delta_i \in M\tau$.

3. *If* $\Delta' =$ "τ"("C_i""$\cup$"Δ'_1) *then* $V\Delta = \tau(C_i \cup (V\Delta_1 \cup \nu))$.

Lemma 2.6 *Let* $\Delta \in M\tau$*. For a term* $\ell \in \tau(F \cup \nu)$ *there is a substitution* Θ *such that* $\Theta(\ell) \in \Delta \Leftrightarrow \ell \in V\Delta$

Proof.

Define $\overline{\Delta} = \{\ell \in \tau(F \cup \nu) \mid \Theta(\ell) \in \Delta$ for some substitution $\Theta\}$

The statement of the lemma then is that $\overline{\Delta} = V\Delta$. Again we prove this by induction on $d(\Delta)$.

Base case. This is when $d(\Delta) = 2$. In this case Δ must be of the form $\tau(C_i)$ for which the result is trivial.

Induction case. This is when $d(\Delta) > 2$. We have the following two special cases.

Case 1. $\Delta = \Delta_1 \cup \Delta_2 \cup \ldots \cup \Delta_m$ where $\Delta_i \in M\tau$ and $d(\Delta_i) < d(\Delta)$. Thus

$$\overline{\Delta} = \overline{\Delta_1} \cup \overline{\Delta_2} \cup \ldots \cup \overline{\Delta_m}$$

and by induction

$$\overline{\Delta} = V\Delta_1 \cup \ldots \cup V\Delta_m = V\Delta$$

Case 2. When $\Delta = \tau(C_i \cup \Delta_1)$ where $\Delta_1 \in M\tau$ and $d(\Delta_1) = d(\Delta) - 1$. Clearly $\overline{\Delta} = \tau(C_i \cup \overline{\Delta_1} \cup \nu) = \tau(C_i \cup \nu \cup V\Delta_1) = V\Delta$. This completes the induction step. $\square$

The next result tells us when a rewrite rule can reduce some element of Δ to some other element of Δ in one step.

Lemma 2.7 *It is decidable given a pair of elements* ℓ , $r \in V\Delta$ *(where* $\Delta \in M\tau$*) whether there is a substitution* Θ *such that both* $\Theta(\ell)$ *and* $\Theta(r)$ *are elements of* Δ

Proof.

During this proof we will regard an element of $M\tau$ as being the same as its minimal representation. This means when we write something like 'Δ is of the form $\tau(C_i \cup \Delta')$ ' we mean that some minimal representation of Δ has the form "τ"("C_i""$\cup$"$\Delta 1$) where $\Delta 1 \in \mathcal{M}$ and $\pi(\Delta 1) = \Delta' \in \ M\tau$.

For a term t and an occurrence u let t_u be the root symbol of t/u. For each $u \in \mathcal{O}(l)$ such that $l/u \notin \nu$ choose $C_u \in \{C_1, \ldots, C_n\}$ such that $l_u \in C_u$. Let X be this choice i.e $X = \{C_u \in \{C_1, \ldots, C_n\} \mid u \in \mathcal{O}(l)$ and $l/u \notin \nu$ and $l_u \in C_u\}$. We call X a choice for l.

Define $T(X)$ to be a tree with one node for each of the C_u. Each node is labeled with one of the C_u, and the node C_u is a descendant of the node C_v iff u is a descendant of v. Also define a tree $T(\Delta)$ whenever Δ is of the form $\tau(C_i' \cup \Delta')$ as follows. If $\Delta = \tau(C_i)$ then $T(\Delta)$ has just one node which is labeled with C_i. If $\Delta = \tau(C_i \cup \Delta_1 \cup \ldots \Delta_n)$ where each Δ_j is of the form $\tau(C_{i_j} \cup \Delta_{i_j})$ then $T(\Delta)$ has root C_i which has n branches. At occurrence $\Lambda.r$, where $1 \leq r \leq n$ we place the tree $T(\Delta_r)$.

Given a tree T whose nodes are labeled by the C_is we can construct from it a unique element of $M\tau$, which we will also denote T. Infact $T(\Delta)$, Δ are the same element of $M\tau$.

We say that X is distinguished (for l) if there is a function

$$f : \mathcal{O}(T(X)) \longrightarrow \mathcal{O}(T(\Delta))$$

with the following properties.

1. If $f(u) = v$ then $C_u = T(\Delta)_v$.

2. If $f(u) = v$ then $f(\text{descendant of } u) = \text{descendant of } v$.

We call such a function an X-function. Note that v is a descendant of v. Given the above definitions we have the following fact which we will not prove. There is a substitution Θ such that $\Theta(l) \in \Delta$ iff there is a distinguished X.

It is clear from the definition that we can find all distinguished X. Assume that X is distinguished for the rest of the proof and let f be an X-function. For each $x \in \nu(l)$ let u_x be any occurrence such that for some integer i, $l/u_x.i = x$, if $l = x$ put $u_x = \Lambda$. Let Θ_X be any substitution such that $\Theta_X(x) \in T(\Delta)/f(u_x)$ for each $x \in \nu(l)$. Then $\Theta_X(l) \in \Delta$. Infact any Θ such that $\Theta(l) \in \Delta$ can be constructed in this way, for some X.

We now have the following procedure for determining whether there is a Θ such that $\Theta(l), \Theta(r) \in \Delta$. For the rest of the proof we will always use X to denote a distinguished choice for l and Y to denote a distinguished choice for r. Let *Test* be a boolean variable.

```
For each X,Y do

    For each X-function f and each Y-function g do
        Test := True
        For each x ∈ ν(l) ∩ ν(r) do
            If
                ⋂_{u_x} T(Δ)/f(u_x) ∩ T(Δ)/g(u_x) = ∅
            then Test := False
        End loop
        If Test = True then halt with output:
        "∃ Θ such that Θ(l), Θ(r) ∈ Δ ".
    End loop

End loop
If Test = False then halt with output:
"There is no Θ such that Θ(l), Θ(r) ∈ Δ ".
```

To check if $T(\Delta)/f(u_x) \cap T(\Delta)/g(u_x) = \emptyset$, we only have to test if $T(\Delta)/f(u_x)$ and $T(\Delta)/g(u_x)$ have any constant symbols in common. Note that the above procedure only works when $\Delta = \tau(C_i' \cup \Delta')$, but it is easy to see how to generalise to the case $\Delta = \Delta_1 \cup \ldots \cup \Delta_n$. $\Box$

Lemmas 2.6 and 2.7 will be important later on. Lemma 2.6 states that is decidable for a given rule whether there is any element of Δ which can be reduced by that rule. Lemma 2.7 says that it is decidable for a given rule whether there are a pair of elements in Δ such that the first reduces to the second, in one step, by the given rule.

3 Inductive and Partially Inductive Theorems

In this section we develop the general theory behind the inductive completion procedure of the next section. The results in this section are independent of the previous section. The results here go some way towards suggesting how we might tackle the problem of partially defined functions.

For the rest of this section let $\Delta \subseteq \tau(F)$ be fixed. Let R be a rewrite system and define

$$\Delta_R = \{t \in \tau(F) \mid \exists s \in \Delta \text{ such that } t =_R s\}.$$

Definition 3.1 *Let A be a subset of $\tau(F)$. We say that (R, Δ) retracts A iff*

$$\forall s \in A \; \exists\,!\, t \in \Delta \text{ such that } s =_R t.$$

We shall say R retracts A instead of (R, Δ) retracts A, whenever it causes no confusion.

Definition 3.2 *We define (R, Δ) to be rounded iff for every $t \in \Delta_R$ and every occurrence u of t we have that $t/u \in \Delta_R$. We will say R is rounded instead of saying (R, Δ) is rounded whenever it causes no confusion.*

For example, suppose that $\Delta = \tau(C)$ for some subset C of F and $\Delta_R = \tau(F)$. Then trivially R is rounded, further R retracts $\Delta_R \Leftrightarrow R$ satisfies Huet and Hullots' principle of definition [5].

Proposition 3.3 *Let R_1, R_2 be two rewrite systems. Let (R_1, Δ) be rounded and retract Δ_{R_1} also let $=_{R_1} \subseteq =_{R_2}$. If Δ_{R_1} is closed under R_2 then (R_2, Δ) retracts $\Delta_{R_1} \Leftrightarrow \forall \ell \to r \in R_2$ and $\forall\, \Theta : \nu \to \tau(F)$ if $\Theta(\ell), \Theta(r) \in \Delta_{R_1}$ then $\Theta(\ell) =_{R_1} \Theta(r)$.*

Proof.

$' \Rightarrow '$ (By contradiction). Suppose, in order to get a contradiction, that $\exists \ell \to r \in R_2$ and $\exists \Theta : \nu \to \tau(F)$ such that $\Theta(\ell), \Theta(r) \in \Delta_{R_1}$ and $\Theta(\ell) \neq_{R_1} \Theta(r)$.

Since R_1 retracts Δ_{R_1} $\exists\,!\, \alpha, \beta \in \Delta$ with $\alpha \neq \beta$ and $\Theta(\ell) =_{R_1} \alpha, \Theta(r) =_{R_1} \beta$. But $=_{R_1} \subseteq =_{R_2}$ and thus $\alpha =_{R_2} \Theta(\ell) =_{R_2} \Theta(r) =_{R_2} \beta$. However R_2 retracts Δ_{R_1} and therefore $\alpha \neq_{R_2} \beta$ as α and β are distinct. Thus we have a contradiction as required.

$' \Leftarrow '$

The easiest way to see this is to prove that for $a, b \in \Delta_{R_1}$ we have $a =_{R_1} b \Leftrightarrow a =_{R_2} b$. Since $=_{R_1} \subseteq =_{R_2}$ we only need to prove the reverse implication. Suppose that $a, b \in \Delta_{R_1}$ and for some $\ell \longrightarrow r \in R_2$ some $\Theta : \nu \to \tau(F)$ and some $u \in \mathcal{O}(a)$ we have $a/u = \Theta(\ell)$ and $a[u \leftarrow \Theta(r)] = b$. If we prove that $a =_{R_1} b$ then we are done. Since R_1 is rounded $\Theta(\ell), \Theta(r) \in \Delta_{R_1}$. Hence $\Theta(\ell) =_{R_1} \Theta(r)$.

Trivialy this means $a =_{R_1} b$ and we are done. □

Definition 3.4 *Let R be a set of rewrite rules over $\tau(F \cup \nu)$. If $N, M \in \tau(F \cup \nu)$ and $\Theta(M) =_R \Theta(N)$ for every ground substitution Θ then $M = N$ is an inductive theorem of R.*

Note that if $\Delta_{R_1} = \tau(F)$ then it follows automatically that R_1 is rounded and that Δ_{R_1} is closed under R_2. Given this situation we have the following obvious corollary.

Corollary 3.5 *Let R_1, R_2 be two rewrite systems. Let $\Delta_{R_1} = \tau(F)$ and let (R_1, Δ) retract $\tau(F)$, also let $=_{R_1} \subseteq =_{R_2}$.*
Then (R_2, Δ) retracts $\tau(F) \Leftrightarrow \forall \ell \to r \in R_2$ the equation $\ell = r$ is an inductive theorem of R_1.

Example.

Let $F = \{s, 0, \ell t\}$ where s is unary, 0 is constant and ℓt is binary.
Let R_1 be the following rules,

$$\ell t(0, s(0)) \longrightarrow s(0)$$

$$\ell t(s(x), s(y)) \longrightarrow \ell t(x, y),$$

and let $\Delta = \tau(\{s, 0\})$. We can think of Δ as the natural numbers and ℓt as a predicate which partially defines the $<$ relation on natural numbers i.e. $\ell t(0, s(0)) \to s(0)$ represents $0 < 1$ and the second rule represents $a+1 < b+1 \Leftrightarrow a < b$. Note, for example, that R_1 does not evaluate $\ell t(0, 0)$, that is $\ell t(0, 0)$ is not equivalent to any element of Δ. This means that the predicate lt as defined by R_1 on the natural numbers is only partially defined.

Let R_2 be $R_1 \cup \{\ell t(x, s(x)) \to s(0)\}$. It is left to the reader to check that R_1 and R_2 satisfy the hypothesis of propn. 3.3. Thus to show that $\ell t(\alpha, s(\alpha))) =_{R_1} s(0)$ for every α in Δ it would be sufficient to prove that R_2 retracts Δ_{R_1}. This is again left as an exercise for the reader, but is indeed true. $\square$

This example, although an interesting illustration of the use of propn. 3.3 to prove properties of partial functions (namely ℓt), has skipped over all the glaringly difficult problems. Some of which are how to determine that R_1 is rounded, how to show that R_1 retracts Δ_{R_1}, and how to see that Δ_{R_1} is closed under R_2.

Unfortunately (if we are to use some automatic procedure to solve these problems) the only way we intend to deal with these problems is to avoid them, by insisting that $\Delta_{R_1} = \tau(F)$. Thus automatically R_1 is rounded and Δ_{R_1} is closed under R_2. However in restricting to this case we have excluded the possibility of having partially defined functions, which is when $\Delta_{R_1} \neq \tau(F)$. We do have a result which deals with this case, 4.16, but the hypothesis is rather hard to test in practice. The following definition is equivalent to that given by Musser for unambiguity [12].

Definition 3.6 *Let R be a rewrite system. We define (R, Δ) to be a retract if (R, Δ) retracts $\tau(F)$. We will say R is a retract, instead of (R, Δ) is a retract whenever it causes no confusion.*

Note that if R is a retract then $\Delta_R = \tau(F)$ and R is rounded. Again notice that when $\Delta = \tau(C)$ where $C \subseteq F$, that this is Huet and Hullot's principle of definition. We can now restate corollary 3.5 as follows.

Corollary 3.7 *Let R_1, R_2 be rewrite systems. Let (R_1, Δ) be a retract and let $=_{R_1} \subseteq =_{R_2}$. Then (R_2, Δ) is a retract $\iff$*
$\forall \ell \to r \in R_2$ the equation $\ell = r$ is an inductive theorem of R_1. $\square$

4 The Inductive Completion Procedure

In this section the results of the last two sections will be combined to give our inductive completion procedure, and then to verify it. Again let Δ be a fixed subset of $\tau(F)$.

Definition 4.1 *Let (R, Δ) be a rounded rewrite system. We define an equation $s = t$ to be a partial inductive theorem (of (R, Δ)) iff for every $\Theta : \nu \to \tau(F)$ such that $\Theta(s), \Theta(t) \in \Delta_R$ we have $\Theta(s) =_R \Theta(t)$*

Proposition 4.2 *Let R_1, R_2 be rewrite systems which satisfy the hypothesis of propn. 3.3. Suppose also that Δ_{R_1} is closed under R_2, that R_2 is canonical and that $I(R_2) \cap \Delta_{R_1} \subseteq \Delta$. Then*
$I(R_2) \cap \Delta_{R_1} = \Delta \Leftrightarrow \forall \ell \to r \in R_2$ the equation $\ell = r$ is a partial inductive theorem of (R_1, Δ).

Proof.
By proposition 3.3 we only need to show that $I(R_2) \cap \Delta_{R_1} = \Delta \Leftrightarrow R_2$ retracts Δ_{R_1}. However R_2 retracts Δ_{R_1} iff the quotient map

$$\Delta \to \Delta_{R_1}/ =_{R_2}$$

(given by the composition $\Delta \hookrightarrow \Delta_{R_1} \to \Delta_{R_1}/ =_{R_1} \to \Delta_{R_1}/ =_{R_2}$) is a bijection. Moreover we have the following commutative diagram.

$$\begin{tikzcd}
\Delta \arrow[d] & I(R_2) \cap \Delta_{R_1} \arrow[l] \arrow[dl] \\
\Delta_{R_1}/{=_{R_2}} &
\end{tikzcd}$$

Where the horizontal arrow is just the inclusion given by the hypothesis, and the diagonal arrow is given by sending a term to its conjugacy class. Because R_2 is canonical the diagonal arrow is a bijection. Hence the result.
$\square$

Corollary 4.3 *(c.f [12])*
Let R_1, R_2 be rewrite systems. Let (R_1, Δ) be a retract and let $=_{R_1} \subseteq =_{R_2}$. If R_2 is canonical and $I(R_2) \subseteq \Delta$ then $I(R_2) = \Delta \Leftrightarrow \forall \ell \to r \in R_2$ the equation $\ell = r$ is an inductive theorem of R_1. □

Really the conditions of corollary 4.3 are not too extreme. In order that R_1 be a retract we are simply asking that Δ be an F-algebra. Where for each $f \in F$ and $a_1, \ldots, a_n \in \Delta$ we can define $f(a_1, \ldots, a_n)$ to be a unique term $b \in \Delta$ such that $f(a_1, \ldots, a_n) =_{R_1} b$.

For the rest of this section we will assume that $\Delta \in M\tau$, ($M\tau$ is shorthand for $M\tau(C_1, \ldots, C_n)$).

Definition 4.4 *Define* $V\Delta 2 = \{(l, r) \mid \exists\, \Theta : \nu \longrightarrow \tau(F \cup \nu) : \Theta(l)\ ,\ \Theta(r) \in \Delta\}$. □

Notice that for a rewrite rule $\ell \to r$, there is an element of Δ which $\ell \to r$ can be applied to $\Leftrightarrow \ell \in V\Delta$. Also notice that there are elements $a, b \in \Delta$ such that a reduces to b, in one step, by $\ell \to r \Leftrightarrow (\ell, r) \in V\Delta 2$. It is important to notice that by lemma 2.7 it is decidable whether a pair of terms is an element of $V\Delta 2$. As an example if $\Delta = \tau(C)$, for $C \subseteq F$, then $V\Delta = \tau(C \cup \nu)$ and $V\Delta 2 = (\tau(C \cup \nu))^2$.

Inductive Completion Procedure

The procedure Ind-Comp-Proc below has the following properties (which we will prove later on). On input of a retract (R, Δ), (with $\Delta \in M\tau$), plus a set of equations ε, if the procedure returns a rewrite system then every equation of ε is an inductive theorem of R. If the procedure returns "False" then (at least) one of the equations is not an inductive theorem of R. In fact we have a stronger result than this (proposition 4.16) for when R is not a retract, but in practise it is difficult to apply. The procedure Ind-Comp-Proc calls two other procedures, $P1$ and test. These are also listed below. The symbol $>$ represents a fixed simplification ordering of terms.

Ind-Comp-Proc = Proc(R:Rewrite System, $\Delta : M\tau$, ε:Set of Equations)
returns(Rewrite System **or** STRING)

S:Set of Equations

$S := \{\lambda = \rho \mid \lambda \longrightarrow \rho \in R\} \cup \varepsilon$

B:Rewrite System

$B := \emptyset$

While TRUE do

X:(Set of Equations, Rewrite System)

$X := (S, B)$

If $P1(X)$ is of type STRING then return(P1(X)).

$(S, B) := P1(X)$

If $S = \emptyset$ then return(B).

end (of while loop)

end Ind-Comp-Proc

$P1$ = Proc(S:Set of Equations,B:Rewrite System)
returns((Set of Equations, Rewrite System) **or** STRING)

λ, ρ:Term

If $S = \emptyset$ then return($(S,\ B)$).

Choose $M = N \in S$.

Let $N{\downarrow}$, $M{\downarrow}$ be normal forms of N, M with respect to B.

If $N{\downarrow} = M{\downarrow}$ then return($(S - \{M = N\},\ B)$).

If $(M{\downarrow}, N{\downarrow}) \in V\Delta 2$ or $(N{\downarrow}, M{\downarrow}) \in V\Delta 2$ then return("FALSE").

If $M{\downarrow}, N{\downarrow} \in V\Delta$ then return("Fail").

If $M{\downarrow} \in V\Delta, N{\downarrow} \notin V\Delta$ then

if test($N{\downarrow}$, $M{\downarrow}$) then $\lambda := N{\downarrow}, \rho := M{\downarrow}$ else return("Fail").

If $N{\downarrow} \in V\Delta, M{\downarrow} \notin V\Delta$ then if test($M\downarrow$, $N\downarrow$) then $\lambda := M\downarrow, \rho := N\downarrow$ else return("Fail").

If test($N\downarrow$, $M\downarrow$) then $\lambda := N\downarrow, \rho := M\downarrow$.

If test($M\downarrow$, $N\downarrow$) then $\lambda := M\downarrow, \rho := N\downarrow$.

If test($M\downarrow$, $N\downarrow$) **and** test($N\downarrow$, $M\downarrow$) = FALSE then return("Fail").

B' :Rewrite System

$B' := \{\lambda' \longrightarrow \rho' \in B \mid \lambda'$ *or* ρ' contains an instance of λ as a subterm $\}$

$B := (B - B') \cup \{\lambda \longrightarrow \rho\}$

$S := (S - \{M = N\}) \cup \{\lambda' = \rho' \mid \lambda' \longrightarrow \rho' \in B'\} \cup \{P = Q \mid \langle P, Q\rangle$ is a critical pair of $B\}$.

return((S, B))

end $P1$

test = Proc(a:Term, b:Term) returns(BOOL)

If $a > b$ **and** $\nu(b) \subseteq \nu(a)$ then return(TRUE) else return(FALSE).

end test □

The following results (Propn 4.10 and Propn 4.11) prove that the above procedure satisfies the properties we say it does. After that we try and find a procedure which is sufficient to prove that a rewrite system R is a retract, though if it fails it does not imply that R is not a retract. From now until lemma 4.12 we assume that R is a retract.

Lemma 4.5 *If Ind-Comp-Proc returns the rewrite system B then B is canonical and equationally equivalent to $R \cup \varepsilon$.*

Proof.
At every stage of Ind-Comp-Proc we know that B is terminating because we use the simplification ordering $>$. Since the algorithm does not return "fail" or "False" it performs exactly as the Knuth Bendix algorithm would

on input $R \cup \varepsilon$. Thus the result is immediate. $\square$

Lemma 4.6 *If the inductive completion procedure returns the rewrite system B then Δ is closed under B.*

Proof

Suppose for a contradiction that there are $t \in \Delta$, $s \notin \Delta$ such that $t \longrightarrow_B s$. Then there is an occurrence u of t, a substitution Θ, and a rule $l \longrightarrow r \in B$ such that $t/u = \Theta(l)$, $t[u \longleftarrow \Theta(r)] = s$. Since $t \in \Delta$ we have that $l \in V\Delta$. However there is an explicit clause in $P1$ which prevents such a rule being included in B, which contradicts that $l \longrightarrow r \in B$. $\square$

Definition 4.7 *For a rewrite system B let $I(B)$ be the set of irreducible ground terms with respect to B.*

Lemma 4.8 *If the inductive completion procedure returns the rewrite system B then $\Delta \subseteq I(B)$.*

Proof.

Again proof by contradiction. If the result is false, since Δ is closed under B by lemma 4.6, there are t , $s \in \Delta$ such that $t \longrightarrow_B s$. Thus for some u , Θ there is a rule $l \longrightarrow r \in B$ such that $t/u = \Theta(l)$, $t[u \longleftarrow \Theta(r)] = s$. Hence $l, r \in V\Delta$. However there is an explicit clause in $P1$ to prevent such a rule ever being added to B. $\square$

Lemma 4.9 *If the inductive completion procedure returns the rewrite system B then $\Delta = I(B)$.*

Proof.

Clearly $=_R \subseteq =_B$. Thus $\forall\ t \in \tau(F)\ \exists\ q \in \Delta$ such that $t =_B q$. By lemma 4.8 q is in canonical normal form hence $t \longrightarrow^*_B q$. Thus $I(B) \subseteq \Delta$ as required. $\square$

Proposition 4.10 *If the inductive completion procedure returns rewrite system B then every equation of ε is an inductive theorem of R.*

Proof.

We have just proved that B is canonical and that $I(B) = \Delta$. Thus by corollary 4.3 $\forall \ell \to r \in B$ the equation $\ell = r$ is an inductive theorem of R. Since B is equationally equivalent to $R \cup \varepsilon$ it is straightforward to see that every equation of ε is an inductive theorem of R. $\square$

Proposition 4.11 *If the inductive completion procedure returns "False" then some equation of ε is not an inductive theorem of R.*

Proof.

Let us suppose that the procedure $P1$ has returned "FALSE" . The procedure returns "FALSE" iff $\exists a, b, \in \Delta$ such that $a =_{B \cup S} b$. But since $=_{B \cup S}$ is the same as $=_{R \cup \varepsilon}$ we know that $R \cup \varepsilon$ is not a retract and so by corollary 3.7 some equation of ε is not an inductive theorem of R. $\square$

Example

In this example we use a version of our inductive completion procedure which has been adapted to allow associative, commutative operators. Take $F = \{s, 0, +, \star, m1\}$, where S is unary, $+$ and $\star$ are binary and $0, m1$ are constants. Let $C_1 = \{s, 0\}$ and $C_2 = \{m1, +\}$. We will take the symbols $+$ and $\star$ to be associative and commutative. Set $\Delta = \tau(C_1) \cup \tau(C_2)$. We should intuitively think of Δ as the integers, where $m1$ is minus 1. So we think of $\tau(C_1)$ as the set of non-negative integers and $\tau(C_2)$ as the set of negative integers.

Let R be the following set of rewrite rules, where $+$ and $*$ are AC.

```
1.   0 + x -> x
2.   0 * x -> 0
3.   s(y) + x -> s(x + y)
4.   s(m1 + x) -> x
```

```
 5.   s(y) * x -> (x * y) + x
 6.   m1*m1 -> s(0)
```

Using the standard completion algorithm it is not possible to complete these rules (that is if we turn the rules back into an equational system S and run the Knuth Bendix algorithm on S the algorithm does not appear to terminate). If we run the inductive completion procedure on R plus $x * (y + z) = x * y + x * z$ it returns the complete rewrite system below, proving that the equation $x * (y + z) = x * y + x * z$ is an inductive theorem of R.

```
Rewrite rules:


 1.   0 + x -> x
 2.   0 * x -> 0
 3.   s(y) + x -> s(x + y)
 4.   s(m1 + x) -> x
 5.   s(y) * x -> (x * y) + x
 6.   (y + z) * x -> (x * y) + (x * z)
 7.   s(m1) -> 0
 8.   (m1 * x) + x -> 0
 9.   m1 * m1 -> s(0)


Equational theory:
    AC:     *    +
```

□

Of course the problem with the inductive completion procedure is that in order to know that the procedure gives a correct answer, on input of $R \cup \varepsilon$, we have to know that R is a retract. The following results give some help towards solving this problem. (Note we are no longer assuming R to be a retract).

Lemma 4.12 *Let R be a rewrite system such that $\Delta_R = \tau(F)$. Suppose that ε is a set of equations such that $R \cup \varepsilon$ is a retract. Then every equation of ε is an inductive theorem of R.*

Proof.

All we need to do is prove that R is a retract. Since $\Delta_R = \tau(F)$ this means we only need to show that for $a, b \in \Delta$ with $a \neq b$ we have that $a \neq_R b$. For a contradiction suppose $\exists a, b \in \Delta$ with $a \neq b$ but with $a =_R b$. However $a =_R b \Rightarrow a =_{R \cup \varepsilon} b$ which implies $R \cup \varepsilon$ is not a retract which is a contradiction. □

From this we can easily see that the following holds.

Lemma 4.13 *Let R be a rewrite system (which is not necessarily known to be a retract) such that $\Delta_R = \tau(F)$. If Ind-Comp-Proc(R, Δ, ε) returns a rewrite system then every equation of ε is an inductive theorem of R.*

Note however that when we only know $\Delta_R = \tau(F)$ (and we don't know that R is a retract) and the procedure returns "False" then we are non the wiser about ε.

Given an R which is canonical we have the following sufficient condition to show that $\Delta_R = \tau(F)$.

Definition 4.14 *Let R be a rewrite system. Define*

1. $\Omega_R = \{t \in V\Delta \mid d(t) \leq d(R) + 1$ *and if* $t/u \in \nu$ *for some occurrence* u *then* $d(u) = d(R) + 1\}$.

2. $\gamma_R = \{t \in \tau(F \cup \nu) \mid t = f(w_1, \ldots, w_n)$ *where* $\forall i\ w_i \in \Omega_R$ *and* $\exists$ *a substitution* Θ *with* $\Theta(w_i) \in \Delta$ *for each* i *and* $\Theta(t) \notin \Delta\}$.

Note that Ω_R is finite and can be calculated since we can decide whether a given term is in $V\Delta$ and there are only finitely many terms of depth less than or equal to $d(R) + 1$. We can also calculate γ_R.

Lemma 4.15 *Let R be a canonical rewrite system. If, for each $f \in F$, every element of the set*

$$f((\Omega_R)^n) - V\Delta$$

is reducible (where $n =$ arity of f) and $V\Delta \cap \gamma_R = \emptyset$ then $\Delta_R = \tau(F)$. □

We now have the following procedure to try and show that an equation (or equations) is an inductive theorem of a rewrite system R.

First use lemma 4.15 to show that $\Delta_R = \tau(F)$. Assuming this succeeds we use the inductive completion procedure on R plus the equation. Then by lemma 4.13 if the procedure returns a rewrite system the equation is an inductive theorem of R. Of course if it returns "False" we know nothing.

This is not quite the whole story. Suppose we take a rewrite system R which has $\Delta_R = \tau(F)$. We now run the inductive completion procedure on just R. If the procedure returns a rewrite system then we have that R is a retract. Thus we can now use the inductive completion procedure with the full weight of propns 4.10 and 4.11 behind us.

In fact this is still not quite the whole story for we have the following result (which as it stands is probably less helpful than the above remarks).

Proposition 4.16 *Let R be a rounded rewrite system which retracts Δ_R. Let ε be a set of equations such that Δ_R is closed under $R \cup \varepsilon$. If Ind-Comp-Proc(R, Δ, ε) returns a rewrite system, then every equation of ε is a partial inductive theorem of R. If it returns "False" then some equation is not a partial inductive theorem of R.*

The proof of this is analogous to the proof of proposition 4.10. The reason that the proposition is not so helpful is that we have no idea how to show R to be rounded in general or how to determine very much about Δ_R when it is not $\tau(F)$.

5 Isomorphisms of Derived Algebras

Now we have our inductive completion procedure we have the problem of what do we use it for. In this section we try to show how we can sometimes prove that two F-algebras are isomorphic.

Suppose we have two sets of function symbols, F_1 and F_2, and two sets of rewrite rules R_1 and R_2. Also we have that $\Delta_1, \Delta_2 \in M\tau$ and that R_i is a retract for Δ_i for $i = 1, 2$. Now we have a third set of function symbols F and we impose an F-algebra structure on Δ_1 and Δ_2 in some way. We would like to have a procedure which tells us if just the F-algebra structures of Δ_1 and Δ_2 are enough to induce an F-algebra isomorphism of Δ_1 and Δ_2. Of course we're going to have to be rather careful about what sort of F-algebra structures we allow.

Definition 5.1 *Let $\delta \in \tau(F_i \cup \nu)$ with distinct variables $x_1, \ldots, x_n$ as they occur from left to right. The term δ induces, in an obvious way, a function*

$$\overline{\delta} : \tau(F_i \cup \nu)^n \longrightarrow \tau(F_i \cup \nu)$$

Namely if we have $t_1, \ldots, t_n \in \tau(F_i \cup \nu)$ let Θ be the substitution given by $\Theta(x_i) = t_i$ and then define $\overline{\delta}(t_1, \ldots, t_n)$ to be $\Theta(\delta)$.

We now restrict ourselves to F-algebra structures on Δ_i of the following kind. For each $f \in F$ of arity n there must be a term $\delta_i(f) \in \tau(F_i \cup \nu)$, such that for $a_1, \ldots, a_n \in \Delta_i$ we have

$$f(a_1, \ldots, a_n) = b \in \Delta_i,$$

where b is the unique term such that $b =_{R_i} \overline{\delta_i(f)}(a_1, \ldots, a_n)$. Since R_i is a retract this means the F-algebra structure is well defined.

We wish to introduce multi-sorts at this point. Let $\mathcal{F} = F_1 \cup F_2$, let S_1, S_2 be sorts and for $f \in F_i$ let f be of type $S_i \times \cdots \times S_i \to S_i$. Now let i, j be two new function symbols where i is of type $S_1 \to S_2$ and j is of type $S_2 \to S_1$. Define the substitution $\Theta(i)$ to be given by $\Theta(i)(x) = i(x)$ for $x \in \nu$, further define the rewrite system $R(i)$ to be

$$R(i) = \{ i(\delta_1(f)) \longrightarrow \Theta(i)(\delta_2(f)) \mid \delta_1\ ,\ \delta_2 \text{ are as above } \}$$

Similarly we define $R(j)$ by swapping 1 with 2 and replacing i by j in the above. Note we are assuming that the variables in $\delta_1(f)$ and $\delta_2(f)$ are labeled consistently otherwise the above definitions make no sense. We think of $R(i)$ as trying to define one half of an isomorphism of Δ_1 to Δ_2. Let $\overline{\imath}$ be a partially defined function $\Delta_1 \longrightarrow \Delta_2$ defined by
$\overline{\imath}(a) = b \iff a \in \Delta_1, b \in \Delta_2$ and b is a unique element such that $i(a) =_{R_1 \cup R_2 \cup R(i)} b$

We would like to say that $\overline{\imath}$ is in fact a function from Δ_1 to Δ_2. This is so if $R_1 \cup R_2 \cup R(i)$ is a retract (of $\tau(\mathcal{F} \cup \{i\})$ to $\Delta_1 \cup \Delta_2$).

Similarly we can define a partial function $\bar{j} : \Delta_2 \longrightarrow \Delta_1$ which is a total function if $R_1 \cup R_2 \cup R(j)$ is a retract (of $\tau(\mathcal{F} \cup \{j\})$ to $\Delta_1 \cup \Delta_2$). Note that if $R_1 \cup R_2 \cup R(i)$, $R_1 \cup R_2 \cup R(j)$ are retracts then $\bar{\imath}$ and $\bar{\jmath}$ are F-algebra morphisms. This is since they were defined precisely so that they would preserve the F-algebra structure.

Let us now suppose we have shown $R_1 \cup R_2 \cup R(i)$, $R_1 \cup R_2 \cup R(j)$ to be retracts , we now want to prove that Δ_1 and Δ_2 are isomorphic F-algebras (where $\bar{\imath}$ and $\bar{\jmath}$ would provide the isomorphism). To do this we only need to prove that $R_1 \cup R_2 \cup R(i) \cup R(j)$ is a retract and that the following two equations are inductive theorems of $R_1 \cup R_2 \cup R(i) \cup R(j)$.

1. $i(j(x)) = x$
2. $j(i(x)) = x$

Given that the reader believes all the results so far (including the inductive completion procedure itself) can be generalised to allow for multisorts, it is clear that the above problems can be dealt with by the methods we have at hand. We illustrate the above idea with the following example which assumes that our results apply to multisorted rewrite systems with associative commutative operators.

Example

Put $F_1 = \{S, 0, +\}, F_2 = \{1, +, 0\}$ and let $+$ be associative and commutative.
Let $R_1 = \{x + 0 \to x, S(x) + y \to S(x + y)\}$
$R_2 = \{x + 0 \to x\}$,
and $\Delta_1 = \tau(\{S, 0\}), \Delta_2 = \tau(\{+, 1\}) \cup \tau(\{0\})$

It is obvious that both Δ_1 and Δ_2 can be identified with the natural numbers with addition given by $a``+"b = (a + b) \downarrow_{R_i}$. Put $F = F_1$ and trivialy Δ_1 has an F-algebra structure. Let Δ_2 have an F algebra structure given by $\delta_2(0) = 0, \delta_2(S) = x + 1$ and $\delta_2(+) = x + y, (\delta_1(0) = 0, \delta_1(S) = S(x), \delta_1(+) = x + y)$.

Following our procedure we now introduce two new function symbols i, j

and rules

$$R(i) = \{i(0) \longrightarrow 0, i(S(x)) \longrightarrow i(x) + 1, i(x + y) \longrightarrow i(x) + i(y)\}$$

$$R(j) = \{j(x + 1) \longrightarrow S(j(x)), j(0) \longrightarrow 0, j(x + y) \longrightarrow j(x) + j(y)\}$$

In order to show that $\bar{\imath}$ and $\bar{\jmath}$ are well defined F-algebra morphisms we need to prove that $R_1 \cup R_2 \cup R(i)$ and $R_1 \cup R_2 \cup R(j)$ are retracts. If we now plug in the (generalised) results of the last section we find that they are retracts. We can now use the (generalised) inductive completion procedure to prove that $i \cdot j(x) = x$ and $j \cdot i(x) = x$ are inductive theorems of $R_1 \cup R_2 \cup R(i) \cup R(j)$. Thus Δ_1 and Δ_2 have been proved (by our automated procedure) to be the same F-algebra.

Now let us extend our example a bit further. Let lt be a new function symbol and let R'_1 be R_1 plus the rules

$$\ell t(0, x) \longrightarrow S(0)$$

$$\ell t(S(x), 0) \longrightarrow 0$$

$$\ell t(S(x), S(y)) \longrightarrow \ell t(x, y)$$

These rules represent the $\leq$ relation on the natural numbers where, for $a, b \in \Delta_1$, $\ell t(a, b) =_{R'_1} S(0) \Leftrightarrow a \leq b$.

Notice R'_1 is a retract (of $\tau(F'_1)$ to Δ_1). We would now expect to be able to prove, say, that $\ell t(x, x + x) = S(0)$ is an inductive theorem of R'_1 by use of the inductive completion procedure of the last section. However if we were to apply the procedure it would loop forever without halting. This is because it is not possible for the Knuth Bendix completion procedure to terminate when run on $R'_1 \cup \{\ell t(x, x + x) \to S(0)\}$

So what we do instead is we define R'_2 to be R_2 plus the rules

$$\ell t(0, x) \to 1$$

$$\ell t(x + 1, 0) \to 0$$

$$\ell t(x + 1, y + 1) \to \ell t(x, y).$$

Then we see R'_2 is a retract. Put $F' = F \cup \{\ell t\}$ and repeat the earlier part of the example to show that Δ_1 and Δ_2 are isomorphic as F'-algebras (with $\delta_i(\ell t) = \ell t(x, y)$). This succeeds and we have the following statement.

The equation $\ell t(x, x + x) = S(0)$ is an inductive theorem of R'_1
$\Leftrightarrow \ell t(x, x + x) = 1$ is an inductive theorem of R'_2.

We know this is so because an equation is an inductive theorem of R'_i iff it is valid in Δ_i and since Δ_1 and Δ_2 are the same F'-algebra the result is immediate.

When we now use our inductive completion procedure on $R'_2 \cup \{lt(x, x + x) = 1\}$ it returns a rewrite system. Thus our method has proved that $\ell t(x, x + x) = S(0)$ is an inductive theorem of R'_1. $\square$

Acknowledgements

I would like to express my thanks to Dr Ursula Martin for all her help and encouragement during the preparation of this paper.

References

[1] Dershowitz, N. 'Applications of the Knuth-Bendix Completion Procedure', Laboratory Operation Aerospace Corporation, Aerospace Report No .ATR-83(8478)-2, May15, 1983.

[2] Dershowitz, N. 'Completion and its Applications' in Proceedings of Colloquium on Resolution of Equations in Algebraic Structures. MCC 3500 West Balconies Center Drive, Austin. Texas 78759-6509. May 4-6 1987.

[3] Fribourg, F. 'A Strong Restriction of the Completion Procedure for Theories with Constructors.' in 'International Colloquium on Automata, Languages and Programming.' editor Kott, L. Springer Verlag, Lecture Notes in Computer Science, 1986.

[4] Huet, G. and Oppen, D. 'Equations and Rewrite Rules:a Survey' in 'Formal Languages: Perspectives and Open Problems'. Ed. Book ,R. Academic Press 1980.

[5] Huet, G. and Hullot, J.M. 'Proof by Induction in Equational Theories with Constructors' , JCCS 25, 2 ,1982 .

[6] Jouannaud, J.P. Kounalis, E. 'Proofs by Induction in Equational Theories without Constructors.' in 'Proceedings of the first IEEE Symposium on Logic in Computer Science', IEEE, Cambridge. Massachusetts 1986.

[7] Kapur, D. and Narendran, P. and Zhang, H. 'Proof by Induction using Test Sets' in 8th International Conference on Automated Deduction, Lecture Notes in Computer Science 230, 1986, 99-117.

[8] Kirchner, H. 'A General Inductive Completion Algorithm and Application to Abstract Data Types.' in 7th International Conference on Automated Deduction, Lecture Notes in Computer Science 170, 1984.

[9] Knuth, D. and Bendix, P. 'Simple Word Problems in Universal Algebras' in 'Computational Problems in Abstract Algebra'. Ed. Leech, J., Pergamon Press, 1970 ,263-297.

[10] Lescanne, P. 'Term Rewriting Systems and Algebra', 7th International Conference on Automated Deduction, Lecture Notes in Computer Science 170, 1984.

[11] Musser, D.R. 'On Proving Inductive Properties Of Abstract Data Types', Proceedings of the seventh ACM Symposium on Principals of Programming Languages ,Las Vegas, Nevada, Jan 1980 154-162.

[12] Musser, D.R. and Kapur, D. 'Proof by Consistency', Artificial Intelligence, 31(2), 125-157, 1987.

[13] Plaisted,D.A. 'Semantic Confluence Tests and Completion Methods',Information and Control 65,182-215, 1985.

[14] Tyama, Y. 'How to Prove Equivalence of Term Rewriting Systems without Induction.', 8th International Conference on Automated Deduction, Lecture Notes in Computer Science 230, 1986, 118-127.