Representing Inductively Defined Sets by Wellorderings in Martin-Löf's Type Theory

Peter Dybjer *

February 6, 1996

Abstract

We prove that every strictly positive endofunctor on the category of sets generated by Martin-Löf's extensional type theory has an initial algebra. This representation of inductively defined sets uses essentially the wellorderings introduced by Martin-Löf in "Constructive Mathematics and Computer Programming".

1 Background

Martin-Löf [10] introduced a general set former for wellorderings in intuitionistic type theory. It has formation rule

$$\frac{A\,set \qquad (x:A)\\ \qquad\qquad B(x)\,set}{W_{x:A}B(x)\,set}$$

introduction rule

$$\frac{a:A \qquad (x:B(a))\\ \qquad\qquad b(x):W_{x:A}B(x)}{sup(a,b):W_{x:A}B(x)}.$$

elimination rule

$$\frac{c:W_{x:A}B(x) \qquad (x:A,\,y:B(x)\to W_{x:A}B(x),\,z:\prod_{t:B(x)}C(y(t)))\\ \qquad\qquad d(x,y,z):C(sup(a,b))}{T(c,d):C(c)}$$

and equality rule

$$\frac{a:A \qquad (x:B(a))\\ \qquad b(x):W_{x:A}B(x) \qquad (x:A,\,y:B(x)\to W_{x:A}B(x),\,z:\prod_{t:B(x)}C(y(t)))\\ \qquad\qquad d(x,y,z):C(sup(a,b))}{T(sup(a,b),d)=d(a,b,\lambda t.T(b(t),d):C(c)}$$

The elimination rule can be viewed either as a rule of transfinite induction or as a rule of definition by transfinite recursion of a function f which maps a $c:W_{x:A}B(x)$ into a set $C(c)$. If we let

$$f(c)=T(c,d)$$

then the recursion equation

$$f(sup(a,b))=d(a,b,\lambda t.f(b(t)))$$

follows from the equality rule for wellorderings.

Martin-Löf [11] also showed that one can encode both the set of natural numbers and the set of ordinals of the second number class in terms of the wellorderings, the finite sets N_0, N_1, N_2, and the

*Author's address: Department of Computing Science, Chalmers University of Technology and University of Göteborg, S-412 96 Göteborg, Sweden.

first universe U. (N_i is the finite set with i elements $0_i, 1_i, \ldots, (i-1)_i$ and R_i is the i-ary conditional appearing in the elimination rule for N_i.)

The set N of natural numbers can be encoded by

$$N = W_{x:N_2} B_N(x),$$

where $B_N(0_2) = N_0$ and $B_N(1_2) = N_1$ (The explicit definition of B_N uses the first universe U.) Moreover,

$$\begin{aligned} 0 &= sup(0_2, (x)R_0(x)), \\ s(a) &= sup(1_2, (x)R_1(x, a)), \end{aligned}$$

and the recursion operator R for N can be defined by

$$R(n, d, e) = T(n, g)$$

where

$$\begin{aligned} g(0_2, y, z) &= d : C(0) = C(sup(0_2, R_0)) = C(sup(0_2, y)) \\ g(1_2, y, z) &= e(y(0_1), z(0_1)) : C(s(y(0_1))) = C(sup(1_2, (x)y(0_1))) = C(sup(1_2, y)) \end{aligned}$$

Here we have used that in extensional type theory

$$y = R_0 : N_0 \to C$$

for all $y : N_0 \to C$ and

$$y = (x)y(0_1) : N_1 \to C$$

for all $y : N_1 \to C$. These judgement are not derivable in *intensional* type theory, however, since the left-hand side and right-hand side of the respective equalities have different normal forms.

Similarly, we can define the ordinals of the second number class by

$$\mathcal{O} = W_{x:N_2} B_{\mathcal{O}}(x)$$

where $B_{\mathcal{O}}(0_2) = N_0$ and $B_{\mathcal{O}}(1_2) = N$.

We shall show that these are just two instances of a general representation theorem for inductively defined sets in extensional type theory.

2 The representation theorem

The general result refers to sets which are inductively generated by *strictly positive operators*, that is, operators built up from set variables and constants using $+$, $\times$, and $\to$ with the restriction that no set variable occurs to the left of $\to$. For example, the set N of natural numbers is generated by $\Phi(X) = N_1 + X$ and the set $\mathcal{O}$ of ordinals of the second number class is generated by $\Phi(X) = N_1 + (N \to X)$.

Each strictly positive operator can be extended to a *strictly positive functor* on the category of sets generated by extensional type theory. This category has sets in the sense of extensional type theory as objects and elements of the set $A \to B$ as morphisms. Two morphisms are equal iff they are equal elements of $A \to B$ iff they are extensionally equal functions from A to B. We can then formulate:

Theorem 1 *Each strictly positive functor on the category of sets generated by extensional type theory has an initial algebra.*

Proof. The initial algebra for the functor $\Phi(X) = \sum_{x:A}(B(x) \to X)$ is the arrow $i : \Phi(W) \to W$, where

$$W = W_{x:A} B(x)$$

and

$$i(\langle a, b\rangle) = sup(a, b).$$

Given another Φ-algebra $e : \Phi(C) \to C$, there is a morphism of Φ-algebras $h : W \to C$, where

$$h(c) = T(c, d)$$

for

$$d(x, y, z) = e(\langle x, z\rangle).$$

That h is the unique such morphism is an "η-rule" which follows from extensionality of equality of morphisms.

Theorem 1 now follows from lemma 3 below, which states that each strictly positive functor is isomorphic to a functor of the form $\Phi(X) = \sum_{x:A}(B(x) \to X)$ for some A and B.

The proof of lemma 3 uses the following fact:

Lemma 2 *The following isomorphisms are valid in the category of sets generated by extensional type theory:*

$$\begin{aligned}
N_1 &\cong N_0 \to A, && (1)\\
A &\cong N_1 \to A, && (2)\\
A &\cong A \times N_1, && (3)\\
A &\cong N_1 \times A, && (4)\\
\prod_{x:A}\sum_{y:B(x)} C(x,y) &\cong \sum_{f:\prod_{x:A}B(x)}\prod_{x:A} C(x, f(x)), && (5)\\
\prod_{x:A}\prod_{y:B(x)} C(x,y) &\cong \prod_{z:\sum_{x:A}B(x)} C(p(z), q(z)), && (6)\\
\sum_{x':A'} B'(x') + \sum_{x'':A''} B''(x'') &\cong \sum_{x:A'+A''} B(x). && (7)
\end{aligned}$$

where, in the last isomorphism $B(i(x')) = B'(x')$ and $B(j(x'')) = B''(x'')$.

The proof of lemma 2 uses extensionality. The reason is similar to what was explained in the introduction about the use of extensionality for proving the correctness of the encoding of natural numbers.

Lemma 3 *For any strictly positive set operator Φ, we can find a set A and a family of sets $B(x)$ indexed by elements $x : A$, such that, for all sets X*

$$\Phi(X) \cong \sum_{x:A}(B(x) \to X).$$

Proof. We proceed by induction on the structure of Φ. Assume for the induction step that $\Phi'(X) = \sum_{x:A'}(B'(x) \to X)$ for some A', B' and $\Phi''(X) = \sum_{x:A''}(B''(x) \to X)$ for some A'', B''. There are the following cases

Variable. $\Phi(X) = X$. Let $A = N_1$ and $B(x) = N_1$. We need to show that

$$X \cong N_1 \times (N_1 \to X).$$

But this follows directly from (4) and (2).

Constant. $\Phi(X) = K$ for some constant set K. Let $A = K$ and $B(x) = N_0$.

We need to show that

$$K \cong K \times (N_0 \to X).$$

But this follows directly from (3) and (1)

Disjoint union. $\Phi(X) = \Phi'(X) + \Phi''(X)$. Let $A = A' + A''$ and $B(i(x')) = B'(x')$ and $B(j(x'')) = B''(x'')$.

We need to show that

$$\sum_{x':A'}(B'(x') \to X) + \sum_{x'':A''}(B''(x'') \to X) \cong \sum_{x:A'+A''}(B(x) \to X),$$

where $B(i(x')) = B'(x')$ and $B(j(x'')) = B''(x'')$. But this follows directly from (7).

Cartesian product. $\Phi(X) = \Phi'(X) \times \Phi''(X)$. Let $A = A' \times A''$ and $B(\langle x', x'' \rangle) = B'(x') + B''(x'')$. We need to show that

$$\sum_{x':A'} (B'(x') \to X) \times \sum_{x'':A''} (B''(x'') \to X) \cong \sum_{x:A' \times A''} (B(x) \to X).$$

But this follows since we can construct the isomorphism pair (ι, ι'), where

$$\iota(\langle\langle a', f' \rangle, \langle a'', f'' \rangle\rangle) = \langle\langle a', a'' \rangle, f \rangle$$

for $f(i(x')) = f'(x')$ and $f(j(x'')) = f''(x'')$, and

$$\iota'(\langle\langle a', a'' \rangle, f \rangle) = \langle\langle a', f' \rangle, \langle a'', f'' \rangle\rangle$$

for $f'(x') = f(i(x'))$ and $f''(x'') = f(j(x''))$.

Function space. $\Phi(X) = K \to \Phi'(X)$ for some constant set K. Let $A = K \to A'$ and $B(f) = \sum_{y:K} B'(Ap(f, y))$. We prove

$$\begin{aligned} K \to \sum_{x':A'} (B'(x') \to X) &\cong \sum_{f:K \to A'} \prod_{y:K} (B'(Ap(f, y)) \to X) \\ &\cong \sum_{f:K \to A'} (\sum_{y:K} B'(Ap(f, y)) \to X) \end{aligned}$$

by using special cases of (5) and (6) respectively. □

We can now apply the proof to $\Phi(X) = N_1 + X$ and get an isomorphic representation of the natural numbers as $W_{x:N_1+N_1} B(x)$, where $B(i(0_1)) = N_0$ and $B(j(0_1)) = N_1$, which is essentially the same as [11] given above.

Remark. Lemma 3 is the key result of the paper and theorem 1 is just a corollary. Given an appropriate categorical formulation of initial algebras in general models of dependent types ([2], [12], [5], [7]), we believe a stronger version of theorem 1 could be formulated and proved as a corollary of lemma 3.

3 Some remarks on intensional type theory

As already mentioned, this representation does not work directly in intensional type theory. Consider for example Martin-Löf's representation of the natural numbers. In intensional type theory we can still derive the introduction rules but not the elimination rule. The reason is that we cannot prove by transfinite induction without using extensionality that $C(c)$ for all $c : W_{x:N_2} B_N(x)$ from $C(sup(0_2, (x)R_0(x)))$ and $C(a)$ implies $C(sup(1_2, (x)R_1(x, a))$. We cannot even prove $C(sup(0_2, b))$ from $C(sup(0_2, (x)R_0(x)))$, since we cannot prove that b and $(x)R_0(x)$ are intensionally equal.

For similar reasons the isomorphisms in lemma 2 cannot be proved if equality in our category of sets is equality in intensional type theory.

However, Hofmann [6] has shown how to build a category of *setoids* (sets with equivalence relations) inside intensional type theory. This category is a model of extensional type theory. By applying theorem 1 to this category we get an indirect representation theorem for inductively defined sets in intensional type theory.

4 Related work

Paulson [14] used wellorderings for deriving a variety of well-founded recursion operators in extensional type theory.

Palmgren [13] showed how to use wellorderings for representing inductively defined predicates in type theory.

Petersson and Synek [15] introduced a new set constructor for general trees in type theory. These trees are related to the wellorderings, but encode extra information which can also be used for representing inductively defined families of sets in a similar style as the representation described in the present paper.

General rules for initial algebras in dependent type theory can be found in Coquand and Paulin [2], Mendler [12], Geuvers [5], and Jacobs [7].

Natural deduction formulations for inductively defined sets in type theory can be found in Backhouse [1] and for inductively defined families in Dybjer [4, 3]. This is an alternative approach to inductive definitions in type theory modelled on Martin-Löf's natural deduction formulation of inductively defined predicates in predicate logic [9]. This approach has also been formulated for inductive definitions in the calculus of constructions by Coquand and Paulin [2] and Luo [8].

References

[1] R. Backhouse. On the meaning and construction of the rules in Martin-Löf's theory of types. In *Proceedings of the Workshop on General Logic, Edinburgh, February 1987*. Laboratory for Foundations of Computer Science, Department of Computer Science, University of Edinburgh, 1988. ECS-LFCS-88-52.

[2] T. Coquand and C. Paulin. Inductively defined types, preliminary version. In *LNCS 417, COLOG '88, International Conference on Computer Logic*. Springer-Verlag, 1990.

[3] P. Dybjer. Inductive sets and families in Martin-Löf's type theory and their set-theoretic semantics. In *Logical Frameworks*, pages 280–306. Cambridge University Press, 1991.

[4] P. Dybjer. Inductive families. *Formal Aspects of Computing*, pages 440–465, 1994.

[5] H. Geuvers. Inductive and coinductive types with iteration and recursion. Manuscript, October 1991.

[6] M. Hofmann. *Extensional concepts in intensional type theory*. PhD thesis, University of Edinburgh, 1995.

[7] B. Jacobs. Inductively defined types in dependent type theory. Draft version, 1993.

[8] Z. Luo. *Computation and Reasoning: A Type Theory for Computer Science*. Oxford University Press, 1994.

[9] P. Martin-Löf. Hauptsatz for the intuitionistic theory of iterated inductive definitions. In J. E. Fenstad, editor, *Proceedings of the Second Scandinavian Logic Symposium*, pages 179–216. North-Holland, 1971.

[10] P. Martin-Löf. Constructive mathematics and computer programming. In *Logic, Methodology and Philosophy of Science, VI, 1979*, pages 153–175. North-Holland, 1982.

[11] P. Martin-Löf. *Intuitionistic Type Theory*. Bibliopolis, 1984.

[12] N.-P. Mendler. Inductive types via initial algebras, in Martin-Löf type theory. Manuscript, July 1990.

[13] E. Palmgren. *On Fixed Point Operators, Inductive Definitions and Universes in Martin-Löf's Type Theory*. PhD thesis, Uppsala University, 1991.

[14] L. C. Paulson. Proving termination of normalization functions for conditional expressions. *Journal of Automated Reasoning*, 2:63–74, 1986.

[15] K. Petersson and D. Synek. A set constructor for inductive sets in Martin-Löf's type theory. In *Category Theory and Computer Science*, pages 128–140. Springer-Verlag, LNCS 389, 1989.