Ordinal Arithmetic: A Case Study for Rippling in a Higher Order Domain*

Alan Smaill** and Louise A. Dennis

Division of Informatics, University of Edinburgh, 80 South Bridge, Edinburgh, EH1 1HN, UK,
A.Smaill@ed.ac.uk, louised@dai.ed.ac.uk

1 Introduction

This abstract reports a case study for the automation of ordinal arithmetic in a higher order proof planning system, $\lambda Clam$. A theory of ordinal arithmetic was implemented and the system was used to plan proofs of examples appearing in a number of text books $[3, 1, 2]$ with encouraging results. As far as we know this is the first time that automation of such proofs has been attempted in a fully higher-order fashion.

2 A Theory of Ordinal Arithmetic

We know that 0 is an ordinal and that there are two ways of forming larger ordinals, either by applying the successor, or taking the union (limit) of a set of ordinals. Of these only the representation of limit ordinals is of interest. We wanted a common form for limit ordinals to exploit pattern matching. We chose the notation lim rather than the more ususal $\cup$. We use $\lim_\alpha F$ to represent $\lim_{\beta<\alpha} F(\beta)$. In particular we used $\alpha = \lim_\alpha \lambda x.x$ for limit ordinals.

The first two clauses in the definition of plus are as for the natural number type. The statement of continuity of addition is $x + \lim_{\beta<\alpha} F(\beta) = \lim_{\beta<\alpha}(x + F(\beta))$ and was used as a third definitional rewrite. Multiplication and exponentiation were given rewrite rules in the same way. We also included a small number of other rewrite lemmas. These included cancellation rules, identity and four lemmas expressing general properties of limits and limit ordinals.

The induction scheme used is standard for ordinals, though the condition that the variable is a limit ordinal in the "new" step case is omitted. Instead we use pattern matching with our representation of limit ordinals. The new step case, for a goal of the form $\forall x : ordl.P(x)$ is

$$\forall \beta : ordl.\beta < \alpha \to P(\beta) \vdash P(\lim_\alpha \lambda x.x). \tag{1}$$

The choice of initial embedding to motivate rippling here raises some problems. The goal is not syntactically an expansion of the hypothesis; there is no implication or quantifier in the conclusion. We use the notion of a *wild card* symbol, $*$. This allowed us to embed $P(*)$ in $P(\lim_\alpha \lambda x.x)$ by matching $*$ to $\lim_\alpha \lambda x.x$.

3 Evaluation

This machinery was enough to plan automatically 14 theorems drawn from exercises in three undergraduate text books $[3, 1, 2]$. They represent all the examples of ordinal arithmetic listed in those books which use only transfinite induction and the definitions of the arithmetical operators (i.e. which don't also use some set theoretic reasoning based on the definitions of ordinals) except for those involving the order relation.

* This research was funded by EPSRC grant Gr/m45030

** Corresponding Author

3.1 The Synthesis of a Fixpoint for a Continuous Function

One of our primary interests in creating $\lambda Clam$ was to investigate the use of least-commitment reasoning in order to perform synthesis proofs. By this we mean placing uninstantiated meta-variables in proof goals which become instantiated during the course of the proof and generate an existential witness. We looked at a simple case where this style of reasoning could be using in the ordinal domain. This synthesis proof made no use of induction.

Theorem 1. *Any continuous ϕ has a fixed point.*

$$\exists \eta . \phi(\eta) = \eta \tag{2}$$

We were able to synthesize such a fixed point using symbolic evaluation and three lemmas. The final proof went as follows.

$$\begin{aligned}
\exists \eta . \phi(\eta) &= \eta, & (3)\\
\phi(E) &= E, & (4)\\
\lim_X \lambda z . \phi(F(z)) &= \lim_X \lambda z . F(z), & (5)\\
\lim_X \lambda z . \phi^{s(z)}(A) &= \lim_X \lambda z . \phi^{z}(A), & (6)\\
\lim_\omega \lambda z . \phi^{z}(A) &= \lim_\omega \lambda z . \phi^{z}(A), & (7)\\
&T. & (8)
\end{aligned}$$

As can be seen the system gradually instantiates the meta-variable, E, as it rewrites the terms.

At the moment this proof only succeeds with a carefully chosen set of lemmas in place but we feel it demonstrates the possible applications for higher-order proof planning systems in the ordinal domain.

4 Conclusion

We have chosen to examine ordinal arithmetic as a test case for the extension of the ideas of rippling and proof planning in a higher order setting. Our aim was to maintain the natural higher-order presentation of the ordinalsn while retaining full automation of the proof generation process.

With a small amount of theory building and a minor extension to rippling in place we were able to successfully plan standard undergraduate textbook examples and exercises. We were also able to demonstrate how proof planning, in particular least commitment devises such as the use of meta-variables, could have a part to play in the mechanisation of proof theoretic results in the ordinal domain. These results were confirmation that $\lambda Clam$ could naturally handle higher order examples and that λProlog was a suitable language in which to express such mathematical concepts. This is also an illustration of the generality of the rippling heuristic.

References

1. Herbert B. Enderton. *Elements of Set Theory.* Academic Press, 1977.
2. A. G. Hamilton. *Numbers, sets and axioms: the apparatus of mathematics.* Cambridge University Press, 1982.
3. P. Suppes. *Axiomatic Set Theory.* Van Nostrand, Princeton, NJ, 1960.