Final Report on EPSRC Grant GR/S30450/01
Extensions of Dependent Type Theory – Induction, Interaction, Universes

Principal Investigator: Anton Setzer

1 Background/Context

Dependent Type Theory *(DTT)*[1] can be seen as a uniform programming paradigm, in which one can express both programs and correctness proofs of programs in the same language. Therefore there is not much difference in the techniques used for writing a program and for proving its correctness. DTT allows as well to write more generic and generative programs, i.e. programs which can be instantiated more easily to different situations than programs, which do not make use of dependent types. However, up to now most of the research and development in DTT was focused on the functional core of it, which only allows to write batch programs, i.e. programs which have a fixed number of inputs and outputs without any interaction. The goal of this project is to overcome these restrictions:

(1) We developed and investigated the representation of interactive programs in DTT, i.e. of programs, that continously receive stimuli from the outside world and react in response to those.

(2) We found out how to integrate concepts from object-oriented programming into DTT, which are the first steps towards our vision of an object-oriented programming language based on dependent types. This language would allow to combine the flexibility and generic expressiveness of those two powerful programming paradigms.

(3) We investigated a highly generative data type, the data type of inductive-recursive definitions. This data type has codes for a large variety of data types occurring in standard programming languages, and contains all data types occurring in one of the main versions of DTT, Martin-Löf Type Theory ***(MLTT)***. Using this data type one can write generic programs which analyse and manipulate such data types.

(4) We carried out fundamental research on the limits of MLTT by analysing and measuring the strength of extensions of it and formulating novel extensions, which are substantially stronger than those known before.

We made many **new discoveries:** We discovered the relationship between so called *guarded recursion* and the standard rules for interactive programs. This simplifies the writing of interactive programs. We *reduced coalgebras* (which generalise interactive programs) *to existing MLTT* – this simplifies an analysis of the theory of coalgebras. We discovered a new *game theoretic interpretation* of interactive programs. We discovered how to write in DTT object-oriented programs with self-referential calls, which allows to write *dependently typed component based software* containing loops. We developed a new *data type of partial-recursive functions* with a very generic recursion scheme. And we developed the *Π_3-reflecting universe*, which is the largest predicatively justified extensions of MLTT at present. These were only some of the many new often unexpected insights gained in the course of this project, which significantly expanded our knowledge about the expressive power of DTT.

This project involved many **collaborators**, some of which (most prominently MR and TA) are world leading experts in their area of research:

M Michelbrink (***MM***). RA, employed by this project (for 2 months he held a research position with MH at Bergen, Norway).
R.H. Abdul Rauf (***RHAR***). PhD student, jointly supervised by AS and UB, funded by a grant of the University of Technology MARA, Malaysia. Working on functional programming in C++. Author/coauthor of 4 publications and a PhD thesis close to submission.
C. M. Chuang (***CMC***). PhD student supervised by AS, self-funded. Working on the implementation of interactive programs in the theorem prover Agda.

T. Arai (***TA***). Professor at Kobe, Japan. Collaboration with AS on impredicative proof theory. AS visited TA during two 1-month research visits. The second visit was funded by a grant of TA.
U. Berger (***UB***). Reader at Swansea, proof theorist, coauthor of AS, co-supervisor of RHAR.
P. Dybjer (***PD***). Professor at Chalmers Technical University, Sweden. Coauthor of AS on inductive-

[1] The introduction of abbreviations is marked by writing them in boldface italic.

recursive definitions.
P. Hancock (***PH***). RA at Nottingham. Coauthor of AS, collaboration on interactive programs in DTT.
W. Harwood (***WH***). PhD student of FM, coauthor of AS, collaboration on ordinals and process algebras.
M. Haveraaen (***MH***). Professor at Bergen, Norway. Coauthor of MM in the area of C++ and functional programming.
F. Moller (***FM***). Professor at Swansea. Coauthor of AS, collaboration on ordinals and process algebras.
M. Rathjen (***MR***). Professor, formerly Ohio State University (OSU), USA, now Leeds. Collaboration with AS on impredicative proof theory. Invited AS for a 1 month research visit at OSU, funded by the Mathematical Research Institute of OSU.

2 Key Advances and Supporting Methodology

Interactive Programs in DTT

State dependent IO and Final Coalgebras in DTT. Together with PH we studied in [HS05a,HS05b] the representation of interactive programs in DTT. We developed both a non-state dependent and a state dependent version, and developed the notions of client and server side programs. All versions turned out to be given as weakly final coalgebras of a general form of polynomial functors. We investigated these coalgebras and developed formation/introduction/elimination/equality rules for them. An important result was the fact that guarded induction is nothing but a variant of the introduction rules for weakly final coalgebras.
The **implementation of final coalgebras in standard MLTT** is a result which goes beyond the original proposal. Originally we only wanted to explore the notion of a weakly final coalgebra as an extension of MLTT by additional rules. Surprisingly, we found out, together with MM ([MS05,Mi06b]), that these additional rules are not needed: In original MLTT, extended by uniqueness of identity proofs, one can show the existence of weakly final coalgebras arising from state-dependent interactive programs. We explored as well monad style IO, and proved a generalised version of the monad laws for this operation. All proofs were carried out in the theorem prover Agda, which is based on MLTT.
Interactive programs as games. The technique used in the [MS05,Mi06b] was to represent interactive programs as 2-player games, which are easier to model. In [Mi06a] MM investigated these 2-player games game-theoretically. Programs turned out to be strategies for these games. MM discovered links to the refinement calculus, linear logic, formal topology and process algebra. Interfaces together with the refinement relation build a complete lattice. A program/strategy on a refinement of an interface gives a strategy on the interface. MM defined several operators on interfaces: tensor, par, choice, bang, etc. Every notion can be dualised by interchanging player and opponent. MM developed the notion of a fair strategy on $A \multimap B$, and showed that fair strategies are closed under composition, identity and the application function. MM conjectures that, if one identifies strategies by some kind of behavioural equivalence, one obtains a linear category. All results were shown in intensional MLTT, and all but one were verified in Agda.

[HS05a] PH, AS: Interactive programs and weakly final coalgebras in dependent type theory. In: L. Crosilla, P. Schuster (Eds.): *From sets and types to topology and analysis.* Oxford Logic Guides, Oxford University Press, pp. 115 – 136, 2005. (Refereed proceedings).

[HS05b] PH, AS: Guarded induction and weakly final coalgebras in dependent type theory. In: T. Altenkirch, M. Hofmann, J. Hughes (Eds.): *Dependently typed programming.* Dagstuhl Seminar Proceedings 04381, Dagstuhl, Germany, 2005. http://drops.dagstuhl.de/opus/volltexte/2005/176.

[MS05] MM, AS: *State-dependent IO monads in type theory.* Electronic Notes in Theoretical Computer Science, Elsevier, 122, 127 – 146, 2005.

[Mi06a] MM: Interfaces as Games, Programs as Strategies. In J.-C. Filiatre, C. Paulin, B. Werner (Eds.): *Types for Proofs and Programs.* Springer Lecture Notes in Computer Science 3839, pp. 215 – 231, 2006. (Refereed proceedings).

[Mi06b] MM: *Interfaces as functors, programs as coalgebras: a final coalgebra theorem in intensional type theory.* Theor. Comput. Sci 360, 1:415–439, 2006.

[Mi06c] MM: *A generalisation of Hancock-Setzer Interfaces.* 8 pp. Draft, available via http://www.cs.swan.ac.uk/~csmichel/articles/GInterface-01.dvi, 2006.

Weak Bisimulation Approximants. Using our proof theoretic perspective on coalgebraic structures, we investigated with WH and FM approximations to the bisimulation relation in the context of Basic Parallel Processes (BPP). It is known that for any process algebra weak bisimilarity $\approx$ is equal to its αth approximant $\approx_\alpha$ for some α. The question is to determine the least α s.t. $\approx = \approx_\alpha$. For BPP only a trivial upper bound $\alpha \leq \omega_1^{\mathrm{ck}}$ was known. In [HMS06] we were able to give a long-standing non-trivial upper bound for BPPs for this alpha, namely $\alpha \leq \omega^\omega$. For this we gave a novel proof of Dickson's Lemma and obtained a tight ordinal bound, namely ω^n, on the order type of non-increasing sequences of n-tuples of natural numbers. In [MS07] we gave a geometric interpretation of Dickson's Lemma and determined the exact order type for non-increasing sequences of n-tuples.

[HMS06] WH, FM, AS: Weak Bisimulation Approximants. In: Z. Ésik (Ed.): *Computer Science Logic, 20th International Workshop, CSL 2006*, Springer Lecture Notes in Computer Science 4207, pp. 365 – 379, 2006. (Refereed Proceedings).

[MS07] MM, AS: *Exact Dickson Ordinals.* Draft, 2007.

Object-Oriented Programming (*OOP*) in DTT and Component Based Software

We used the idea of representing objects as interactive programs in order to develop concepts from OOP in DTT. We represented methods, interfaces and the interaction between finitely many objects, including self-referential calls. We were able to overcome the unexpected complications arising from the fact that while making a call to another object, the internal state of an object might change due to calls to the original object. We developed a monad like syntax for developing objects in DTT.

Objects can be considered as components: each component has an incoming and an outgoing interface. Since we allow self-referential calls, we were able not only to deal with arrangements corresponding to combinatorial circuits, but also those containing loops.

This research is published in the following 2 conference proceedings. A journal publication of this research is in preparation.

[Se06a] AS: Object-oriented programming in dependent type theory In: *Proceedings of TFP 2006.* Technical report, Department of Computer Science, University of Nottingham, 2006. (Refereed Proceedings.) http://www.cs.nott.ac.uk/~nhn/TFP2006/TFP2006-Programme.html

[Se07a] AS: Object-oriented programming in dependent type theory. To appear in *Post-symposium Proceedings of TFP 2006.* Series Trends in Functional Programming, Intellect, 2007. (Refereed Post-Proceedings.)

Functional Programming in C++ and Java. As part of our research on the relationship between OOP and DTT we started to investigate the implementation of functional concepts in OOP languages. We developed a direct representation of the λ-calculus in Java using inner classes ([Se03]). In the PhD project of RHAR we developed the representation of the λ-calculus in C++ and proved its correctness. This turned out to be much more complicated because of the lack of inner classes in C++ and the complexity of C++. This research gave rise to the publications and submissions [ABS06,AR06,ABS07a,ABS07b,AR07]. RHAR wrote a tool for translating λ-terms into C++. In order to verify the correctness of our translation, we developed a theoretical model of C++. We believe that this object-oriented model can be used outside this specific application, since it contains an abstract notion of a heap and is therefore capable of representing programs with side effects. We developed an abstract translation function of λ-terms into this model, and then showed that the semantics of λ-terms coincides with the semantics of the translated λ-terms in our theoretical model of C++, showing the correctness of the translation. Finally we considered the representation of lazy concepts in C++.

MM showed together with MH ([HM07]) how to translate the simply typed λ-calculus into C++-templates.

[Se03] AS: Java as a functional programming language. In: H. Geuvers, F. Wiedijk (Eds.): *Types for proofs and programs.* Springer Lecture Notes in Computer Science 2646, pp. 279 – 298, 2003. (Refereed proceedings).

[ABS06] RHAR, UB, AS: Functional Concepts in C++. In: Proceedings of TFP 2006, technical report, Department of Computer Science, University of Nottingham, 2006. (Refereed Proceedings). http://www.cs.nott.ac.uk/~nhn/TFP2006/TFP2006-Programme.html

[AR06] RHAR: Integrating Functional Programming into C++: Implementation and Verification. In: A. Beckmann, U. Berger, B. Löwe, J. Tucker: *Logical Approaches to Computational Barriers.* Report # CSR 7-2006, Department of Computer Science, University of Wales Swansea, pp. 2 – 11, 2006. (Refereed Proceedings). http://www.cs.swansea.ac.uk/reports/2006.html

[ABS07a] RHAR, UB, AS: *Functional Concepts in C++.* To appear in Post-symposium Proceedings of TFP 2006, Series Trends in Functional Programming, Intellect, 2007. (Refereed proceedings).

[ABS07b] RHAR, UB, AS: *A provably correct translation of the λ-calculus into a mathematical model of C++.* 22 pp. Invited submission to J Theo Comp Sys, 2007. (Refereed post proceedings of the conference CiE 2006).

[AR07] RHAR: *Functional Concepts in C++.* PhD thesis, Dept. of Computer Science, University of Wales Swansea, 2007, in preparation.

[HM07] MH, MM: *A direct translation of the simply typed lambda calculus into C++-templates.* Submitted to Journal of Functional Programming, 2007.

Closed Formalisation of Inductive Recursive Definitions (*IRD*s)

Induction-recursion (***IR***) is a definition method in DTT, which extends (generalised) inductive definitions in order to include a large variety of universe constructions. Indexed inductive-recursive definitions (***IIRDs***) generalise IRDs to the simultaneous inductive-recursive definition of indexed sets. All sets occurring in standard MLTT are instances of IIRDs, including universes, many inductive data types, and novel data structures. In [DS03] (which was the most downloaded article of Ann. Pure Appl. Logic in the period January - March 2004) we gave a new compact formalisation of IRDs by modelling them as initial algebras in slice categories. We gave generic formation/introduction/elimination/equality rules, and investigated the relationship of these rules with the principle of the existence of initial algebras for certain endofunctors. We discovered that the external Mahlo universe is a subtheory of IRDs, and obtained using this fact a lower bound for the strength of IRDs.

In [DS06] we extended this theory to a closed formalisation of IIRDs. As a special case we obtained a closed formalisation of indexed inductive definitions. We discovered that there are two different versions of IIRDs: One corresponds to the data-construct in the original version of the proof assistant Agda for MLTT. The second one admits a more general form of an introduction rule, which allows defining of the intensional identity relation. We showed that extensionally the theories of restricted and general IIRDs are equivalent. Finally, we constructed a model of IIRDs in classical set theory extended by a Mahlo cardinal.

[DS03] PD, AS: *Induction-recursion and initial algebras.* Annals of Pure and Applied Logic, 124:1 – 47, 2003.

[DS06] PD, AS: *Indexed induction-recursion.* Journal of Logic and Algebraic Programming 66, 1 - 49, 2006.

Using IIRDs for Representing Partial-Recursive Functions in MLTT.

In [Se06b,Se07c] we investigated how to represent partial recursive functions in MLTT using IIRDs. The representation is based on the approach by Bove and Capretta, which makes use of IIRDs. We showed how to restrict the IIRDs used so that we obtain directly executable partial recursive functions. Then we introduced a data type of partial recursive functions. We showed how to evaluate elements of this data type inside MLTT, and that therefore the functions defined by this data type are in fact partial recursive. The data type formulates a very general schema for defining functions recursively in DTT. The initial version of this data type, for which we introduced an induction principle, had to be expanded, in order to obtain closure under composition. We obtained two versions of this expanded data type, and proved that they define the same set of partial recursive functions. Both versions will be large types. Next we proved a Kleene-style normal form theorem. Using it we showed how to obtain a data type of partial recursive functions which is a small set. Finally, we showed how to define self-evaluation as a partial recursive function. We obtained a version of this evaluation function, which not only computes recursively a result, but as well a proof that the result is correct.

[Se06b] AS: Partial recursive functions in Martin-Löf Type Theory. In: A. Beckmann, U. Berger, B. Löwe, J. Tucker (Eds): *Logical Approaches to Computational Barriers: Conference CiE 2006.* Springer Lecture Notes in Computer Science, 3988, pp. 505 – 515, 2006.

[Se07c] AS: A data type of partial recursive functions in Martin-Löf type theory. 20 pp. Invited submission to J Log Comp. (Post proceedings of the conference CiE 2006), 2007.

Proof Theory of MLTT and the Π_3-Reflecting Universe

In the invited article [Se04] we gave an overview over the current state of the art of proof theory of MLTT. After an overview over the historic development of proof theory, we exhibited the need for constructive theories in order to supplement a proof theoretic analysis. Then we considered the proof theoretic analysis of MLTT with W-type and one microscopic, one full, and one Mahlo universe. Finally we reviewed the concept of IRDs and what is known about its strength.

As part of our investigations into the Π_3-reflecting universe, MM introduced in [Mi06d] a Buchholz-style notation system for the infinitary derivations occurring in the ordinal analysis of $\mathrm{KP} + \Pi_3$-Reflection of MR. This allowed a finitary ordinal analysis and a characterisation of the provably recursive functions of $\mathrm{KP} + \Pi_3$-Reflection.

In [Se06c] we introduced models for universes in MLTT. We considered a simple universe, E. Palmgren's super-universe, and the Mahlo universe. The models were introduced in extensions of KP formalising the existence of a recursively inaccessible, a recursively hyper-inaccessible, and a recursively Mahlo ordinal. This allowed to determined upper bounds for the proof theoretic strength of these theories. In case of simple universes and the Mahlo universe, these bounds have been shown by AS to be sharp.

In [Se07b], we introduced an extension of MLTT, for which we have handwritten notes showing that it has the same proof theoretic strength as Kripke-Platek set theory (KP) extended by one Π_3-reflecting ordinal and finitely many admissibles above it. That means that the proof theoretic strength of this type theory is substantially bigger than that of any previous predicatively justified extensions of MLTT, including the Mahlo universe. The universe is constructed following the principles of ordinal notation systems of strength KP plus one Π_3-reflecting ordinal, therefore extracting key ideas of these notation systems. We introduced a model for this type theory, and determined an upper bound for its proof theoretic strength.

[Se04] AS: *Proof theory of Martin-Löf type theory - An overview.* Mathematiques et Sciences Humaines, 42 année, n^o 165, pp. 59–99, 2004. (Invited journal article).

[Mi06d] MM: *A Buchholz derivation system for the ordinal analysis of KP+* Π_3*-reflection.* J. Symbolic Logic 71, 4: 1237 – 1283, 2006.

[Se06c] AS: Universes in type theory I – Inaccessibles and Mahlo. In: A. Andretta, K. Kearnes, D. Zambella: *Logic Colloquium '04.* Association of Symbolic Logic, Lecture Notes in Logic 29, pp. 123 - 156, 2007. (Invited conference postproceedings.)

[Se07b] AS: *Universes in Type Theory Part II – Autonomous Mahlo and* Π_3*-Reflection.* 32 pp. After revision to appear in J. Sym. Logic, 2007.

3 Project Plan Review

In the proposal we had the following objectives:

- Interactive programs into DTT. We developed state-dependent IO in DTT and generalised this to weakly final coalgebras. While doing this we discovered that weakly final coalgebras arising from IO can be modelled in original MLTT and we found astonishing connections to game theory and linear logic. These discoveries required substantial verification work using Agda, and therefore delayed the implementation of interactive and multithreaded[2] programs, which has now become part of a PhD project with CMC. We have carried out case studies on verifying railway control systems in Agda, and verified sorting algorithms in Agda. A project on verifying algorithms from Lynch's book[3] in Agda has just started. As an additional not initial intended topic we studied the ordinal theory of process algebras (BPPs).
- **OOP in DTT.** We formalised concepts from OOP in DTT, and discovered a much richer structure than expected. We have carried out small case studies, but type checking in Agda is still work in progress. We expanded this topic by studying the implementation of functional programming in OOP, which resulted in a C++ program carrying out the translation, and in 6 publications/submissions.
- **Closed Formalisation of IRDs.** We did a much more intensive and thorough study of IRDs, and found many applications. As a case study on applications in generic programming we explored partial-recursive functions in DTT, which became a major research project on its own.
- **The Π_3-reflecting universe and extensions.** We wrote a series of two articles in which we developed models and upper bounds for the strength of many extensions of MLTT, including the Mahlo and the Π_3-reflecting universe. Since we were invited to write a major survey article on the proof theory of MLTT, we spent more time on this, and therefore we were able to write down the lower bound for the strength of the Π_3-reflecting universe only as handwritten notes. In extension of the original objectives, MM published a mayor result on the provable recursive functions in KP + Π_3-reflection. As announced in the proposal, there was a high risk of not finding a generic extension of IRDs to cover the Mahlo universe, and despite many attempts this turned out to be true.

4 Dissemination of Results

The published output consists of 6 refereed journal articles (all reviewed), 12 conference proceedings (5 articles are awaiting publication, with final proofs received and corrected; 5 have been reviewed or are about to be reviewed, 2 will be reviewed, if the standard practice of Mathematical Reviews is kept). Furthermore, we have reviewed 21 articles in Zentralblatt MATH and Mathematical Reviews. Under submission are 2 invited journal publications; 1 journal publication accepted subject to corrections; 1 other journal publication with the international coauthor MH.[4] Furthermore, one PhD thesis is to be submitted by Sept 2007,

[2] Based on PH, AS: The IO monad in dependent type theory. In: Electronic Proceedings of Workshop on dependent type theory, http://www.md.chalmers.se/Cs/Research/Semantics/APPSEM/dtp99/proceedings.html, 1999.

[3] N Lynch: *Distributed Algorithms.* Morgan Kaufmann, 1996.

[4] In the table summarising publication details we have included all articles which are "to appear" and noted as reviewed all articles which are expected to be reviewed, and excluded all articles which are submitted but not yet accepted in final form.

and 2 drafts were written. Our lecture notes on interactive theorem proving[5] form now part of the documentation of the theorem prover Agda.[6]

The results were presented at 1 spring school; 2 invited plenary/special session talks at major conferences; 2 invited international and 1 invited national workshop; 1 Oberwolfach and 1 Dagstuhl workshop talk; 8 international conferences (of which 4 were held in UK but were directed at an international audience); 1 national conference; 3 one month intercontinental research visits with associated seminar talks; 6 international, 7 national and 7 local seminar talks. Apart from the associated visits for the above we participated at 5 international conferences (of which 2 were of international flavour but held in UK), and 1 national conference.

5 Research Impact and Benefits to Society

The research in this project had significant impact on current international research in the area of type theory, formal methods, category theory, and OOP.

There are currently discussions on implementing interactive programs in the type theoretic theorem prover Agda (the industrial research centre AIST in Osaka, Japan has recently joined Chalmers in developing Agda), and this has been influenced by the research of this project on interactive programs in DTT. AS is working with CM on implementing directly our theory of interactive programs in Agda. The distinction between restricted and general IIRDs has found its way as well into Agda by having both “data” and “idata” types.

We met a lot of interest by type theorists on our research on OOP in DTT and were invited to have a joint proposal with Z. Luo (Royal Holloway) on this topic. We hope that this research will expand further and on the long run result in an OOP language based on dependent types. Our research on IRDs has been taken up by PH and N. Ghani who invited AS to submit a joint grant proposal on “Categorical foundations of induction recursion, and its practical ramifications, mainly in final coalgebras”.

As reflected by the invitations to visit key researchers in US and Japan, our proof theoretic results on the Π_3-reflecting universe have raised big hopes about getting a better understanding of ordinal theoretic impredicative proof theory, and of obtaining a fresh impetus on research in this area.

This project helped to make Swansea become a subsite of the EU network TYPES. Furthermore, AS became over the course of this project an editor for the journal Logical Methods in Computer Science.

[5] http://www.cs.swan.ac.uk/∼csetzer/lectures/intertheo/
[6] http://unit.aist.go.jp/cvs/Agda/

6 Explanation of Expenditure

We had 3 research visits of 1 month each. Two were funded by our partners, and one visit to Japan turned out to be (because of cheap accommodation offered by Kobe University) much cheaper than expected. Furthermore, 3 international conferences were held in Swansea (with waived conference fees) and we got invitations to many international and national conferences. Because of this we did not use as much of our travel budget as originally intended while still presenting and discussing the results of this project at an international scale. We originally intended to buy a large and expensive monitor, but realised that a much cheaper solution is to use 2 monitors instead (which provides in fact an even bigger screen). This again saved some money originally intended for equipment.

After contacting EPSRC, we used the money saved in order to fund extra months for our RA. Those extra months were highly productive (5 articles and drafts were finalised or written), therefore this change of budget substantially increased the output of this project.

7 Further Research and Dissemination Activities

Apart from the invitation to two joint grant proposals mentioned in Section 5 we have submitted an EPSRC grant proposal with Paul Taylor (London) on Abstract Stone Duality II (£289 K) with the goal of developing ASD into a type theory usable by ordinary mathematicians that automatically gives them the topology on and programs for their constructions.

We are preparing with E. Palmgren and T. Coquand a 2 volume monograph on type theory and constructive mathematics. We have started a collaboration with the director of AIST, Osaka, Japan, Y. Kinoshita, on developing documentation for the theorem prover Agda, which will hopefully result in another research monograph, partly based on our lecture notes on interactive theorem proving. Furthermore, we are working on converting all drafts mentioned in this proposal into conference or journal articles.

This year AS is an invited plenary speaker at the conference “1907 – 2007, cent ans d’intuitionnisme”, at Cerisy, France 2007, and will give a talk at the conference TYPES 2007 at Udine, Italy. AS is as well organising with colleagues the proof theory conference PCC 2006 in Swansea, and will be co-organising the British Logic Colloquium in Swansea 2009. Both events will be excellent opportunities to stimulate research in areas related to this project.

On the proof theoretic side the main project of AS is to write a journal publication containing the lower bounds for the proof theoretic strength of the Π_3-reflecting universe, and then to formulate and determine the proof theoretic strength of the Π_n-reflecting, Π_α-reflecting, and the Π^1_1-reflecting universe.