Contents

Chapter 1

The StarDust Constellation

The StarDust Constellation is a small group of "projects" based in Github repositories.

They are concerned with *hybrid deductive intelligence* and its applications.

High-level descriptions of the constellation and its members are provided in hypertext wikis associated with each of the four repositories, which are organised so as to combine into a printable PDF. This general pattern is to present

- the aims of the project or subproject
- the philosophy underpinning the proposed approach (using the term philosophy here broadly)
- the architecture of the proposed solution
- the strategy for refining and implementing the architecture

This StarDust repo provides the highest level view, gives a first indication of the role of the other three sub-projects and how they fit

together, and addresses issues which do not naturally fit into any one of those sub-projects.

- The Aims of StarDust
- Philosophical Fundamentals
- The Architecture
- The Strategy

1.1 The Aims of StarDust

The aim is to contribute to the realisation of *hybrid deductive intelligence* and its applications, and ultimately to *universal fulfilment*.

The whole of mathematics, particularly the theoretical development of mathematics, reaching its most abstract and abstruse parts, though most likely addressing applicable mathematics first, is regarded as a core competence for the level of intelligence sought. Intelligence of a this order is beyond the present state-of-the-art for completely automated reasoning. And so, in this project we envisage, as in the field of Interactive Theorem Proving, that less trivial results will be realised (at least in the early stages) by man and machine working together, to which arrangement I refer by the term *hybrid intelligence*. *Hybrid* intelligence of course, we already have, that's how mathematics is done, the challenge is enable the machines to contribute at higher levels, and to integrate the competence into more general problem solving capabilities.

Many believe that the pace of change is growing and that in relatively brief timescales radical advances in artificial intelligence will be secured. Because these projects are conceived of as contributing to a technological domain which is expected to be radically transformed,

its future relevance depends upon how well it is aligned, not so much with todays information technology, but with the very different environment into which we will soon be thrust. For this reason I seek, in describing the project, to place it in the context of that future as I see it, and to make a connection between the details of the proposed architecture and the character of that future.

Context for StarDust

Chapter 2

context for StarDust

2.0.1 Hybrid Deductive Intelligence

Think of this as a variant on *Artificial General Intelligence* in the following ways:

- hybrid,
  in lots of ways combining:
  - *meat* and *machine*,
  - *explicit propositional knowledge* with *intuitive knowledge or capabilities* embedded in deep neural nets
  - *general foundational knowledge*, and highly specialised *domain specific knowledge*
  - *intuitive inference* by deep neural nets, with high level *heuristically guided search*
- deductive
  a key aspect of the architectural perspective proposed is its en-

gagement with a paradigm shift from data and computation to propositions and inference. This is a key enabler for achieving trust and precision in systems whose effectiveness is underpinned by neural and other innovative information processing architectures.

- intelligence
  Contemporary advances in artificial intelligence are mainly oriented to low level perceptual or intuitive judgements, and the kind of knowledge that is built and exploited by neural net-like architectures in addressing these problems. These are sometimes realised by supervised or unsupervised learning. Even when supervised learning is used, the supervision is very basic, nothing like the kind of explanation or teaching which is normally considered essential for effective learning by our own brains of the kinds of material which are taught even in secondary let alone higher education.

2.0.2 Universal Fulfilment

The idea is that everyone should be able to realise their own *true potential*, and that in the future, with the automation of the means of producing physical and intellectual goods, this will be almost completely realisable. The main limits in the realisation of universal fulfilment will be incompatibilities between the needs and desires of different individuals or groups.

Not everyone will see this goal as sufficient, or even desirable. Putting aside for the moment those who doubt its desirability, and addressing those who consider it insufficient, their own desire for more (than universal fulfilment) will form a part of the notion of universal fulfilment

(assuming their fulfilment depended upon it), and their greater ambitions will therefore be incorporated into the apparently lesser goal.

Thus, knowing that there will be many who would not feel personally fulfilled as a contributing member of a society with no greater ambition than its own well-being, we also consider as incorporated in this aim the attainment and progression of great questions like "what's it all about?", "Why are we here?".

2.1 Philosophical Fundamentals

As the StarDust project is first being formulated, there is enormous interest not only academic but commercial in "artificial intelligence" as *Deep Learning* effected by simulations of neural nets. Not for the first time, the idea that AI can best be achieved by study of and replication of the methods used in the human brain, has come to preponderate, to the extent now that the more formal approaches, often based in or related to formal logical notations, are scarcely mentioned, save to underline what researchers and application developers are not doing.

I believe this to be unfortunate. Most of the ways in which we have used computers are very different to the ways in which the human brain works, and there is every reason to think, not only that this will continue into the future, but also that methods quite dissimilar to neural nets (even Von Neumann Architectures) can make substantive contributions to the effectiveness with which we address the most difficult intellectual problems.

Be that as it may, it is not my purpose here, or that of the StarDust projects to advance that thesis.

Even if we seek only to re-implement and scale up human-like cogni-

tion, the capabilities which are addressed by StarDust are worthwhile applications of that kind of capability, and the possibility that addressing these applications in advance of the realisation of artificial general intelligence (AGI) might accelerate achievement of AGI may be thought of as a bonus extra.

The Development of Language and The Representation of Knowledge

Current AI orthodoxy is focussed on learning from sensory data to realise "knowledge' represented as synaptic weights in simulations or simplified models of neural nets. The growth of higher knowledge depends upon the development of more sophisticated kinds of language, which are suitable for sharing knowledge, collaboration on its development and exploitation, and its application to reasoning about the environment and our interventions into the environment.

Communication has co-evolved with brain capacity. In the most primitive organisms we have communication by chemical signals, which serves to coordinate the behavioural adaptation of bacteria to their environment. I

Three Kinds of Proposition

The Deductive Paradigm Shift

2.1.1 The Development of Language and The Representation of Knowledge

Current AI orthodoxy is focussed on learning from sensory data to realise "knowledge' represented as synaptic weights in simulations or simplified models of neural nets. The growth of higher knowledge depends upon the development of more sophisticated kinds of language,

which are suitable for sharing knowledge, collaboration on its development and exploitation, and its application to reasoning about the environment and our interventions into the environment.

2.1.2 Three Kinds of Proposition

The philosopher David Hume (among others) drew two important distinctions between kinds of propositions, which provide important insights into how we can develop, organise and create knowledge. For this reason these distinctions form a philosophical basis for the division of StarDust into sub-projects.

The divisions are not uncontroversial, it is not universally agreed that these distinctions are clear, but they are nevertheless built into the terminology and structure of StarDust.

Hume's first distinction is found in his observation that we cannot derive an "ought" from an "is", which we generalise into the idea that moral or evaluative judgements are not logically deducible from purely descriptive claims. One needs, in addition to purely descriptive information, some information about *values* or *ethics* in order to form specific evaluative judgements. A similar distinction is found in the philosophy of G.E. Moore who coined the term "naturalistic fallacy" for the idea that moral judgments might be definable in naturalistic terms.

Hume's second distinction is concerned exclusively with non-evaluative propositions, which Hume observes come in two kinds, those which express "relations between ideas" (which we will call *logical* or *analytic* propositions) and those which convey some "matter of fact", by which "empirical fact" or fact about the world might be understood. This single distinction in Hume was refined by Kant into three separate

distinctions, those between necessary and contingent propositions, between analytic and synthetic sentences, and that between knowledge *a priori* and *a posteriori*. These distinctions were implicitly held by Hume to be coextensive, the description of his distinction combines vocabulary suggestive of each of the three distinctions made by Kant, but were considered by Kant and many other subsequent and contemporary philosophers to be distinct. The view that these distinctions are coextensive is typical of *positivist* philosophies, and was revived in the philosophy of Rudolf Carnap, aspects of which may be seen to be reflected in the structure of the StarDust projects.

The importance of these distinctions (particularly for this project) lies in the different methods which are appropriate for establishing propositions of these kinds. Thus, the (possibly *naive*) view that there is a single trichotomy here leads to the following prescriptions as to how the various kinds of proposition may be established:

- *analytic* propositions can be established *a priori* (typically by logical proof) given only a sufficiently good knowledge of the meaning of the language in which they are expressed.
- *synthetic* propositions cannot be logically proven other than by derivation from other empirical propositions. They say something about how they world is, and one cannot establish their truth without observation or experiment (and even then they may not be deducible conclusively, but are likely to depend on hypothetical generalisations).
- *evaluative* propositions depend not only on *analytic* and *synthetic* claims, but also on some kind of value system

The StarDust project may appear to depend upon the Hume/Carnap thesis of the well-definedness of these distinctions and the coextensiveness of the non-evaluative dichotomies, particularly because

the three sub-projects are scoped in apparently corresponding ways. However, this subdivision of the projects is based on purely methodological criteria which correspond to the functionality of the software belonging to each sub-project, and the precise relationship with the other aspects of the distinctions is not crucial.

2.2 The Architecture of StarDust

The project is intended to develop open source software for deployment in delivering certain cognitive services. The services come in three layers corresponding to the three associated sub-projects which are:

- HoLoTruth
  concerned the establishment and management of Higher Order LOgical Truth, providing Proof As A Service.
- HoLoMod
  concerned with empirical and other models in HOL, with Smart Oracles.
- HoLoVal
  Value refinement and implementation.

Chapter 3

Previously

3.1 The Ambition

The Cognitive Cosmos

The ambition here is broad.

- The Deductive Paradigm Shift
  To understand the proposal you must first engaged with the idea of a major paradigm shift in the way that computers process information.
- Formal Mathematics (Formal-Mathematics)
  The formalisation and automation of deductive mathematics (encompassing and extending the principal aims of QED and Calculemus)
- Deductive Cloud Not just mathematics, but science, engineering and technology can be formalised as mathematical models in Higher Order Logic. Our knowledge of the world can be rendered

in mathematics and made more amenable to rigorous deduction than it would otherwise be, a "semantic web" in abstract HOL (with diverse concrete presentations)

- General Deductive Intelligence Our architecture is intended to provide a suitable context for collaboration between man and machine, each learning from the other, learning and evolving in a competitive autonomous marketplace
- Deductive Smart Contracts
  Deductive proof delivers trustworthy results, and the proposed architecture delivers graduated levels of confidence depending on how trustworthy the various premises and deductive agents involved in the proof are. This may be of value in business processes even where the complexity of reasoning falls well below that required in mathematical research. We now see an impetus for the automation of business processes through so-called "smart contracts" and believe that declarative languages presenting models in abstract HOL may be advantageous in these applications.

Its not envisages that all of this would be happening under this one project. Think in terms of a core and an ecosystem. The sketch above is of the principal parts of the whole. Anything sufficiently substantial and self contained will become a subproject, or a related project. DA-Hol will begin as rhetoric, graduate through architectural exposition and high level design, detailed design of key standards and then prototype and further develop key components.

For further detail on how this breaks down, follow the links (only present once they have gone blue) to the four sub-topics and then see the architectural sketch in this manifesto, after perusing the philosophical underpinnings.

hype cycles

3.1.1 The Cognitive Cosmos

- **cosmic hybrid deductive hyper-intelligence** - a short explanation of the terms
  - cosmic
    think of this a single entity growing outwards across a large part of the milky way, approaching 100,000 light years across
  - hybrid
    just as the brain is not homogeneous having a large cortex and various more central structures, this cosmic intelligence has an expanding out region spreading rapidly out across the galaxy, and probably inanimate, wrapped around smaller regions with intelligent living species present only in more slowly growing inner regions. The intellectual power of the our regions is there only to progress the agenda evolved into the inner regions, capable of stunning independent intellectual accomplishments.
  - deductive
    our project is primarily concerned with formal knowledge representation, deductive reasoning, hard science and its engineering applications, and more generally in the intelligent application of deduction to facilitate all aspects of human enterprise
  - hyper-intelligent
    we are thinking here well beyond "the singularity" if this is construed as the point at which artificial intelligence surpasses that of homo sapiens, which might well happen this century, whereas we consider timescales of hundreds of thousands of years, justifying an expectation of hyper-intelligence.

- Scale and Principles
- Integrating Smart and Scruff AI

3.1.2 The Deductive Paradigm Shift

To engage with it you must first of all contemplate the possibility of a major paradigm shift in information processing, from *computation* to *inference*, taking place on a global scale. In the context of that *possibility*, the aim of this project is to position *Higher Order Logic* (HOL), and more specifically that particular formulation of HOL which has its roots in Russell and Church and was formulated by Mike Gordon, as a universal foundation enabling this paradigm shift. What it means to call HOL a "universal foundation", and the grounds for supposing it so, are not our present concern, but will be addressed later in the *philosophical* underpinnings of the proposal.

Computer support for formal deduction has now been under development for more than half a century, and the use of these methods continues to be, in most application domains, highly labour intensive. Its extensive use with currently available technologies is improbable. Its use, even by that group of academics who have been most concerned with logical proofs, those conducting research in pure mathematics, has been very limited indeed. The paradigm shift I anticipate is therefore predicated on substantial advances in the deductive capabilities of machine intelligence, and the architecture proposed aims (inter alia) to provide an incubator for that kind of machine intelligence.

I am not myself an expert in machine intelligence, and in believing that this is the moment for a project which anticipates and provides a home for deductive machine intelligence I am not claiming that I have any special recipe for its attainment (though several aspects of

this proposal are designed with that in mind). It is the present state, rate of advancement and scale of commitment to machine intelligence (independently of anything which might take place under this project) that gives me belief that progress of machine intelligence in this special yet general domain, not yet conspicuous, may be soon upon us.

Of the various factors contributing to this present state, it is advances in specialised hardware which may be the most important: - by bringing machines up to a level of performance in "deep learning" which yields effective commercial and industrial applications - in promising rapid further reductions in the cost of training, as specialised hardware performance achieves a better-than-Moore's-law improvement trajectory

Commercial applications of deep learning are now so pervasive that no organisation can afford to ignore them for fear of disruption by competitors who do not. This commercial imperative is fuelling the advances in hardware which will continue to multiply the domains of effective application.

Nevertheless, those working on the automation of proof are likely to look at the achievements of deep learning and be skeptical about whether they come anywhere near delivering in machines the kind of intelligence which we see in mathematicians, and which may be thought prerequisite for the mooted paradigm shift. We see many examples of the kind of intuitive pattern-matching capabilities, such as sound and image recognition, which were not thought of as requiring intelligence at all until AI research showed how hard they were to mechanise. Their present realisation may be thought due rather to the scale of raw compute now available than to the (undoubted) sophistication of the algorithms now available.

There are however, examples which fall outside this pattern and which may provide a model for some not-so-distant achievement in deductive

intelligence. Well known among these is the AlphaGo system. This champion Go player is an application of deep neural nets. But these deep neural nets provide heuristics which guide the trajectory and evaluate positions in a forward search which is itself programmed rather than realised by learning in a neural net. Thus we obtain an intelligence which comes closer to reasoning about the problems it addresses than many applications of deep learning, and an approach which seems potentially adaptable to deductive applications.

Further elaboration of the ambition which inspires this project is presented in four parts, which are related to the structure of knowledge and the machinery for its exploitation, considering firstly the logico-mathematical machinery with which we construct scientific models of the world about us and the detailed engineering models which support technology development and industrial and commercial applications. Mechanisms for incubation of deductive intelligence are mooted, and the ways in which this deductive intelligence, together with data from sensors and other oracular sources feed inference processes enabling deductive 'smart contracts' and other varieties of deductive business process automation.

- How Deduction Trumps Computation

3.1.2.1 How Deduction Trumps Computation

Before considering the question whether logical deduction could be placed centre stage in an all encompassing "Intelligent Deductive Web" you may want to know why anyone would want to do that. Surely this idea smacks of GOFAI (Good Old Fashioned AI) and is put forward at a time when AI as Deep Learning in Neural Nets is rapidly taking over the world.

Neural nets don't do deduction, of course they could, but its not one of their fortes. They do consume huge computational resources. So to talk, as I am here, of a paradigm shift away from computation toward deduction seems to be flying in the face of progress.

However, deduction does not exclude computation, computation is one of the methods which are used in proof. What a deductive paradigm does is to add an extra layer of meaning, so that the significance of the computations is known, and their effective exploitation can be more completely automated.

An important (some might say the most important) feature of the use of Deep Learning is the shift from programming a computer to solve a problem, to the computer learning for itself how to solve the problem. The use of deductive logic allows us to combine this kind of un-programmed functionality with high levels of trust.

Its worth looking at how this works.

Consider the application of deep learning in medicine. Powerful computers using deep learning can assimilate enormous volumes of medical knowledge and use this to good effect in suggesting diagnoses and offering references to pertinent literature. To be effective and valuable, such a system does not need to be highly reliable. It makes suggestions and provides information, a qualified and experienced medical consultant will make the decisions. The machine suggests solutions, the consultant validates the suggestions.

There is a general recipe here for achieving both high performance and high levels of trust, which is particularly valuable in problem domains where validation can be undertaken automatically. Mathematics is an example of such a domain. Mathematicians can guess general mathematical theorems using their deep knowledge of the particular mathematical subject matter, but often their guesses will prove to be

mistaken. Deductive proof has been adopted as a method of ensuring the truth of mathematical hypotheses before they accepted into the body of accepted theory. The nice thing about proofs is that they are an effective check for the truth of a theorem, and the correctness of a proof is itself either mechanically checkable (in the case of a formal proof) or at least, substantially easier and more reliably checked than unsupported hypotheses.

Thus, by combining deep learning with logical deduction we have the prospect of combining intelligence and high degrees of reliability, in a context in which problems are solved without programming.

Lets consider an example far away from mathematics in which the combination of un-programmed functionality and high levels of trust is important. Consider the construction of a smart building as the Internet-of-Things advances. Let suppose that the building has been designed in detail be a firm of architects, and a contract is placed with a construction company to supply a building to that specification. The design incorporates a variety of sensors throughout the building which will be essential to the efficient operation of the completed building, but will also continuously supply data about the progress of the construction.

The contract includes a detailed schedule of the stages in the construction, when they should be completed, and what payments are due on completion. Using data from the sensors (supplemented by reports from inspectors), the deductive cloud continuously monitors the progress of the construction, and will automatically deduce completion of a stage when all the necessary compliance conditions have been met. It may then report compliance to a smart contract running on some blockchain which triggers payment to the contractor. There is no need here to write algorithms to determine compliance with the requirements for each stage of the contract, this will be undertaken

deductively from the detailed terms of the contract.

The deductive involvement can be pushed back further. Once the detailed architectural design is in place, there is a planning activity which plays into the details of the construction contract. This planning activity an itself be undertaken deductively working from the detailed design.

Pushing yet further back, if the detailed architecture is preceded by a careful formulation of the key requirements which the building must satisfy, intelligent deduction can validate that the proposed design will meet those requirements, and might provide support to the architects in the design process, or even largely automate the design, so that the role of the architects becomes one of requirements specification, perhaps being more involved in the aesthetic aspects of the design.

3.1.3 Formal Mathematics

The principal features here are:

- Full support for formal derivation and application of "classical" mathematics, either in the *abstract* simple type theory (STT) or in Higher Order Set theory (HOST), via STT with a strong axiom of infinity (or an axiomatic Higher Order Set Theory). Congenial concrete presentations of mathematical syntax do not belong to the abstract core.
- To achieve computational efficiency HOL will be both the object language and the meta-language, and the logical kernels (there may be more than one) may evaluate computable expressions in either the meta- or the object-language as part of the proof/computation process.

- If more exotic type systems, classical or constructive, are desired, it is recommended that embeddings are used, though, since there is no native concrete syntax, such embedded concrete languages are at a par with HOL itself.

I think of DA-Hol as a 21st Century sequel to the QED project, though the proposal is much broader in scope. I do therefore regard it as a core requirement that good support for mathematics is provided.

However, my own sense of priorities in this may be quite different to that of some, or perhaps even many other QED protagonists, so I will clarify this here, and spell out the consequences these differences have for the core technologies proposed.

Though keen to support mathematics, I am more interested in the applications of mathematics than I am in its further development. I am of the opinion that engineers are more likely to make significant use of proof technology than academic mathematicians who further develop mathematical theories. Furthermore, I believe that even for pure mathematicians, proof technologies will only be adopted when then make it possible to achieve results which could not have been achieved without them, and that they will then be transforming the way mathematical theories are developed, not providing perfect support for continuing to do mathematics in the way that it has always been done (and of course, there is no such way, the practice of mathematics has continuously evolved, sometimes faster, sometimes slower).

It is very hard to anticipate what kinds of foundation system, if any, may ultimately find favour among this kind of professional mathematician. Most of them aren't really interested, some of them think they don't need a foundation at all, much less a formal one, and when foundational ideas become relatively fashionable among mathematicians, they are likely (as, I suggest, in the case of HOTT), to be even less acceptable to the majority of mathematicians as a working environ-

ment than the traditional ones which they have been happy to mostly ignore.

We have no time to debate foundations, there is an opportunity in the state of Machine Learning today to achieve a step function in the proficiency of machine mathematics and we should seize the initiative today with what we have. This sentiment is I think underpinned by the enduring popularity of Mike Gordan's HOL, despite three decades of debate about the merits of the simple type system.

3.1.3.1 Deductive Cloud

The idea presented here as the *Deductive Cloud* is analogous to the W3C "Semantic Web" but strictly formal and deductive.

As I understand it, the motivation of the "Semantic Web" is to enable web search which is sensitive to meanings, not confined to seeking syntactic features of web documents. The Semantic Web initiative sought to realise this by defining special markup languages which could be used to annotate web documents giving information about their content. This is nn enabler for *intelligent* search, an application of machine intelligence, and one imagines it as working primarily in the realm of *common sense*, in the way in which a human being might search the web, but rather more rapidly.

DA-Hol counterposes and aims to provide infrastructure for intelligent deductive inference (which encompasses search) in a strictly formal context, and it is this strictly formal counterpart to the semantic web which is called here the "Deductive Cloud".

There are two main faces to this.

- The formalisation of science, engineering, and other domains of

knowledge, by manual encoding of the relevant theories as formal models.
It should be noted here that the proposal is that empirical theories are introduced in much the same manner as mathematical ones, by conservative extension, aka definition. The theories are not asserted, they are just defined. When applied, a definition is given using these defined scientific terms of some hypothetical (or actual, it makes no difference) physical situation, and theorems are derived which establish, for example, whether that system satisfies some specification of how it is intended to behave. All this is purely mathematical, no claim about the physical world is made, but results are obtained which can be interpreted as conditional observations about the physical world.

- ??

3.1.4 General Deductive Intelligence

Though the basic machinery envisaged by this project is intended to contribute to the use of formal deduction, in its present state, it is oriented to facilitating and exploiting the advancement towards deductive machine intelligence.

First let me explain how I am using this term 'deductive intelligence'.

There was a time, many decades ago, when first-order theorem proving was the dominant paradigm for research into general AI. This was prevalent in the 1970's, when advances in theorem proving methods, notably Robinson's resolution and further developements of that method, stimulated a great deal of optimism. When these methods failed to deliver the advances sought there was a parting of ways between the AI community and the theorem proving community, and formal methods, the desire to formally verify computer systems soft-

ware and hardware, became the dominant driver of research into the automation of formal deductive proof, and a more modest expectation for how much machines could be expected to achieve was reflected in the development of interactive proof tools intended to enable human beings with machine assistance to tackle the larger problems perceived as beyond the reach of complete proof automation.

This kind of *formal* problem then became less significant in AI research, which became more concerned, for example, with perceptual problems, and, even for AI 'logicists' such as McCarthy, with replicating common sense reasoning rather than formal mathematical reasoning or its applications to science and engineering.

The 'deductive intelligence' which is the concern of this project is not aimed at common sense reasoning or at the solution of everyday problems. It is concered with strictly formal reasoning applied in a priori sciences such as mathematics and software programming, and in empirical sciences and engineering, through the use of logical and mathematical models of the physical systems about which reasoning is required. This kind of reasoning is intended ultimately to automate or partially automate engineering design, and the solution of other real-world problems through automated deduction using mathematical models.

The desire is to provide an architecture which can be used to advance this kind of formal modelling in default of machine intelligence in much the same way as it is conducted at present using interactive theorem provers with a patchwork of proof automation capabilities, but whicn provides a context in which more powerful machine intelligence can contribute as it evolves, and which is conducive to that evolution.

Certain features of the intended architecture can be touched upon here (and expanded later in the architectural exposition) as intended to facilitate and exploit deductive intelligence.

[...]

3.1.5 Deductive Smart Contracts

When considering the potential for some new technology it is common to ask "What is the killer application?".

I have thought about the long term prospects for proof technology for many years, and for most of that time I have considered that the "killer app" was design automation. Ultimately, once AI advanced far enough, we would not need to implement a design (and hope to formally verify that it met the requirements). We would write a specification of the required system, and the machine (possibly with a bit of help and guidance) would construct a formally verified solution.

This is a part of the ambition for this project, but of course, its a long way down the track, and I now believe that there are possible killer applications which are simpler and closer.

Proof is about trust. Trust is achieved partly from using simple languages with precisely defined semantics, relative to which formal proof rules can be unambiguously prescribed which enable true sentences to be established conclusively.

We are moving into a world in which business processes are automated through so-called 'smart contracts'. There is a lot of hype around crypto-currencies and distributed ledgers at present, and I am myself a qualified skeptic about whether smart contract is a proper name for this kind of process automation, and whether these contracts should be implemented on public blockchains. I am not sceptical about the more general idea that business processes, including of course asset transfers, will be increasingly automated.

When business processes are automated, whether by block-chain resident smart contracts or by some other means, the steps in the process will often be triggered by events in the real world, information about which is injected into the system by 'oracles'. An oracle is some device or person which feeds in a fact about some state of the world or some event which has happened. If contract or business process involves very complex real world activities, then when formally specified a stage in the process may be completed by a very complex combination of events reported by such oracles. If the contract is formally specified in some language which is essentially syntax over abstract HOL, then the completion of the stage will be deducible from the reports of the relevant oracles.

The normal conception of 'smart contract' is algorithmic, and like all algorithms these may be inscrutable. A high-level declarative specification of such a 'contract' could be written using some approprate notation embedded into HOL, and the inference from the oracular inputs to the completion of some stage in the contracct would be relatively simple and easily automated.

Though initial applications might be simple, the placement of this capability in the context of a deductive cloud pvaes the way for more complex processes to be specified and automated, involving more sophisticated inferences from the mass of information in the relevant parts of the cloud.

3.1.6 Hype Cycles

I like to think, even speculate, about "the big picture" and DA-Hol is one big part of an even bigger story I am painting. In this I connect DA-Hol with some ideas which may be thought now near a peak of un-realistic expectation. I think here of *Deep Learning*, at present the

most celebrated face of machine intelligence, and "the blockchain", enabling technology for cypto-currencies and now touted as central to a complete transformation of financial services, property transactions and business processes more generally.

I'd like here to say a few words about my own assessment of these technologies, how great their potential contribution to DA-Hol might be, and to what extent my belief in the merits of this proposal depends upon the soundness of these connected ideas. This might help the reader form an opinion about how well grounded this proposal is, and just how far into the clouds my head protrudes.

3.1.6.1 Deep Learning and Machine Intelligence

I have a long standing conviction that genuine machine intelligence will eventually come, and by that I mean that machines will eventually be able to attain the levels of proficiency in constructing rigorous proofs of difficult mathematical theorems that we associate with the best professional mathematicians. I also doubt that mathematicians will make much use of proof technology until machine intelligence approaches that point, though possibly new generations of mathematicians might turn out more positively inclined to the technology.

I have watched from a distance several great hype-cycles in AI, when expectations have been raised by some new achievement which in the end has proved more limited than it first seemed.

Do I think "Deep Learning" yet another hype-cycle, or has the problem now been cracked?

Well I don't have any doubt that Deep Learning is delivering commercially valuable capabilities, that a worthwhile number of the present avalanche of applications will pay off. Are we anywhere near achieving

the kind of machine intelligence which would be needed for a significant impact on the automation of deductive reasoning? I don't see it yet. But I do see a model in the AlphaGo project, for a way of using deep learning which might well pay off in interactive or automatic theorem proving, a combination of deep learning and forward search. Presumably, a pure neural net solution could accomplish this kind of effect, for we humans do it that way, but that's not how the Deep Mind team did it, and its probably not the way theorem proving will be cracked.

So you may see here, that with some reservations I am optimistic about the prospects for applications of deep learning in theorem proving. However, the project I propose here does not entirely depend on success in machine intelligence. I do think that the range and scale of application which I think in terms of does depend on it, but that the architecture I outline will be valuable before such advances are secured, and it is intended to provide an environment conducive for the evolution of intelligent deductive automation.

3.1.6.2 BlockChain and Smart Contracts

My enthusiasm for blockchain technologies is much more heavily qualified, and these qualifications need to be spelled out.

[...]

3.2 The Architecture

The proposed cognitive cosmos may be thought of as formed in three layers.

Beneath this top level the enterprise is structured into three layers, which correspond to different kinds of knowledge, with correspondingly different ways of establishing and exploiting knowledge. Its best to describe these from the bottom up, and we give here only the briefest descriptions with links to the separate wiki for each layer.

- HoLoTruth The most fundamental layer in this structure is concerned with knowledge *a priori*, by which is meant knowledge, such as that of logic and mathematics, which tells us nothing about, and therefore can be established and used without reference to the state of the observable physical world.
- HoLoMod The next layer is concerned with knowledge *a posteriori*, i.e. obtained from observations of and experiments in the physical world around us. This includes formal accounts of hard science, and engineering and many other matters, and it formalisation is through logical or mathematical models of the physical world. This layer depends upon the *a priori* knowledge developed in the lower *a priori* level.
- HoLoVal The final layer goes beyond descriptive knowledge and is concerned with values, morals, with our choices about what kind of world we want to live in and the ways in which we can realise those choices.
- StarDust is the uppermost layer in this stack and one way of thinking about this is that it goes beyond the particular ideals we have that belong to *DA-Value*. Technology, and the automation which it delivers, provides quality of life, and it is plausible that at some stage (probably not in the near future, but likely long before the time periods we consider in *StarDust*) a good synthesis of the diverse values of each of us in a pluralistic society will give high levels of fulfilment of the aspirations of all of us. There is one great aspiration common to humankind, perhaps to life in

general, which goes beyond personal fulfilment, and that is to explore and expand outwards. It is the trajectory of this goal which goes beyond the scope of *DA-Value* which is the subject matter of *StarDust*.

3.3 The Strategy

Chapter 4

Refining and Effecting Value Systems

First, the aims, then the philosophical underpinnings, the principal structural elements of the proposed solution and the long term strategy and near term plan for effecting this solution.

- [Ambition]
- [Philosophy]
- [Architecture]
- [Strategy]

Chapter 5

Sense, Model, Plan, Action

Chapter 6

Language Truth and Logic

The aim of this project is to build software to support an *Intelligent Deductive Cloud* in which Higher Order Logic (HOL) is a universal semantic and deductive foundation.

- The Ambition
- The Philosophy
- The Architecture
- Strategy and Plan

a note on hype cycles

6.1 The Philosophy

That the ambition embraces General Autonomous Deductive Intelligence demands a philosophical underpinning for an appropriate knowledge architecture capable of the desired level of diversity while main-

taining consistency and rigour even while employing the most advanced methods from Artificial Intelligence and Deep Learning.

Is this section a suitable minimalistic philosophical foundation is outlined.

- Some Philosophical Background
- The Problem and the Method
- Metaphysics and Ontology
- Language and Logic
- [Deduction and Induction]
- [Consistency and Trust]
- [Connecting Models to the World]

6.1.1 Some Philosophical Background

A natural basis for a minimalistic philosophy oriented towards formal science, engineering and technology is the positivistic tradition, particulalry in its most recent manifestation in the philosophy of Rudolf Carnap. The philosophy outlined here can therefore be considered a moderate 21st Century Neo-positivism. If you don't know what that means, don't worry, all will be explained. If you think you know what it means, bear in mind that positivism is traditionally *anti-philosophical*. It sweeps most philosophy into the waste-bin, and is therefore generally condemned and misrepresented by virtually all professional philosophers.

So here are a few notes mentioning doctrines commonly associated with Rudolf Carnap which were not in fact present or significant in his mature philosophy, and are not here adopted.

6.1.1.1 What Carnap's Philosophy Was Not

- phenomenalism, i.e. the idea that only phenomena are real, that there are only sense data

  Carnap never was a phenomenalist, not even while he was writing his *Aufbau* (an attempt to establish a phenomenalistic account of the world).

- nominalism, the desire to keep to an absolute minimum the kinds of things which are supposed to exist

  Nor was Carnap a nominalist, ontologically he was a free-wheeling pragmatist, quite happy to countenance the greatest extravagances of Cantorian set theory if these proved convenient for science.

- the verification principle, i.e. the idea that the meaning of a sentence is it method of empirical verification, and that a sentence which lacks this kind of content is meaningless.

  This certainly was one of Carnap's views, early in his career. However, he recognised soon enough that it was not tenable, and gradually watered it down to almost nothing. He did continue to have an interest in the ways in which empirical science could be connected with the evidence supporting (or refuting) it, and this came later as work on confirmation theory and on the testing of empirical theories.

When conventional refutations are given of "logical positivism", by which is principally meant the philosopy of Rudolf Carnap, the authors rarely pay much attention to what Carnap clearly documents as his principal aims. They do not judge him against what he was trying to achieve. What Carnap wanted to do was to facilitate the use in

science of the kinds of formal notation to which he was first introduced by Gottlob Frege as a student, and which he was inspired to apply to make philosophy "scientific" and science formally rigorous.

Admittedly, Carnap might not be considered as very successful in that. The widespread adoption of formal notations in science, as in mathematics, depends upon a level of technological support (in dealing with the extra complexity of detailed formal proofs) which has, 50 years after his death, still not been realised.

It is the purpose of this project to contribute to the realisation of the required technical infrastructure, and it depends on getting a minimal philosophical framework in place to render that enterprise intelligible.

6.1.2 The Problem and The Method

Preliminary Discussion

Wittgenstein's philosophy, and his conception of language, fell into two phases. In the first, manifest in his "Tractatus Logico-Philosophicus" his conception of language was derived from the new formal notations devised by Frege and Russell primarily for the purpose of achieving rigour in mathematics. He then came to recognise that this model of language did not accommodate the great variety in how language is used.

Though his first narrow conception of language did not encompass all, the kind of language it describes is of particular importance to science. We will call this "propositional" language, it is the kind of language with which this project is principally concerned. Its importance derives from its ability to provide objective knowledge of the nature and state of the world, which can be recorded and re-used in places and

at times very different to those in which is was first encountered. Using this kind of knowledge, it is possible to predict how the world and artefacts in it will behave in situations which we have not yet encountered. We are therefore empowered to take steps to mitigate the consequences of accidents of nature, and to design tools and architect buildings which will be effective and safe. Underlying this kind of science and engineering there is propositional knowledge, and underlying the predictive capabilities it supports there is deduction.

The deductive cloud is intended to hold and to reason using this kind of knowledge, and the primary requirement for philosophical underpinning is to give an account of propositional language, how it can be represented and exploited deductively.

This is an enterprise which can be related to just some aspects of Wittgenstein's *Tractatus*, and it therefore demands not great tomes of broad ranging philosophical doctrine, but rather, a compact and clear account of the special kinds of language involved, the ways in which they related to our world, and the ways in which they can be effectively exploited in the deductive cloud.

Simplicity and precision are key to the logical fundamentals, and the principal method adopted here for giving a minimalistic philosophical account suitable for our purposes may be understood through an analogy with the idea of feeding the Universe through the eye of a needle.

The problem to be solved by this analogy is what I call "the problem of regress in the semantics of language". This is the problem that in order to define the meaning of some language we must use language, but for the definition to make sense, we need to define the language in which it was given, and then the language in which *that* definition was given. We therefore require either an infinite regression of language definitions, or some "meta-circularity" (the definition directly

or indirectly given in the language itself).

Of course, it is known that meta-circular definitions may have more than one fixed point, and therefore do not by themselves suffice to fix a language but the combination of a meta-circular definition with a careful description in a small and relatively unambiguous portion of natural language will suffice for our purposes. This small and unambiguous portion of natural language is the eye of our needle. Through this eye we will pull a language at first very simple, but powerful, abstract HOL. This language will by incremental conservative extension grow to provide for the formalisation of mathematics, science, engineering, and any other kind of propositional knowledge. Though the underlying abstract language in all these domains is formal, and therefore precise enough to support long chains of rigorous deduction, concrete presentations can be in whatever form best suits its intended audience, even through the ubiquitous chat-bot.

6.1.2.1 Preliminary Discussion

The deductive cloud is a repository for knowledge about logic, mathematics, and about the physical world we inhabit, in suitable form for rigorous deductive reasoning by intelligent artefacts.

To devise an architecture for such a system, to explain that architecture and give grounds for the supposition that the architecture will deliver the desired capabilities, and will be trustworthy, can only be done in the context of a philosophical framework which embraces metaphysics, logic, philosophy of language, epistemology, science and more. The enterprise depends upon, and potentially disrupts, very many branches of philosophy. To devise an architecture which coherently embraces all those things which potentially fall under the scope of our deductive cloud and the synthetic intelligence we seek in it, re-

quires a broad philosophical synthesis. But for a philosophy to provide a stable base for this enterprise it must be simple and solid. I therefore believe that the kind of minimalistic philosophy belonging to the positivistic tradition is likely to serve best, and will seek to articulate such a system.

I see the philosophy as a prerequisite for good architectural thinking, and the architecture itself as a kind of *constructive philosophy*. Taking cognitive capabilities a matter for design, demands a perspective on them very different from that traditional in philosophy.

A fundamental issue, and one which has been a matter of controversy throughout the history of artificial intelligence, is the manner in which knowledge is represented. This is particularly significant in that the approach to AI which is at present making the greatest impact is sometimes associated with the idea that no "representation" is needed. I shall not here consider that extreme position, but recognise that the manner of representation can vary very widely, between the symbolic representations which seem to be of the essence in deductive reasoning, and the distributed weightings which we might regard as a kind of representation of knowledge in a neural net.

That the main purpose of this project is the exploitation of formal deductive inference makes the symbolic representation of knowledge a core feature of the project and this will be the focus of much of our attention both in these philosophical preliminaries and in the architectural sketch which follows it. Despite this, the exploitation of deep neural nets will in my opinion be essential to achieving anything close to the intended deductive intelligence, and these nets will store knowledge in ways more similar to synaptic weights than logical formulae.

6.1.3 Metaphysics and Ontology

In this minimalistic philosophy one might expect the philosophical high ground of metaphysics to be poorly represented, particularly since we are following in the footstep's of Carnap who dismissed metaphysics as meaningless. In fact, Carnap was pragmatic about metaphysics, but did not use the term "metaphysics" for those abstract theories which play an essential role in the formulation of scientific hypotheses or models (such as the definition of spatio-temporal concepts as needed for the articulation of the theory of relativity, or the exotic ontology of cantorian set theory as a practical way of underpinning the mathematical theories of real and complex analysis required for large parts of science and engineering).

This kind of metaphysics need not concern us here, since it will forms a natural part of the use of HOL in the formalisation of science.

What concerns us here are the ontological pre-requisites for defining and giving meaning to the HOL abstract language. This I propose to do using the concept of a "purely abstract entity".

First a discursive explanation. A concrete entity is something physical. The term "abstract" may be used for anything which is not concrete. Some arguably abstract entities may have concrete constituents. Thus the set of members of parliament may be considered abstract, even though the members themselves are concrete. To call an entity *purely abstract* is to say that neither is concrete nor has any concrete constituents (at any level).

Now this discursive definition, in all likelihood, would not be regarded as satisfactory by all philosophers. The problem "what is an abstract entity" is a philosophical problem which will probably continue to be debated as long as there is more than one philosopher. Consensus is

unlikely ever to be reached. One reason for this is that the asking of the question is symptomatic of the presumption that the term *abstract* has some definite meaning, which I suggest is quite unlikely. It is highly probably that over the course of history not only have philosophers disagreed about what abstract entities are, they have used the term "abstract entity" in quite different ways, have given it in their usage, quite distinct meanings to that given to it by their philosophical antagonists.

The resolution of these problems is not required for our enterprise. For our purposes, it suffices to give to the term "purely abstract entity" that meaning which renders it convenient for our purposes.

In doing this it is convenient to take a leaf out of the book of the later Wittgenstein, by construing language as a game and taking the liberty of arranging the rules to suit our project. In the following talk of (purely-)abstract entities the rules of the game, of the language associated with this kind of entity, are as follows. A kind of abstract entity is defined by giving properties common to all entities of this kind, and giving existence criteria determining which of the possible entities with those properties exist. A simple but important example is that of pure extensional sets. The features common to all such entities are, firstly that they may have members which are themselves pure extensional sets, they may be members of other pure extensional sets, and they are extensionally unique; no other pure extensional set has exactly the same members. We then stipulate which of these sets exist, for simplicity let us say that, there exists an empty set and that for any finite collection of pure extensional sets there is a pure extensional set whose extension is that collection.

The question whether such sets "really exist" has no meaning in this fragment of language. The meaning of the phrase "there exists a pure extensional set x such that P x" is determined exclusively by the rules

of the game just given.

Bertrand Russell talked of such entities as "logical fictions". This is a position which we could here also adopt without any pragmatic consequences. I don't care for it myself, because to call something a fiction is to deny its existence, and since we are in the business of devising new languages for technical purposes, we get to chose what "exists" means for the abstract entities which are in the domain of discourse of these languages, and in relation to that chosen meaning, the existential theorems provable can reasonably be said to be true rather than fictional.

6.2 The Architecture

The principal elements of the architecture proposed are as follows:

- The Logical Kernel
  There may be more than one logical kernel available to work in the context of the knowledge hierarchy we envisage, often adaptations of existing systems, but we have in mind one brand new kernel with a number of special features which are intended to support the ambitions we have articulated.
- Distributed Theory Hierarchy as Knowledge Base
  Rather than keeping a theory hierarchy private to each instance of a theorem prover, the DA-Hol architecture provides for a single distributed theory hierarchy and a diversity of proof tools, which undertake or contribute to extensions to that hierarchy.
- [Distributed Theory Development] Not only is the hierarchy of contexts and theories distributed, but the process of deriving new theorems may be distributed across the network with different nodes in then network competing to provide the most effective

capabilites in different special domains which may contribute to a single result.
- [Assurance Levels] In an elaboration of the scheme whereby "oracles" are admitted subject to tagging of results with information about which oracles they depend upon, proof tools have unique identifiers (URIs) which are combined to give "assurance levels" which tag theorems in a theory, and associated RSA key pairs used to tag and sign theorems
- [Deductive Intelligence] providing Deduction-as-a-Service
- Agents (and people) train and evolve in an [Proof-Market Incubator]

6.2.1 The Logical Kernel

The logical kernel is designed to achieve the following effects:

- abstraction from concrete representations (this results from structuring of the context/theory representation and use of abstract rather than concrete syntax except on end-user interfaces)

- Computational efficiency to support *caclulemus* like functionality, i.e. efficient execution of calculable functions (calculemus, named after a dictum of Leibniz, is an initiative aimed at combining the capabilities of computational computer mathematics software such as Mathematica with that of Interactive Theorem Provers).

- full logical metalanguage allows intelligent assistants to understand high level proofs

Here are some features we envisage in such a logical kernel.

- the kernel will follow an augmented LCF paradigm, i.e.:
  - the logical kernel provides inference functions which implement a fixed set of derived rules in the HOL logic.
  - high level presentations of proof computations may be provided using a functional metalanguage
- those features are augmented in the following ways:
  - a derived rule is available which effectively evaluates efficiently an executable part of the HOL object language
  - the metalanguage is also HOL and is used in high level descriptions of proofs
  - theorem proving takes place in two logical contexts at once, an object context and a meta-context. The meta context contains the definitions of all the high-level proof capabilities used in a proof, and the high level proof can therefore be preserved to admit re-validation of proofs after their initial construction.

6.2.2 Distributed Theory Hierarchy

In DA-Hol we use *abstract HOL* as a "universal" representation for knowledge.

Here we sketch some of the key aspects of how that works.

The deductive cloud comes in two principal ways:

- Native
  - Logical Contexts
  - Abstract Theories
  - Concrete Theories
- Non-native
  - Static

– Dynamic

6.2.2.1 Native

6.2.2.1.1 Logical Contexts

All knowledge representation is to be done in languages obtained by extension of the core abstract HOL language. These abstract languages are defined by structures which are called "logical contexts", whose role is to define a HOL signature, and to constrain (define) the names in that signature. In the distributed deductive cloud, each logical context has a Universal Resource Identifier and a new context would be represented by a file which refers to the context or contexts which it extends or combines using the URI for that context including a hash of the definition of the context, and digitally signed by the agent which created the context. This is a pattern which is repeated for all the resources in the deductive cloud. Resources are new files with an associated URI (often a URL) which extend by reference using URI and hash of the resources being extended supplemented by the details of the extension, the whole signed by the creating authority.

Because the internal language is abstract, the contexts are spartan. The abstract language is extended by adding new vocabulary, and the new vocabulary is given meaning by some *constraint* which is a sentence in HOL which will be true only if the values of the new names are as required. Often this will be a *definition*, but sometimes the constraint will not wholly determine the values of the new names. These constraints are usually *conservative*.

6.2.2.1.2 Abstract Theories

When new results are deduced, these are saved in the deductive cloud as *theories*. A theory consists of three main parts. It has references to two logical contexts, and it provides a number of judgements which detail the results of the deductions, viz: certain new theorems together with information about how these theorems were obtained. The two logical contexts required are a context for the object language in which the proven theorem is expressed, and one for the meta-theory containing the necessary vocabulary for a high-level description of the proof of the theorem, or of other meta-theoretic results which are held in the theory.

As in LCF systems generally the language in which proofs are expressed is extensible, and as theories are developed the vocabulary used to describe proofs is often extended as well. In this case this language is an abstract HOL language, but as with all other uses of abstract HOL, the concrete presentation may be chosen to suit the users of the system.

6.2.2.1.3 Concrete Theories

A concrete theory is a theory together with information describing how that theory may be presented to agents which require something more concrete than the abstract structures, typically people, and which facilitates the entry of extensions (of contexts or theories). Once again, HOL is used to define the concrete presentations, and a context is needed which provides the necessary vocabulary for defining the concrete presentation, this is a kind of meta-context and may itself come as a concrete theory.

6.2.2.2 Non-native

DA-Hol is not intended only to support deduction in small logical enclaves of the cloud. It is intended to provide a framework for reasoning deductively about anything which can be found in "the cloud". Of course, it is easier to provide sound reasoning within some context if the system has constructed the context with that in mind. However, a variety of methods can be deployed which will permit reasoning in contexts which include information which has not been curated by DA-Hol.

The problem is particularly difficult for data sources which are dynamic, for if treated naively these will be prolific sources of contradictions, so we sketch here some possible approaches to each of these broad categories in turn.

6.2.2.2.1 Static

There are two distinct classes of method which can be used to reason about static data in the cloud. These both depend on a formal semantics for the data in question. In the first case this is given in the meta-language, and results in the alien data source being defined as an alternative concrete syntax for HOL. In the second case the data is treated as a HOL data value, possibly a finite sequence of characters, and a function is defined in HOL which maps values of this kind into whatever values or propositions best represent then in the appropriate HOL contexts.

Thus, for example, a table with two columns, one a social security number, the second an age, might be imported into a HOL context in which there is a predicate A(n,y) which means "the person whose social security number is n has age y years", using a function which maps a

table of two such columns to the conjunction of the set of predicates A(n,m) for all values n and m which occur together in some row of the table.
A context can be created which imports a non-native data source into a DA-Hol context by treating the data source as concrete syntax for a new language whose abstract syntax is rendered in abstract HOL.

This involves defining in a meta-context (using HOL) the concrete syntax of the language, how this maps to its abstract syntax, and how the abstract syntax of the language is represented as HOL. Thus would typically revolve around defining in HOL new constants which correspond to the constructors in the abstract syntax and capture the semantics of that kind of structure in the language. This the classic pattern for a "shallow" semantic embedding.

Once that is done, the data can be imported to create a matching HOL context.

6.2.2.2.2 Dynamic

The problems arising from non-monotonicity of changes in the premise set for deductive inference has been widely discussed in the AI community, and is often thought to require the use of "non-monotonic logic". DA-Hol itself is founded in a monotonic logic, but the universality of this logical system means that any coherent logical system with a definite semantics will be interpretable in abstract HOL and may therefore be used as a concrete presentation for an idiolect of HOL. The use of such non-monotonic languages may provide an effective way to deal with data from dynamic external contexts, but we also have a number of ideas for methods which may be used without adopting a non-monotonic language.

6.2.3 Logical Contexts

In talking about languages the distinction between meta-language and object-language is important. A meta-language is one in which one talks about some other language, the language about which one speaks is the object language. It may vary from one conversation to the next which language is the meta-language and which the object language, I might use language A to talk about language B at first, and then use language B to talk about language A. The use of abstract HOL as a universal representation means that in any such discussion, both the meta- and the object language will be different (or the same) concrete presentation of sentences for which the underlying representation is abstract HOL, which is therefore both the object- and the meta-language at once. Because of its pluralism in concrete languages and the high levels of automation which it is intended to support, meta-language or meta-notations, and meta-theoretic reasoning will be pervasive.

In order to maximise the re-usability of knowledge, as much of the work as possible will be done in abstract HOL, confining concrete presentations to the dialogue with human beings, and permitting therefore that the same knowledge can be presented in different concrete forms for different purposes or for different people or groups of people.

These considerations lead to a restructuring of the kinds of information which has traditionally been found in the theory hierarchy of a HOL interactive theorem prover (ITP). The most fundamental core of the architecture here proposed is therefore the notion of a logical context, the effect of which is to determine a signature or vocabulary and a collection of constraints on the values which may be taken by the names in the signature.

Each signature has associated with it two sets of names. The first set of names is the names of type constructors. The second set is that of term

constants. The signature contains, for each type constructor name a set of non-negative whole number arities at which the constructor has been constrained, and for each term constant, a disjoint set of "types" at which the constant has been constrained.

A context consists of a signature and a sequence of constraints. Each constraint consists of a set of type constructor/arity pairs and a set of term constant/(polymorphic)type pairs and a sentence (boolean term of abstract HOL) which mentions those type constructors with the given arities, and those constants at the given types. ### HOL for Knowledge Representation

Here we expand upon that idea. Some prior discussion appears in The Philosophy, here we therefore steer clear of the deeper issues and focus on slightly more practical matters.

The architecture is intended to provide a context for superhuman deductive capabilities. In normal human discourse, in all but the rarified strata of research in pure mathematics, chains of deductive reasoning are short, and are checked by common sense evaluation of their results. Nevertheless in those domains outside of mathematics, such as philosophy, where complex reasoning may be employed, the results are inconclusive, partly because of ambiguities in the context which render different lines of reasoning equally plausible despite their incompatible conclusions.

If we move to a context in which long chains of deduction become routine, then this kind of contextual ambiguity or incoherence will soon be exploited by the derivation of explicit contradictions, rendering all further deduction meaningless. For this reason it is crucial to DA-Hol that the context in which reasoning takes place is unambigous and coherent. It is proposed that this is achieved by the use of abstract HOL for all knowledge representation, in a manner similar to that in which it is normally used in existing HOL theorem provers, i.e. exclusively

by conservative extension over the primitive HOL language.

It is a consequence of this that all the truths represented will have a pure interpretation under which they express logical truths (this is sometimes call the broad conception of logical truth, otherwise known as analyticity). This broad notion of logical truth encompasses the truths of mathematics, but not those of empirical science.

Representation of empirical knowledge in science and engineering, or any other subject matter, is made possible in two ways. Firstly, though the possibility of multiple interpretations of the vocabulary in any context. Though all the truths can be interpreted in a pure well-founded set theory, they can also be interpreted in many other ways. When thus interpreted the mechanisms policing conservative extensions will guarantee the existence of an abstract interpretation, and hence the consistency of the resulting theory, but some intended interpretation in which the some of the entities involved are physical entities will permit the language to talk of the real world.

Though the possibility of concrete interpretations does provide one avenue for the use of abstract HOL for the presentation of empirical facts, there are reasons for preferring instead to think in terms of mathematical models which provide useful approximations to the structure of the real world. The main reason is that most scientific theories and engineering theory, uses models which are deliberate simplifications, and therefore, taken literally, are false. The disadvantages of this become lethal if we have two theories each useful, but which are incompatible, e.g. newtonian and relativistic mechanics. So long as we treat these as two logically independent mathematical models then no problems arise in having both of these theories available in some context. But if we were to regard these as both making true claims about the nature of the world we inhabit, then our context would become incoherent, and would no longer provide a basis for sound deductive reasoning.

6.2.4 Language and Logic

We are concerned here exclusively with *propositional* language and *deductive* logic.

Propositional Language
Deductive Logic
Logical Consistency and Completeness
Interpretation and Proof Theoretic Strength
Translation and Expressiveness
The Universality of HOL
Practical Considerations

6.2.4.1 Propositional Language

Propositional language is language in which one can express or assert sentences. A sentence is an expression which has a truth value. The elements of the language have meanings which combine to give meaning to the sentences of the language, and the meaning of a sentence consists of, or includes, its truth conditions. If you understand the meaning, and hence the truth conditions, of a sentence, then when you are told its truth value, you know something about the subject matter of the sentence. You know that the conditions necessary for the truth of the sentence hold.

So that's a thumbnail sketch of what propositional language is.

6.2.4.2 Deductive Logic

A logic is a set of rules which allow inference from a group of sentences in some language to a new sentence. If the logic is sound, then these

rules correspond to the meanings of the sentences of the language in the following way: the rules preserve truth. This means that whenever the rules permit drawing some conclusion C from a set of premises S, then if all the sentences in S are true, then C will also be true. Put another way, in a sound logic the rules only admit inference from S to C if whenever the truth conditions for all the sentences in S are satisfied, then so will be the truth conditions for C, in more jargon, the conjunction of the truth conditions for the sentences in S entails the truth condition for C.

Some kinds of logic are not sound in this way, though there may be controversy about whether these systems really should be accepted as _logic_s. An example is *inductive* logic.

6.2.4.3 Logical Consistency and Completeness

Sometimes the proof rules for some system allow any sentence to be derived, and the logic is then considered to be *inconsistent*, otherwise the logic is consistent. One way to establish consistency is by ensuring that the language has a clearly defined semantics (i.e. the truth conditions of the sentences are well defined) and then proving that there is at least one sentence which is false under that semantics (i.e. its truth conditions are never satisfied) and that the logic is sound. This establishes that no falsehood is derivable and the logic is consistent.

Most logical systems in use are consistent, though there have been many cases in which logical systems once thought consistent were found to be inconsistent.

A logical system is *complete* if all true sentences of the language are derivable according to the rules of the logic. Kurt Godel proved the incompleteness of all (effective) logical systems which contain arith-

metic, and this applies to most of the languages of interest here. We therefore have to settle for logical systems in which only some of the logical truths can be proven, and it is a matter of interest in a logical system how many of the truths expressible in the language can be proven in the logic. Similar considerations apply also to the semantic expressiveness of a language. It might be (and it always will be) that a language is complete only because it is limited in what truths it can express.

These considerations bear upon the claim I have made about "universality of HOL", upon what that claim means, and whether it can be established.

6.2.4.4 Interpretation and Proof Theoretic Strength

6.2.4.5 Translation and Expressiveness

6.2.4.6 Practical Considerations

6.3 Strategy and Plan

Strategy is as yet embryonic.

There is an organisational strategy and a technical strategy.

6.3.1 The Organisational Strategy

The organisation needs and opportunities depend on how many other creative developers decide to contribute to the project. So long as its just me, organisation is almost a non-problem, but even then since

the architecture is intended to support (and even depend upon) autonomous deductive agents, there are "organisational" problems about how that should be done (even if only one human, still, any number of machines).

The scale of the ambition is not strictly compatible with that outcome, however likely it might be. It would then be a contribution to architectural thinking about this important topic, together with some prototyping of key features of the proposal.

Like most open-source development, no formal organisation would be strictly necessary even with a substantial number of collaborators. However, since this is designed to be a contribution to a digital economy, the economic ideas need to be explored and prototyped just as much as the purely deductive aspects, so even with a small number of collaborators I would be hoping to explore formal constitutions with a view to building a digital constitution suitable for automation, under whose auspices "Deduction as a Service" would be prototyped.

The crypto-digital-economy is a moving target, so ideas about how this project fits into it are fluid, but the organisational strategy is to evolve the organisation to intercept what seems, as the architecture is filled out and prototyped, to be the most effective way of delivering the benefits promised by the architecture.

6.3.2 The Technical Strategy

The technical strategy is essentially *architecture* then *intelligence* then *application*. But of course, it is essential that the earlier stages are exposed to feedback from prototyping and implementing the later stages.

I did set out hoping to complete a manifesto first, and then move on to design and prototyping, but of course that's not going to happen. The

manifesto and the follow-up in more detailed architecture and design, will be strung out over a longer time period than I would like, while work also begins on prototyping key elements of the architecture.

The aim is to move to an intelligent deductive capability in a good architectural context at the earliest possible moment, oriented to relatively simple applications which can drive the early economics and drive work into the learning process. While I do want to see a sound and well-documented philosophical and logical basis, I don't want this to delay getting a practical grip on the technologies involved, so I am hoping that some kinds of prototyping activities will be undertaken from the very beginning.

- First of all, build the story, the big picture, in the wiki
- Second, work out some detail (possibly have to step outside the wiki for the more detailed parts) turning the sketch into an architecture, and then into successive levels of design for key elements.

6.3.3 The Plan