Steps Toward a World Wide Digital Library of Formal Mathematics

Robert L. Constable
Computer Science Department
Cornell University

Mathematical Knowledge Management Symposium
Heriot-Watt University, Edinburgh, Scotland
November 25-29, 2003

Introduction

This talk is based on results from two related research projects:

1. Cornell/Cal Tech/Wyoming MURI research project to build a formal digital library (FDL)
2. Cornell project to explore roles for the FDL in the National Science Digital Library (NSDL) being created by the NSF

Goals

MURI Project goal:

- Explore ways of using a digital library of formalized algorithmic knowledge to improve software reliability

NSF goal:

- Explore ways of using a digital library of formalized mathematics to improve mathematics and computer science education, and improve communication of scientific results

General Goal

Investigate the scientific and social potential of an unusual information resource – approximately 50,000 formal theorems and proofs created by interactive theorem provers

> *This resource represents a frontier of a century and a half effort to perfect the notion of a mathematical proof*

I think we captured the historical moment well in our MURI proposal, from which I will quote later

Social and Scientific Potential

- Social aspects:
  - Preservation of formal mathematics
  - Publication opportunities
  - Establishing a field of scholarship
  - Restoration of material, e.g., Automath
- Information Science aspects:
  - Knowledge creation from formal data
  - Automatic clustering and search
  - Datamining and machine learning
  - Natural language translation

Guidelines for our work

- Offer theoretical neutrality
- Try to include results from all major provers
- Avoid a monolithic software system
- Stress knowledge management services

Introduction (cont.)

Claim: Learning how to share formal mathematics among theorem proving systems is an interesting and important problem

Interesting because:

- Technically challenging – Howe, Moran, Allen
- Vision is exciting – a "digital Bourbaki"
- Symbolic database for CS

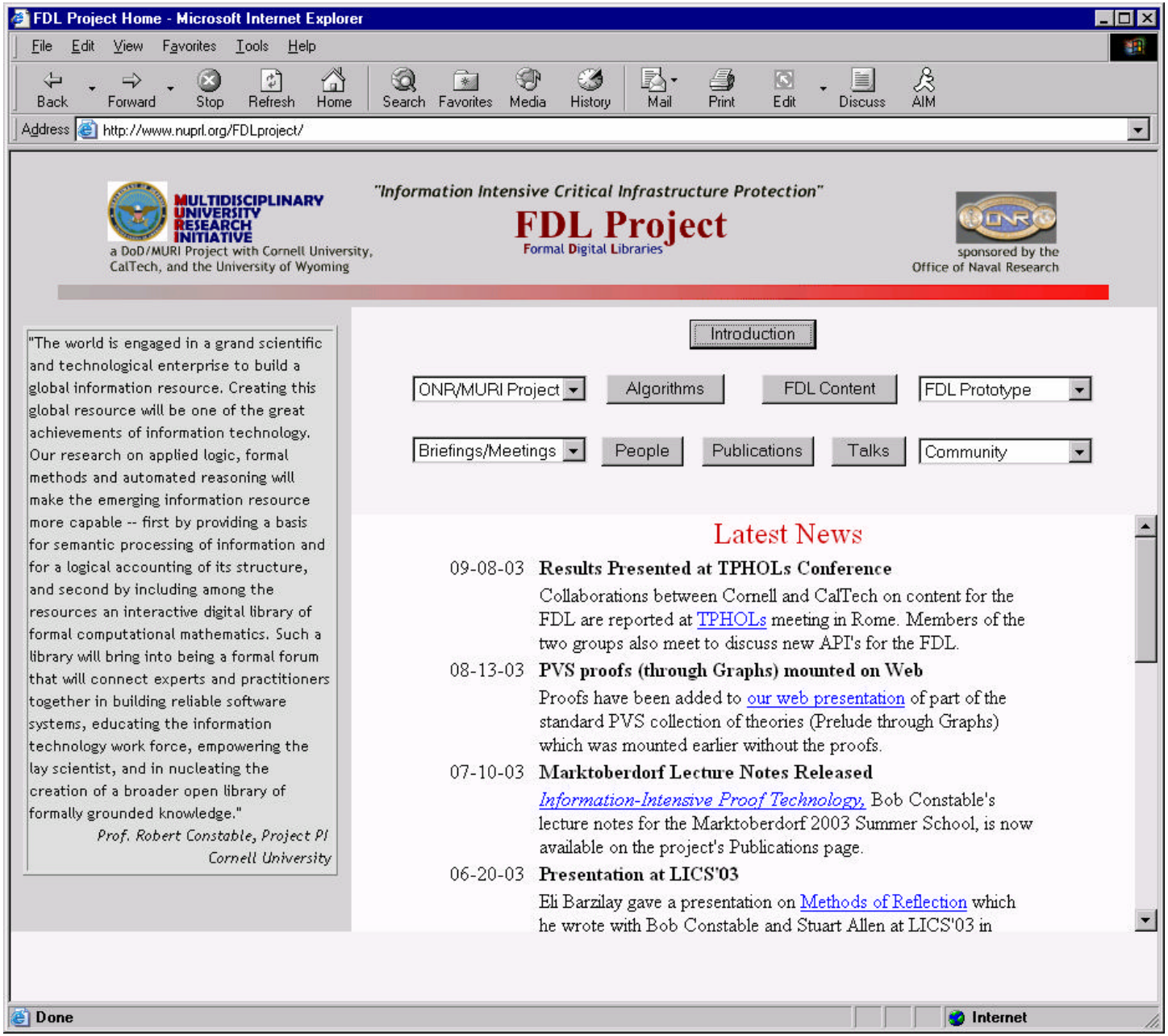
Introduction (cont.)

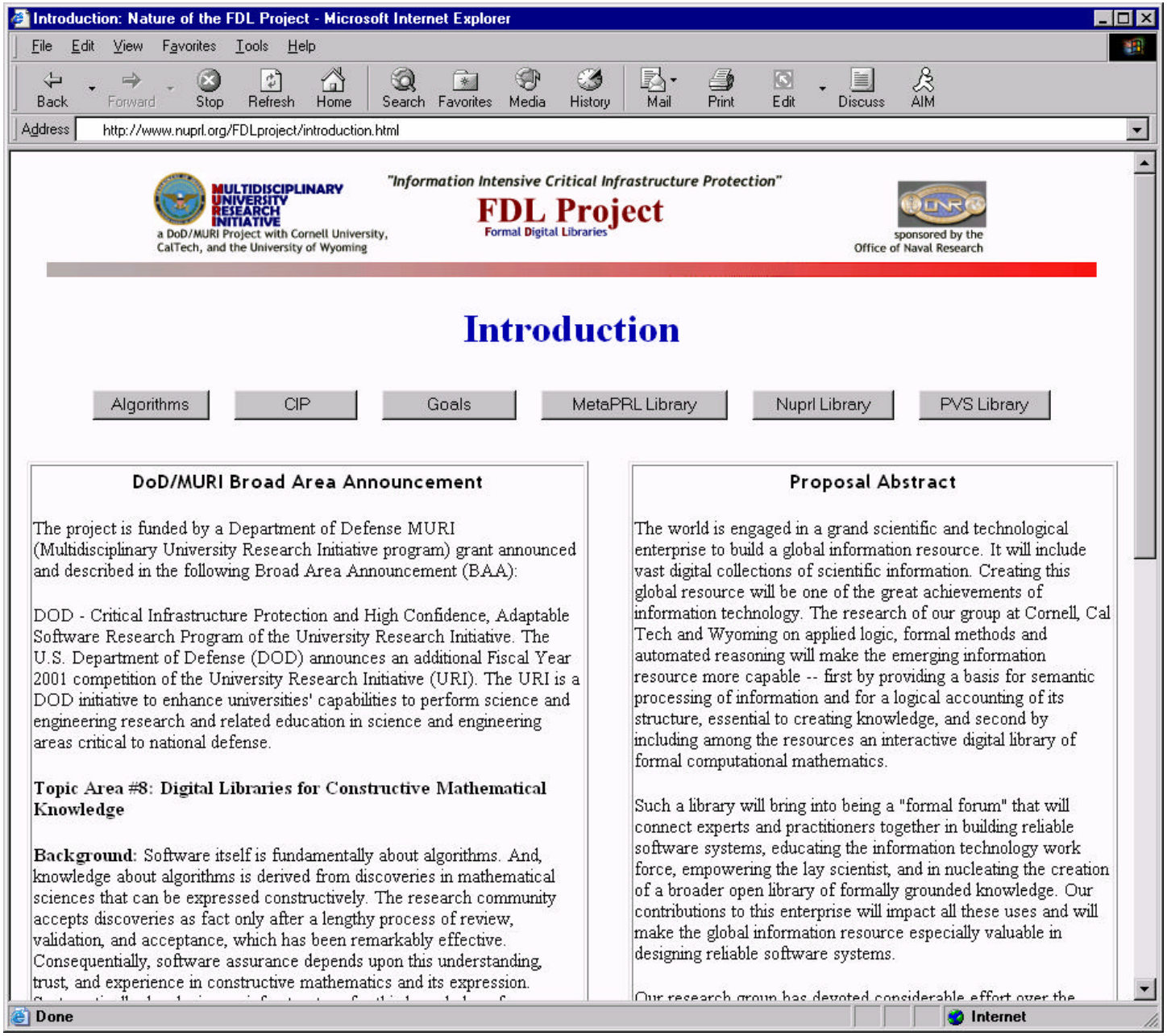
Important because:

- Sharing will enhance verification technology needed to improve software reliability and security
- Enable a richer field of formalized mathematics
- Social impact – scientific communication

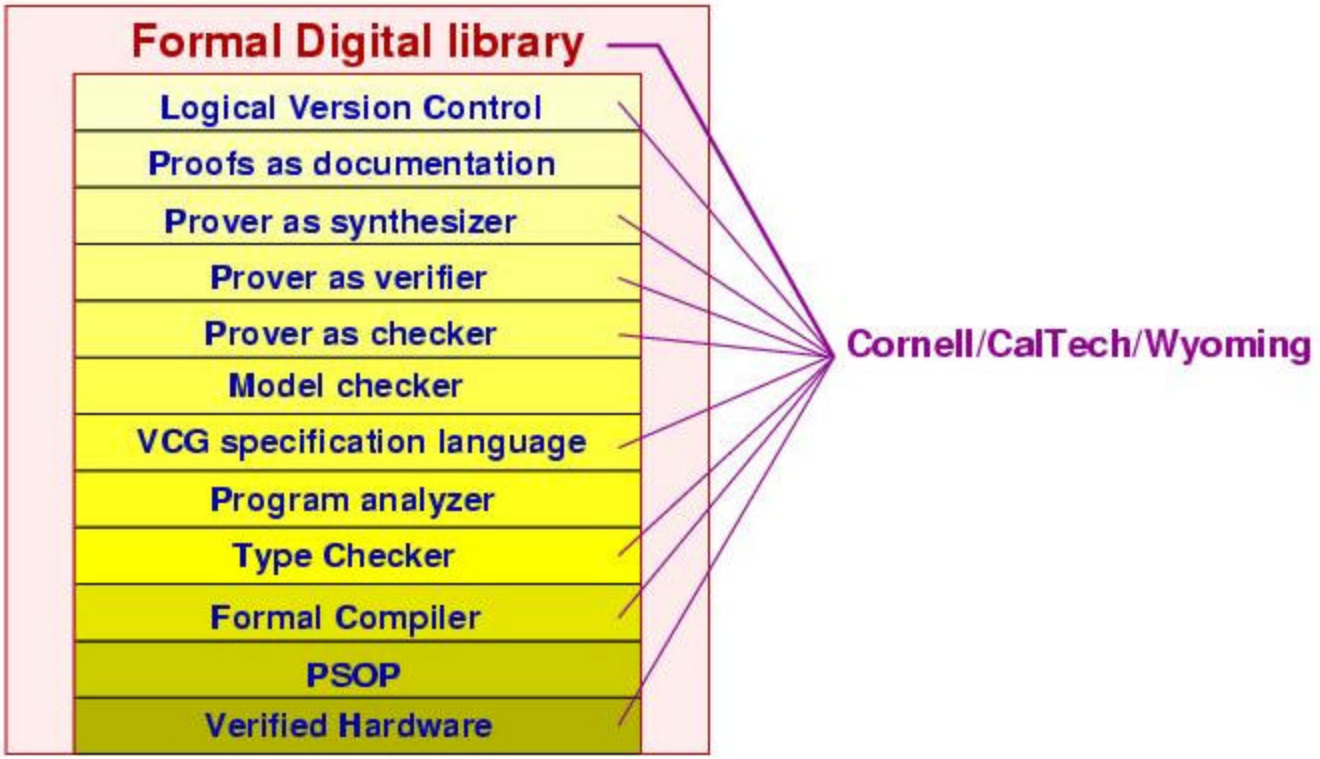
Plan of the Talk

- FDL Role in System Verification
- Features of an FDL
- Theoretical Basis for Sharing
- Conclusion

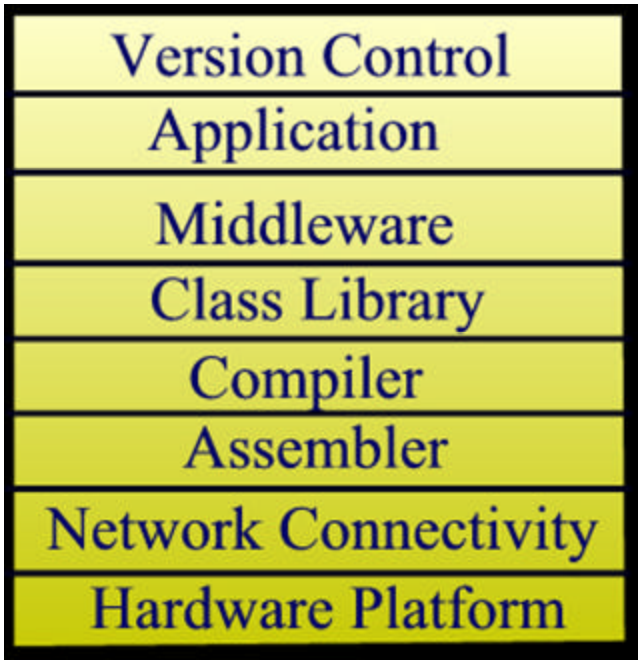

FDL Project Home - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites Media History Mail Print Edit Discuss AIM
Address http://www.nuprl.org/FDLproject/
MULTIDISCIPLINARY UNIVERSITY RESEARCH INITIATIVE
a DoD/MURI Project with Cornell University, CalTech, and the University of Wyoming
"Information Intensive Critical Infrastructure Protection"
FDL Project
Formal Digital Libraries
sponsored by the Office of Naval Research
"The world is engaged in a grand scientific and technological enterprise to build a global information resource. Creating this global resource will be one of the great achievements of information technology. Our research on applied logic, formal methods and automated reasoning will make the emerging information resource more capable -- first by providing a basis for semantic processing of information and for a logical accounting of its structure, and second by including among the resources an interactive digital library of formal computational mathematics. Such a library will bring into being a formal forum that will connect experts and practitioners together in building reliable software systems, educating the information technology work force, empowering the lay scientist, and in nucleating the creation of a broader open library of formally grounded knowledge."
Prof. Robert Constable, Project PI
Cornell University
Introduction
ONR/MURI Project
Algorithms
FDL Content
FDL Prototype
Briefings/Meetings
People
Publications
Talks
Community
Latest News
09-08-03 Results Presented at TPHOLs Conference
Collaborations between Cornell and CalTech on content for the FDL are reported at TPHOLs meeting in Rome. Members of the two groups also meet to discuss new API's for the FDL.
08-13-03 PVS proofs (through Graphs) mounted on Web
Proofs have been added to our web presentation of part of the standard PVS collection of theories (Prelude through Graphs) which was mounted earlier without the proofs.
07-10-03 Marktoberdorf Lecture Notes Released
Information-Intensive Proof Technology, Bob Constable's lecture notes for the Marktoberdorf 2003 Summer School, is now available on the project's Publications page.
06-20-03 Presentation at LICS'03
Eli Barzilay gave a presentation on Methods of Reflection which he wrote with Bob Constable and Stuart Allen at LICS'03 in
Done
Internet


Introduction: Nature of the FDL Project - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites Media History Mail Print Edit Discuss AIM
Address http://www.nuprl.org/FDLproject/introduction.html
MULTIDISCIPLINARY UNIVERSITY RESEARCH INITIATIVE
a DoD/MURI Project with Cornell University, CalTech, and the University of Wyoming
"Information Intensive Critical Infrastructure Protection"
FDL Project
Formal Digital Libraries
sponsored by the Office of Naval Research
Introduction
Algorithms CIP Goals MetaPRL Library Nuprl Library PVS Library
DoD/MURI Broad Area Announcement
The project is funded by a Department of Defense MURI (Multidisciplinary University Research Initiative program) grant announced and described in the following Broad Area Announcement (BAA):
DOD - Critical Infrastructure Protection and High Confidence, Adaptable Software Research Program of the University Research Initiative. The U.S. Department of Defense (DOD) announces an additional Fiscal Year 2001 competition of the University Research Initiative (URI). The URI is a DOD initiative to enhance universities' capabilities to perform science and engineering research and related education in science and engineering areas critical to national defense.
Topic Area #8: Digital Libraries for Constructive Mathematical Knowledge
Background: Software itself is fundamentally about algorithms. And, knowledge about algorithms is derived from discoveries in mathematical sciences that can be expressed constructively. The research community accepts discoveries as fact only after a lengthy process of review, validation, and acceptance, which has been remarkably effective. Consequentially, software assurance depends upon this understanding, trust, and experience in constructive mathematics and its expression.
Proposal Abstract
The world is engaged in a grand scientific and technological enterprise to build a global information resource. It will include vast digital collections of scientific information. Creating this global resource will be one of the great achievements of information technology. The research of our group at Cornell, Cal Tech and Wyoming on applied logic, formal methods and automated reasoning will make the emerging information resource more capable -- first by providing a basis for semantic processing of information and for a logical accounting of its structure, essential to creating knowledge, and second by including among the resources an interactive digital library of formal computational mathematics.
Such a library will bring into being a "formal forum" that will connect experts and practitioners together in building reliable software systems, educating the information technology work force, empowering the lay scientist, and in nucleating the creation of a broader open library of formally grounded knowledge. Our contributions to this enterprise will impact all these uses and will make the global information resource especially valuable in designing reliable software systems.
Done
Internet

Verification Technology Stack



Software Development Stack



Key Observations

- Advanced tools are computation intensive
  - Model Checkers
  - Interactive Theorem Provers
- But there is a missing component; we need a large machine accessible information resource
- We provide an information-intensive approach to CIP/SW, in the verification end of the stack

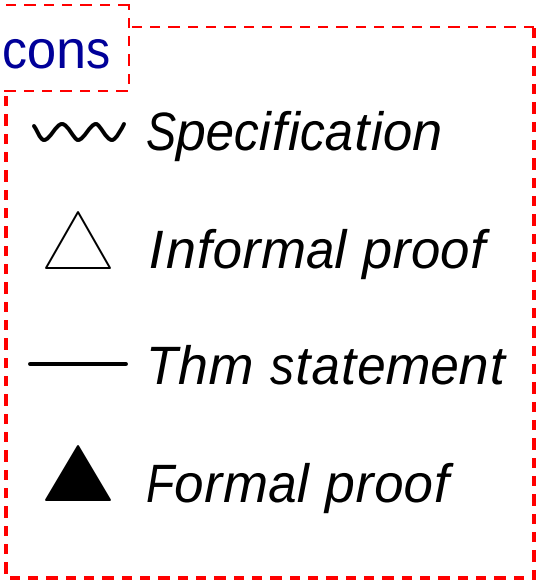
Code Synthesis

- We synthesize code from constructive proofs that a problem is solvable

$$\forall n : \mathbf{N}.\ \exists p : \mathbf{N}.\ \ n < p \ \&\ \mathit{Prime}(p)$$

Extract primes: $\mathbf{N} \rightarrow \mathbf{N}$

Correct-by-construction program

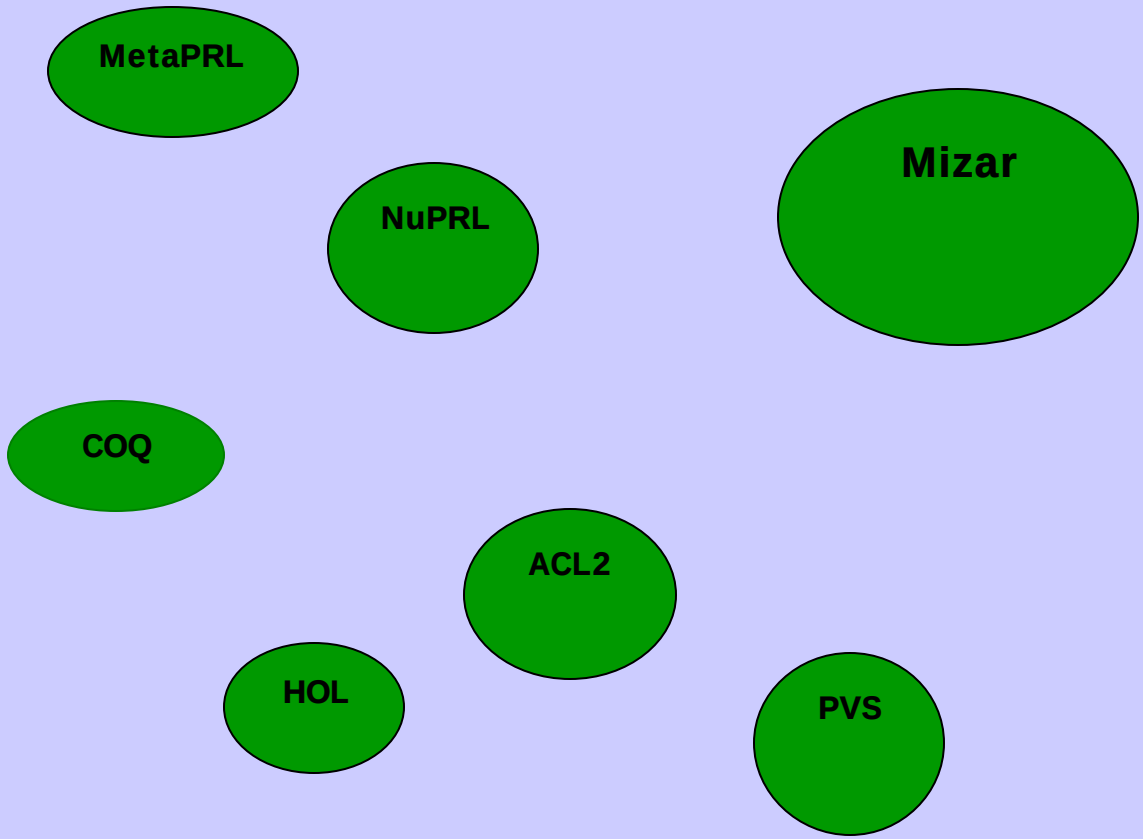


– These proofs are very large objects

– All details are given

The Opportunity and the Challenge

- Significant amounts of formal material available – 50,000 theorems – but in islands; vast duplication
- Can it be shared?
  - Very hard problem
  - Very large payoff



Theorem proving systems as isolated islands, reaching a small community

Why is Sharing Formal Knowledge Hard?

- There are few standard parts, and we don't know how to make them
- Mechanisms for computer processing of the information make accounting for truth difficult
- It is probably necessary, and certainly desirable, that there is more than one verification logic (in the US)
- Given that there will always be several programming languages – C, Java, Lisp, NL – it is easy to understand why there will always be several provers

Why is Sharing Formal Knowledge Hard?

1. PVS

Every natural number is prime or not — intuitive

FOR ALL n:Nat. (Prime(n) OR *NotPrime(n))* — formal

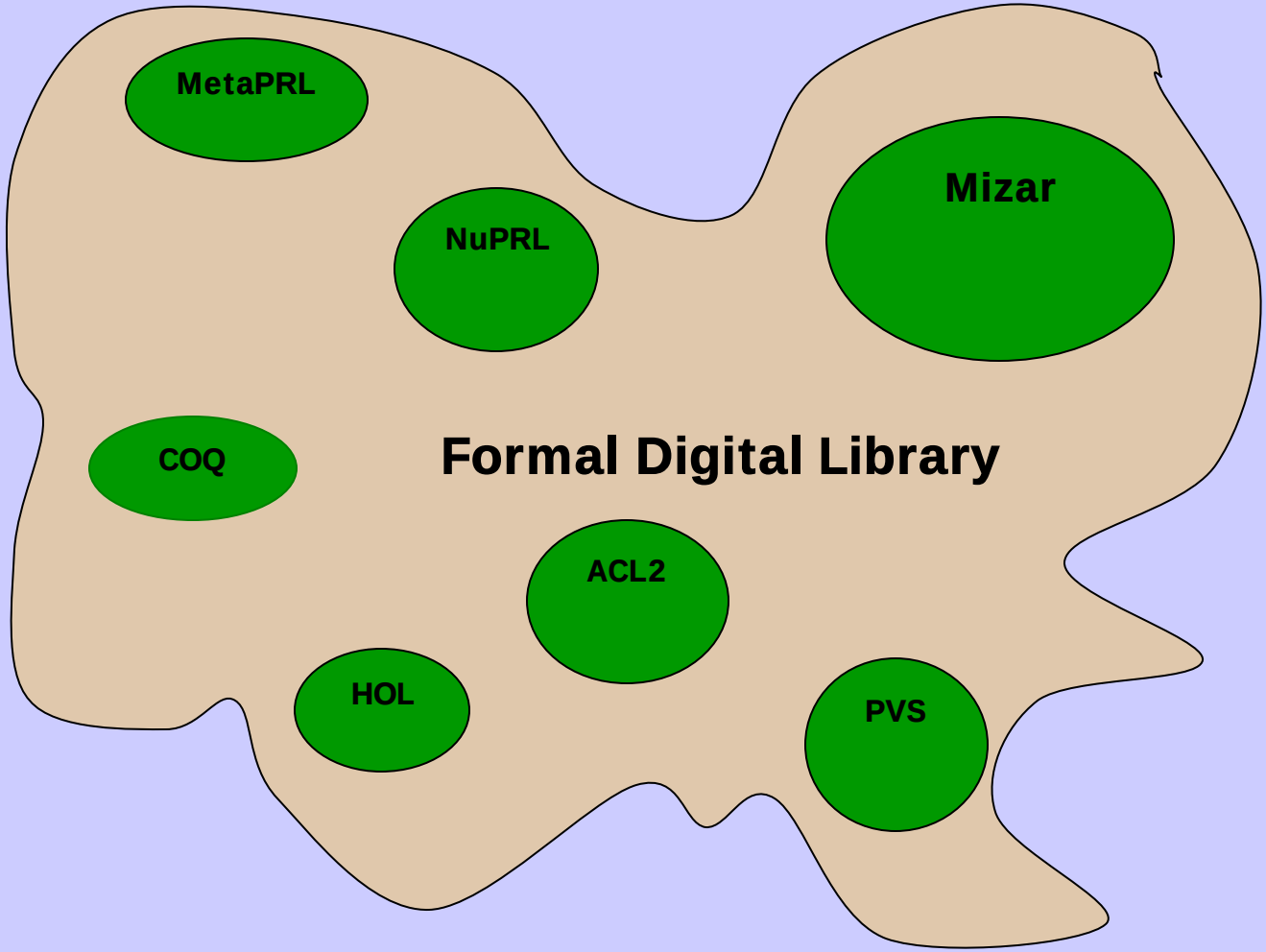
2. Coq

There is a method to decide for each natural number whether it is prime or not.

$$"n : \mathbf{N}.\,(Prime\,(n) \vee \neg Prime\,(n))$$

In the case of 2, a formal proof (▲) contains an implicit algorithm. Coq can extract it as a function: $Prime\,? : \mathbb{N} \to \mathbb{B}$.

Alf, MetaPRL, Nuprl, and Isabelle-CHOL can all do the same (Nuprl pioneered the idea and methods)

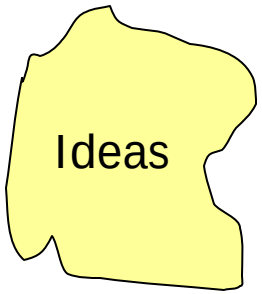


The FDL joins all the small islands into a single big island that can accommodate much more than the sum of all individual islands, and allows people to stand on the ground in between.

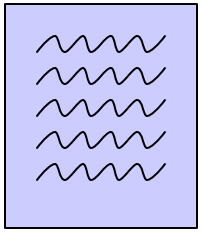
FDL and CIP/SW – First Steps

- A large information resource will speed up verifications
  - Can work at design speed
  - Could combine provers to harden a system in months instead of years
- A large information resource will help in:
  - Education
  - Software engineering

Advanced Tools Scenario



Design (**D**)



Intuitive Claim

Formal Spec

Argument

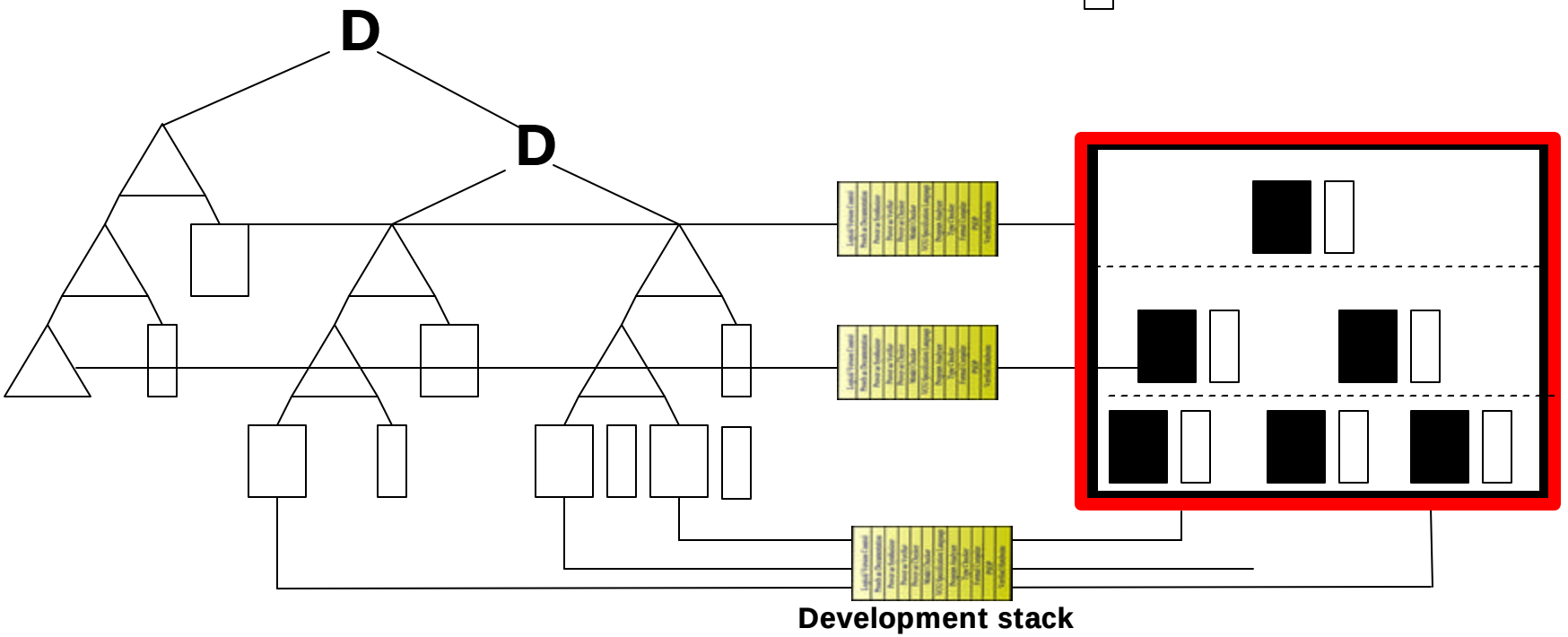
Proof

Abstract Code

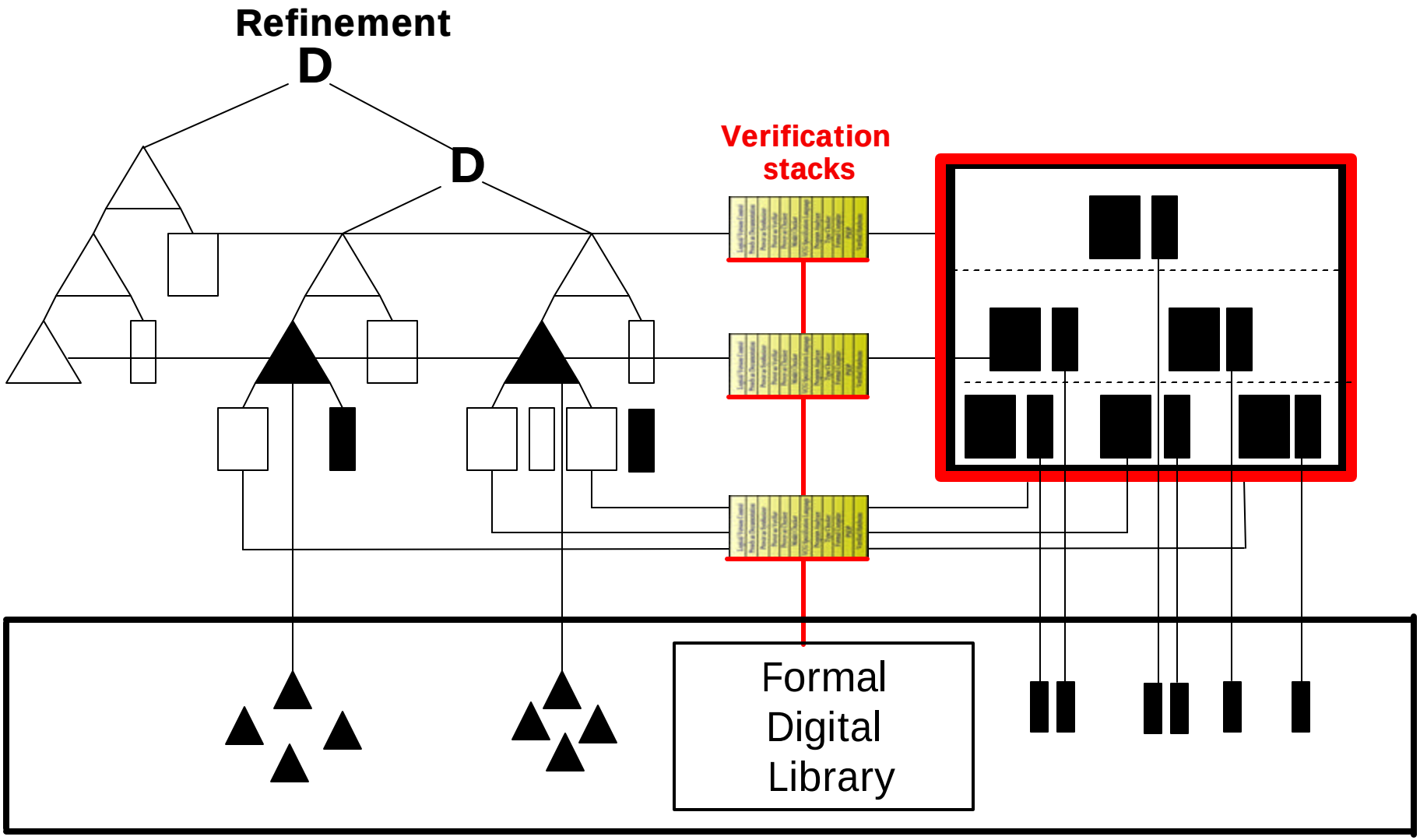
Code

Specs

Refinement

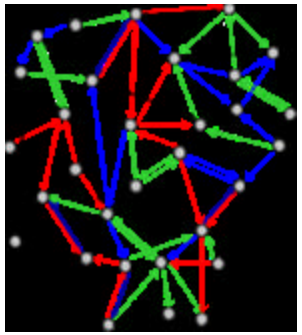


Advanced Verification Scenario



Vision

Grid Computing Service

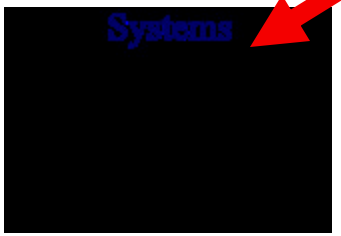


Knowledge Service (FDL)

- Logical Version Control
- Proofs as Documentation
- Prover as Synthesizer
- Prover as Verifier
- Prover as Checker
- Model Checker
- VCG Specification Language
- Program Analyzer
- Type Checker
- Formal Compiler
- PSOP
- Verified Hardware

Development And Verification Tools Stack

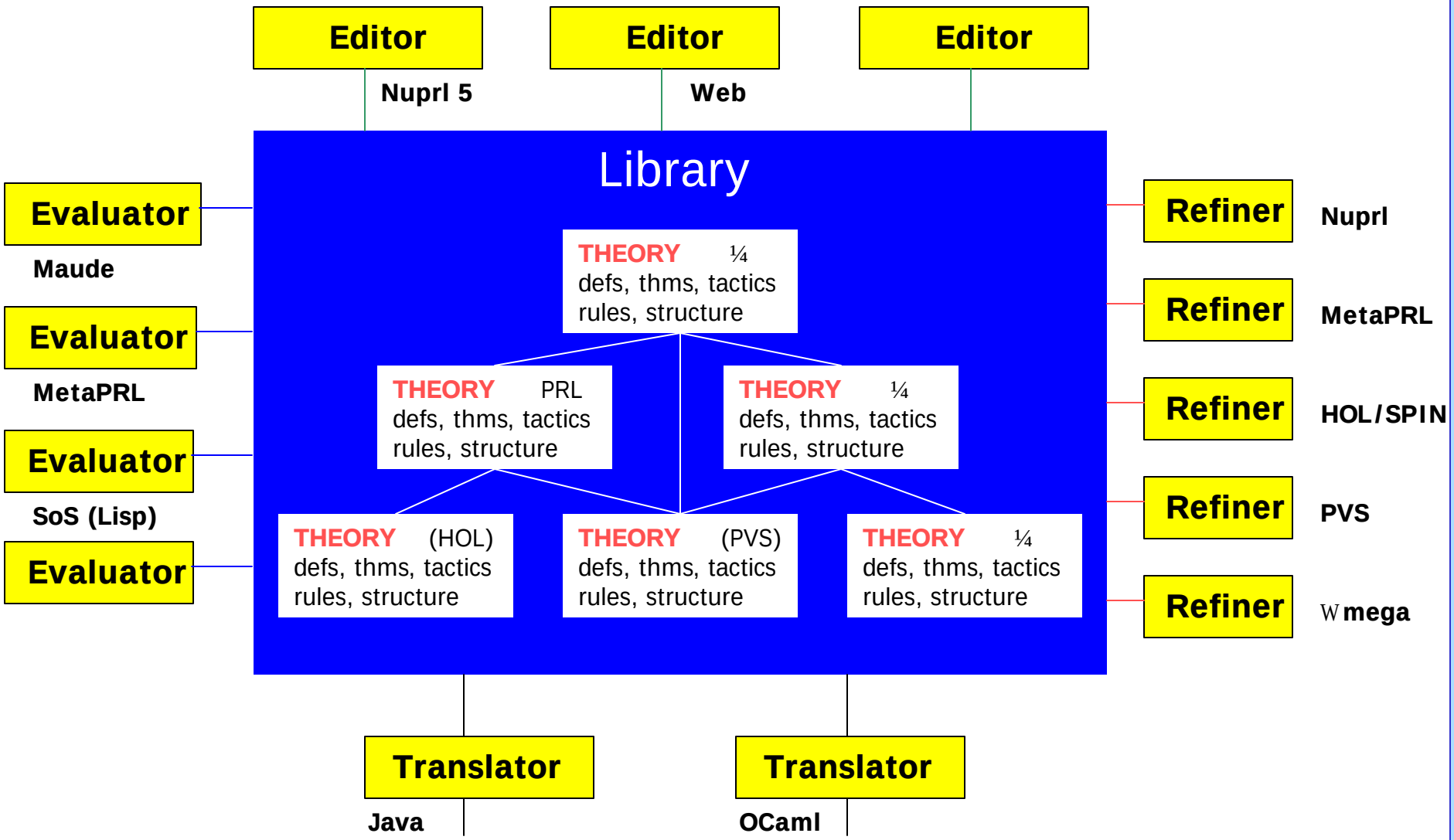
100,000 formal results (verified algorithms, formal classes, reference systems)



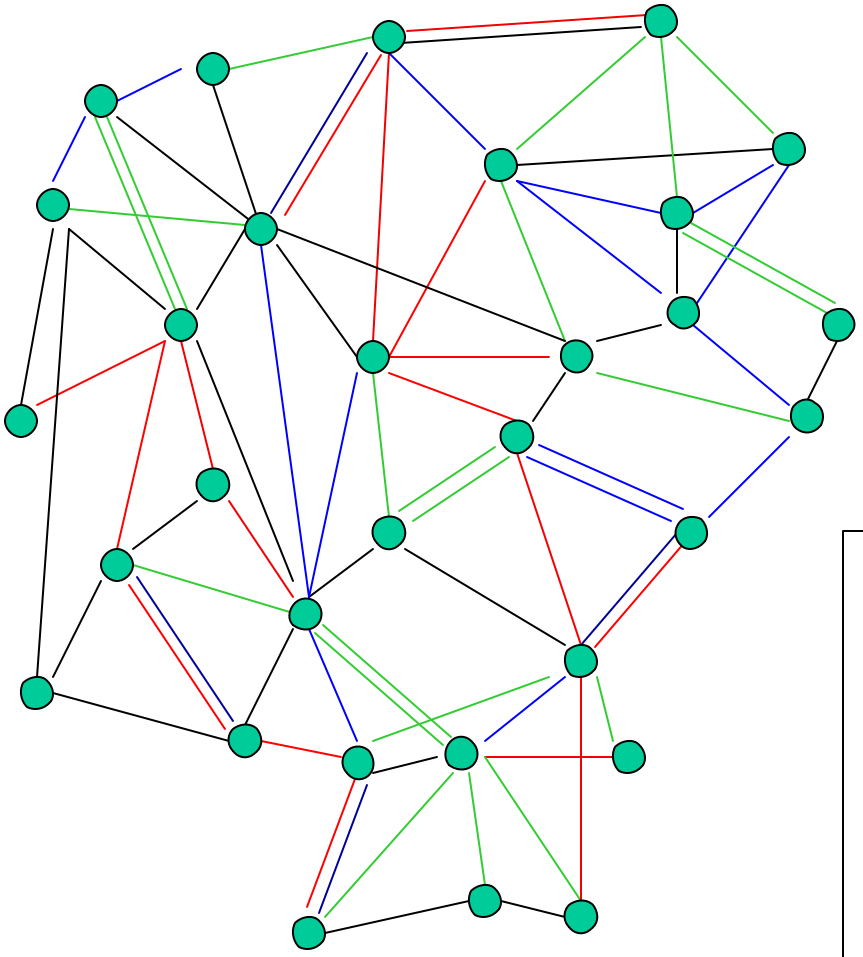
Plan of the Talk

- FDL Role in System Verification
- Features of an FDL
- Theoretical Basis for Sharing
- Conclusion

Formal Digital Library



Information Graph of the FDL

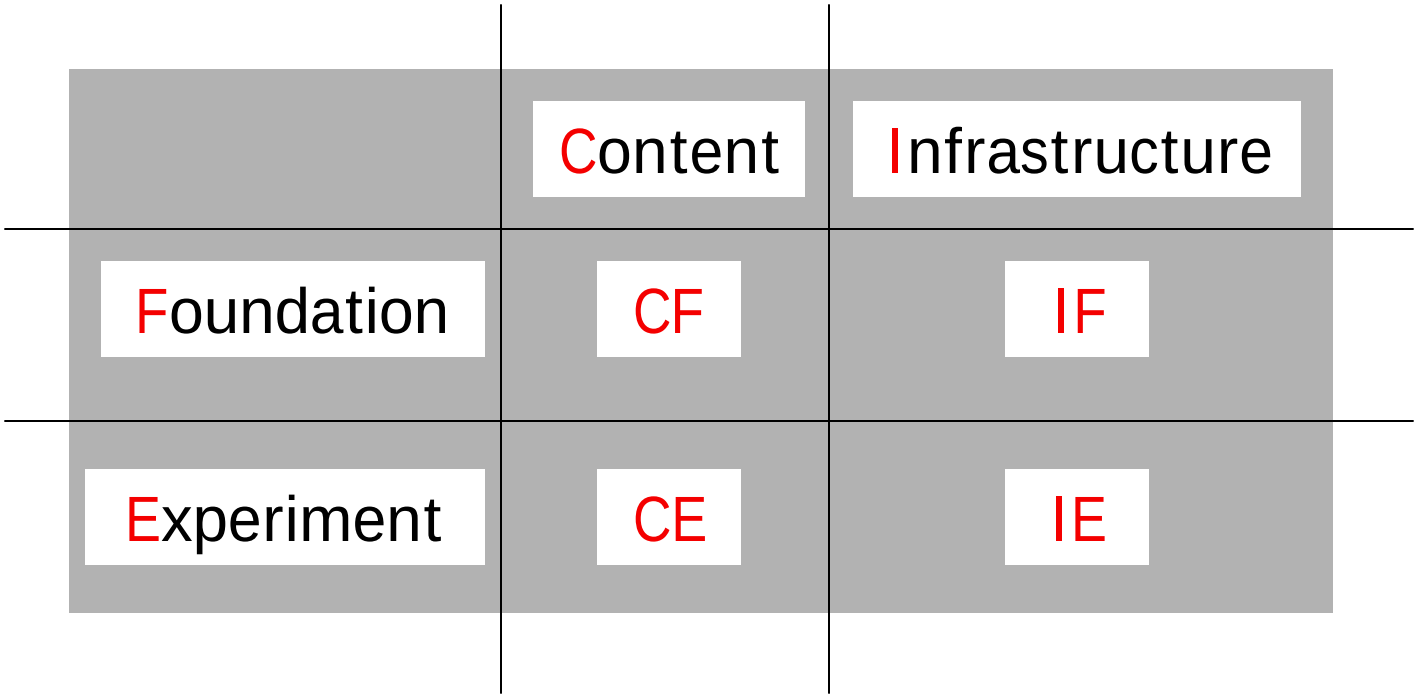


objects
logical dependency
textual links
accounting links
metalogical links

FDL Collects Formal Knowledge

- Knowledge ≠ Information
  - we must account for truth in ways that machines can check
- FDL is a logical library
  - evidence is provided
  - proof organizes evidence
  - very high standards of correctness

Our Technical Approach and Results



Research Questions

1. Content Foundations (CF)

- How can a library soundly and automatically facilitate sharing of content among different theories, different logics, and different implementations?
- What is the appropriate logical base for contemporary computing practice?

Research Questions

2. Content Experiment (CE)

- How can we formalize in a natural way, the concepts used in contemporary computing, e.g., classes, objects, inheritance, concurrency, distributed systems?
- What is the most useful mix of formal and intuitive knowledge?
- Can we integrate content from the three major representatives of US and EC provers?
  - Classical (HOL, Mizar, PVS)
  - Constructive (Alf, Coq, Nuprl)
  - Logical frameworks (Isabelle, MetaPRL, Twelf)

Research Questions

3. Infrastructure Foundations (IF)

- What is the best architecture for a formal digital library?
  - Tree of knowledge (hierarchical file system)

    vs
  - Database
- What accounting mechanisms are needed? How to include well-accepted intuitive knowledge?
- What is the best path to the Web? What comes first?

Research Questions

4. Infrastructure Experiments (IE)

- How to support experiments on the questions in 1–3?
- How to support formal/intuitive documents?
- What are the most enabling library services?
  - Dependency checking?
  - Basic accounting?
  - Web posting?
  - Clustering and classification?
  - Automatic Collection?
  - Search?
  - Translations?
  - Reflections?
  - Archiving?

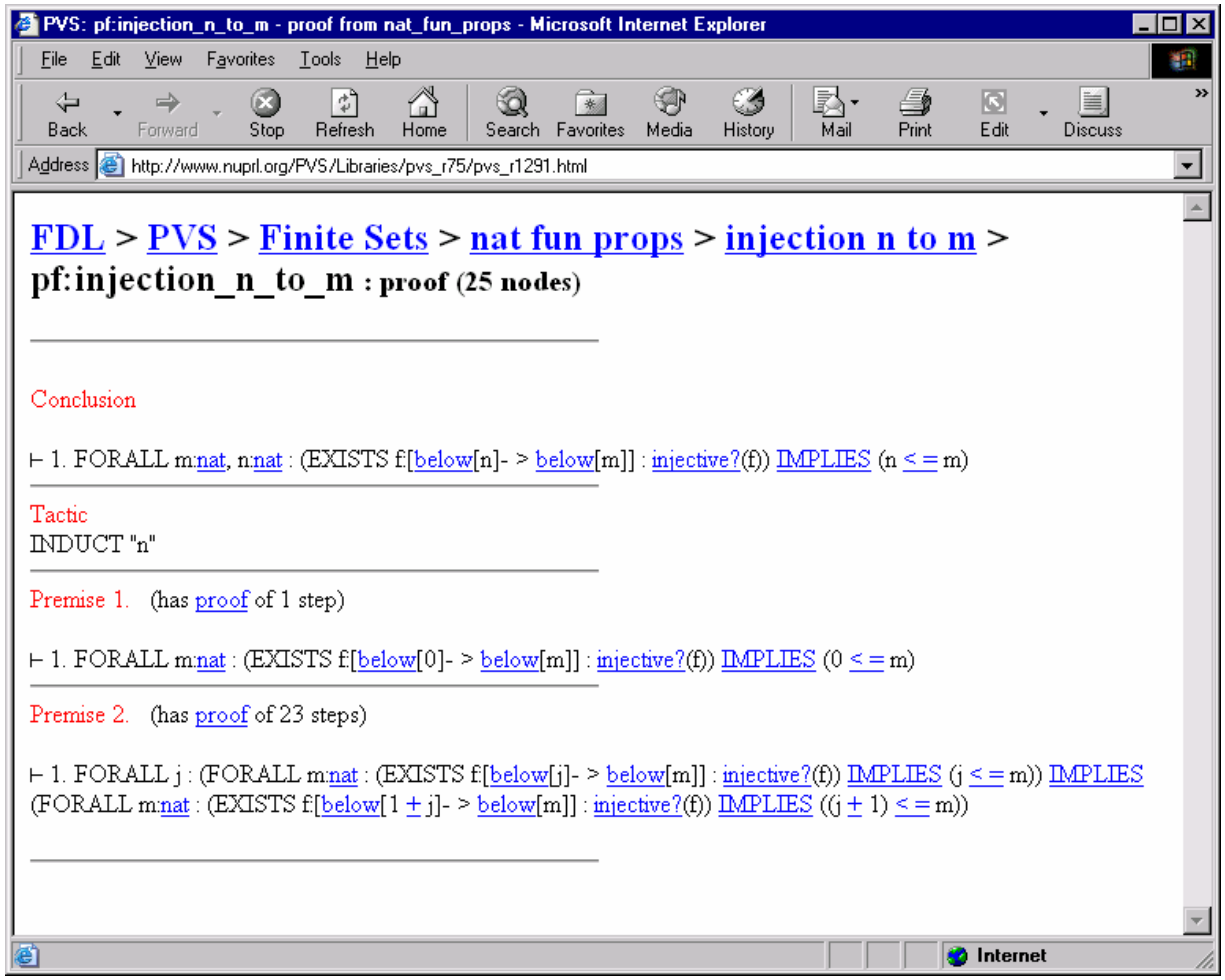
Research Results

4. **Infrastructure Experiments (IE)** (cont.)

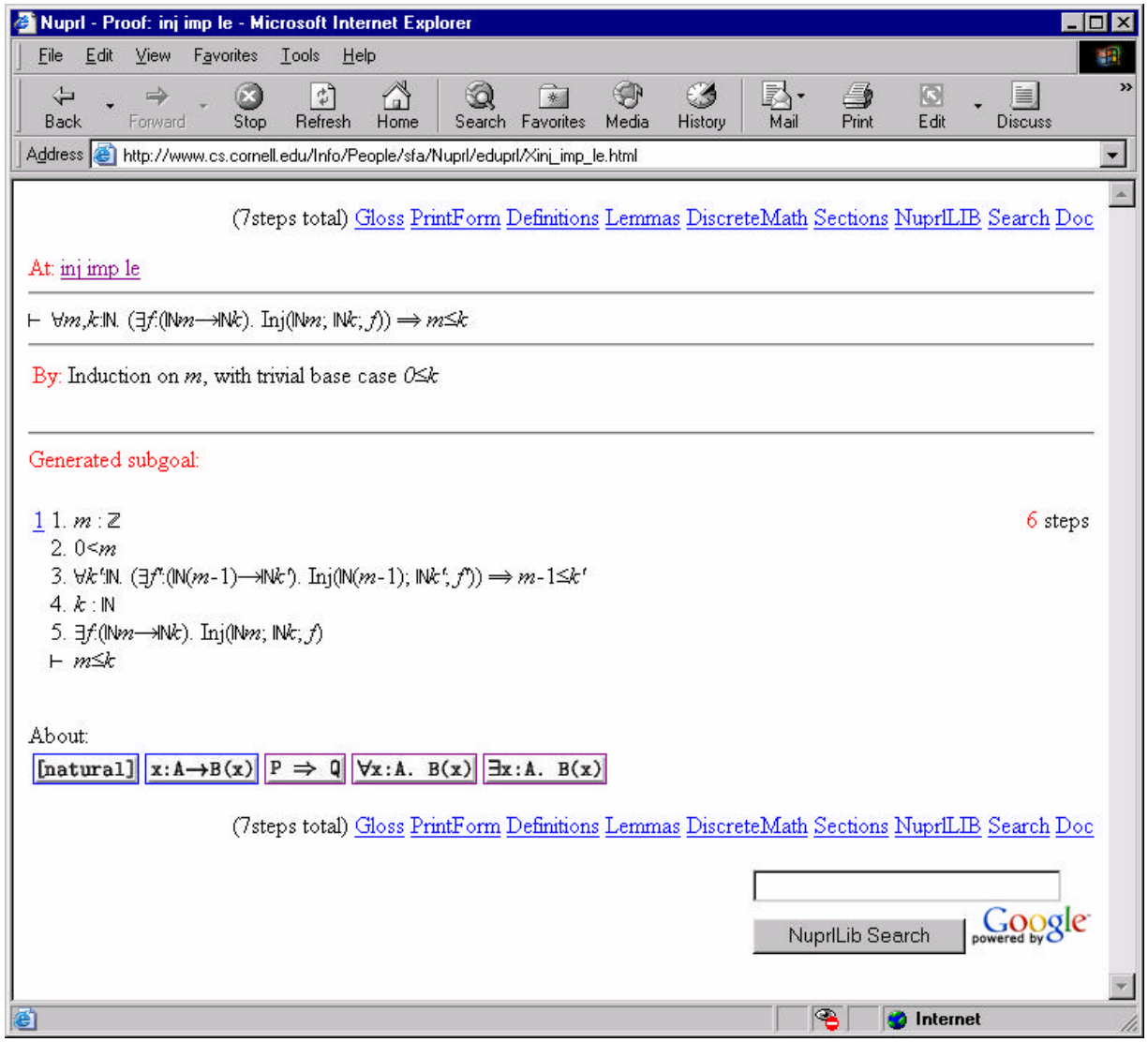
- We have implemented an experimental FDL with content from three provers (top Google source for PVS content). Allen, Eaton, Lorigo, Kreitz, Kopylov, Caldwell.
- We have connected the FDL to other services – Helm, OMDoc. Eaton, Lorigo. This is a significant achievement.
- We have implemented the Web publication service for all theories in the FDL. Allen, Eaton. This is a significant achievement.
- We have experimented with hybrid proofs and multiple-prover verifications. Koplov, Bickford, Kreitz.
- We have created a prototype editor for formal/informal (hyfi) documents. Allen, Eaton, Lorigo.

Accomplishments – Web Publication

Here are comparable results from PVS and Nuprl. We have created proof terms for PVS – a big deal



Accomplishments – Web Publication (cont.)



Accomplishments – Hybrid Documents

Local histories

An event system is a rich enough structure that we can define various "history" operators that list or count previous events having certain properties. Because we can define operators like these we do not need to add "history variables" to the states in order to write specifications and and prove them.

The basic history operator lists all the prior events at a location.

Definition

```
before(e)   ≡  if first(e)
               then []
               else before(pred(e)) @ [pred(e)]
               fi
  [e, e']   ≡  filter(λev.es-ble{i:l}(es;e;ev);before(e') @ [e'])
  rcvs(l;before(e'))   ≡  filter(λe.haslnk(l;e);before(e'))
   snds(l;before(e))   ≡  concat(map(λe.sends(l;e);before(e)))
snds(l, before(e,n))   ≡  snds(l;before(e)) @ firstn(n;sends(l;e))
```

Using these operators we can state (and prove) the following important lemma.

Lemma Fifo

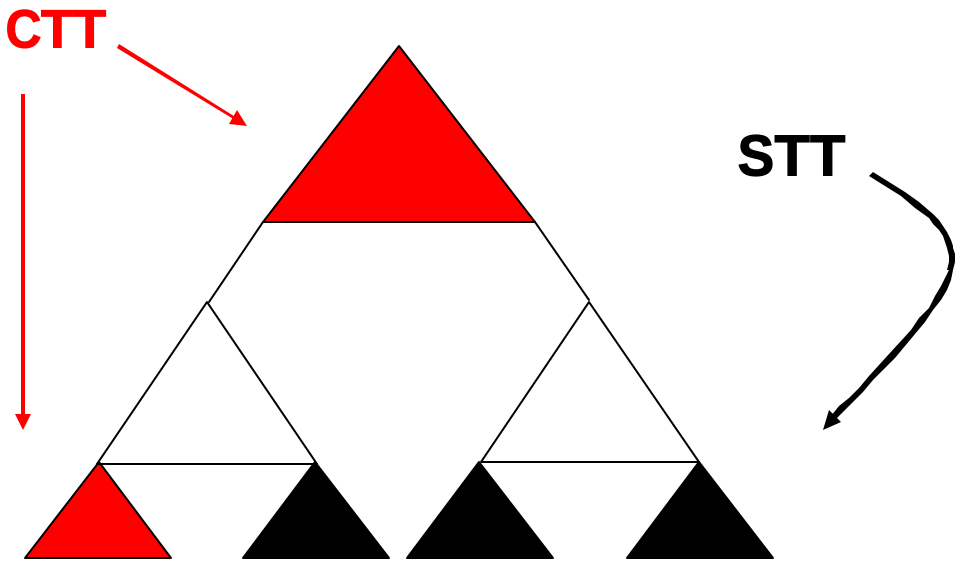
```
∀e:E
((↑isrcv(e))
⇒ (snds(lnk(e), before(sender(e),index(e)))
= msgs(lnk(e);before(e))))
```

proof: The proof is by induction on $<_{loc}$. The full proof is in then FDL.

□

Contents– Working Notes on FDL design (chapter 4)

Hybrid Proof



Goals

- Attract content providers: by the end of five years, people are submitting content from HOL, MetaPRL, and PVS to the FDL with almost no intervention from us (like arXiv)
- Participate in European Community (EC) effort (OMDoc, Helm, and possible Federated Math Library), and build a broad user community (from TpHOLs, MKM, IJCAR, NSDL)
- Use the FDL to support a system in current use, such as the Cal Tech formal compiler
- Attract authors who use FDL content for articles
- Attract reference system contributors and authors
- Attract dataminers
- Experiment with proof formats, e.g. Isar format

Specific Tasks

1. Extend the range of library services and API's, including more fully automatic submissions, automatic harvesting of formal metadata, automatic web posting, clustering methods, and search methods
2. Integrate certificates into the FDL, especially certificates for hybrid proof, and add sentinels
3. Accumulate further elements of a reference distributed system with security layers; use the system to collect and organize related formal verifications and models, especially from PVS
4. Support the MetaPRL formal compiler with general knowledge and related knowledge from other provers such as Isabelle-HOL
5. Accumulate more diverse algorithmic mathematics from such systems as ACL2, Isabelle-HOL, Coq, and Minlog
6. Deepen the links with OMDoc, Mbase, and other European Community library components, exploring cooperation with Mizar and the Journal of Formalized Mathematics
7. Move more FDL capabilities to the Web, including enhanced editors such as the Dynamic Pure Structure (DPS) editor, the Techexplorer services, and the ability to evaluate expressions and tactics

Plan of the Talk

- FDL Role in System Verification
- Features of an FDL
- Theoretical Basis for Sharing
- Conclusion

Howe's Meaning Functions

Howe's Approach to Set-theoretic Semantics:

Like Pitts, Dybjer, Troelstra and other, Doug Howe maps types into sets; for *A,* a type,

$$[\![A]\!] \text{ is a set.}$$

However, he introduces new ideas to handle functionality, polymorphism, subtyping, quotient types, and types as objects (universes). For instance,

$[\![\lambda x.x]\!]_{A \to A}$ is an element j of $[\![A \to A]\!]$

and he writes that $j \lhd \lambda x.x$.

Goal of Howe's Approach

Pure Type Theory

$$\frac{a_1, a_2, \ldots, a_i \ldots}{f\ (a)} \qquad f \Downarrow \lambda x.b \quad p \Downarrow < a, b >$$

$$\lambda x.b \quad < a, b > \quad c_i\left(\overline{a}\right) \quad \mathbb{U}_i$$

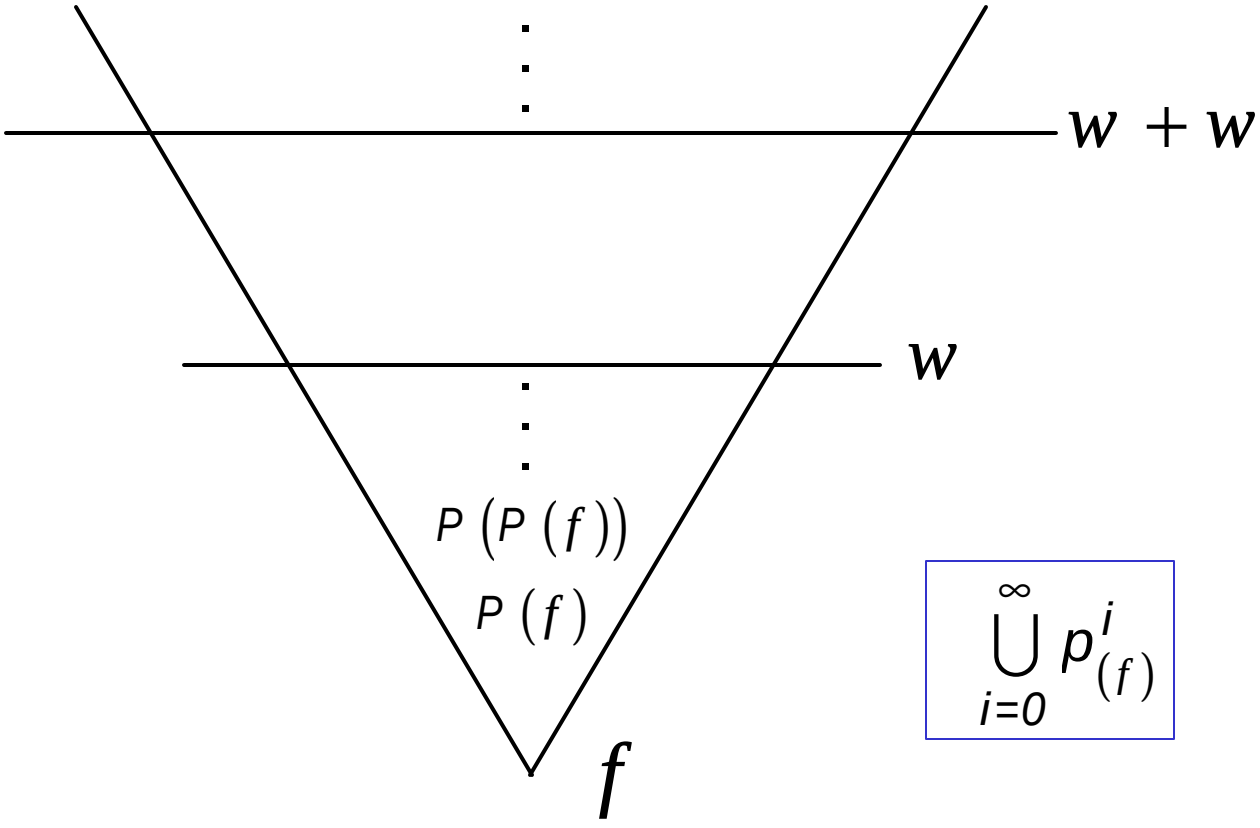
enlarge

Type Theory with Set Terms

$$\frac{a_1, a_2, \ldots, a_i \ldots \qquad a_1, a_2, \ldots, a_i \ldots}{A \quad \hat{g}_A \quad f\ (a) \quad \hat{j}\left(\hat{a}\right)} \qquad f \Downarrow \lambda x.b \quad f \Downarrow \hat{j} \quad c_i\left(\overline{a}\right) \Downarrow c_i\left(\overline{a}\right)$$

$$\lambda x.b \quad a \mapsto b \quad < a, b > \quad < \hat{a}, \hat{b} > \quad c_i\left(\overline{a}\right) \quad c_i\left(\hat{a}\right)$$

The Cumulative Hierarchy of Sets



Captures the intuition of collecting stages and co-finality… unending stages.

Zermelo Fraenkel Set Theory (ZF)

ZF can be axiomatized in 1st order logic by axioms, 2 axiom schemas (separation, replacement). Axiom of choice is a 9th axiom.

The hierarchy is defined as:

$$Z_0 = f$$

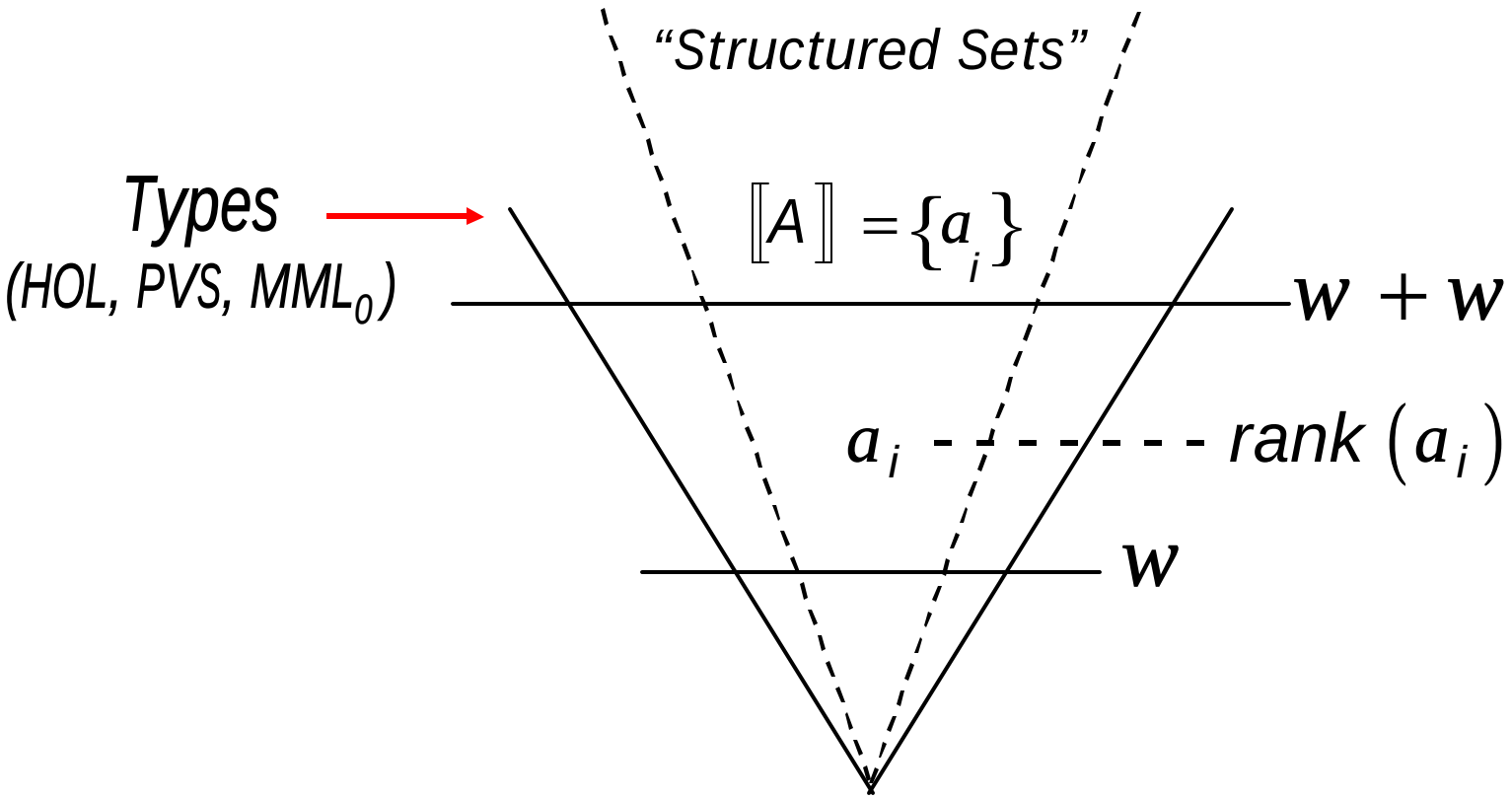
$$Z_a = \bigcup_{b<a} Z_b$$

$$Z_{a+1} = P\left(Z_a\right)$$

$$Z = \bigcup_{a \in Ord} Z_a$$

Every ZF set x is in some Z_α; its rank is the least such ordinal α.

Review of Set Theoretic Semantics



The Cumulative Hierarchy of Sets

Limitations of Set Embedding

Unlike with PVS and HOL, no embedding can cover all of Nuprl because

- Recursive types allow $T = (T \to \mathbb{B}) \to \mathbb{B}$
- Domains all absolutely unsolvable problems

But Howe gets a very substantial part and Moran extends to more. This raises the issue of sub-theories of domains and recursive types.

Goal of Howe's Approach

- Extend the family of "structured sets"
- Create constants for all "structured sets"
- Extend evaluation to all terms
- Define approximation (covering)

$$a_i \lhd a_i$$

to relate sets a_i to terms a_i

Evaluation Rules

$$\frac{f \Downarrow \hat{f} \quad (a, b) ? f \quad a \triangleleft a}{f\ (a) \Downarrow \hat{b}} (ap_f) \qquad \frac{f \Downarrow \lambda x.b \quad b[a/x] \Downarrow v}{f\ (a) \Downarrow v} (ap_\lambda)$$

$$\frac{}{\hat{g} \Downarrow \hat{g}} \qquad \frac{}{\lambda x.b \Downarrow \lambda x.b} \qquad \frac{}{c_i(\overline{a}) \Downarrow c_i(\overline{a})}$$

Approximation Rules

$$\frac{e \Downarrow v \quad a \lhd v}{a \lhd e} \qquad \frac{?j \quad a_j \lhd a_j}{c_i\left(\hat{a}_1, \ldots, \hat{a}_n\right) \lhd c_i\left(\hat{a}_1, \ldots, \hat{a}_n\right)}$$

$$\frac{?(a, b) ? f \quad b \lhd b\left[\hat{a} / x\right]}{f \lhd \lambda x . b}^{\dagger}$$

† Howe does not use this rule because it is unproven when the non-determinism of functions on quotient types is allowed.

Interpreting Sequents

Given a sequent $x_1 : A_1, \ldots, x_n : A_n \vdash g \ ? \ G$,
a closing substitution cl for it is a sequence of closed terms

$$a_1, \ldots, a_n \quad \text{for} \quad x_1, \ldots, x_n .$$

The x_i are the only free variables in g and G.

The sequent is true *iff* $[\![cl\ (g)]\!]_{cl\ (G)}$ is defined and belongs to $[\![cl\ (G)]\!]$ for all closing substitutions cl such that

$$[\![cl\ (x_i)]\!]_{cl\ (A_i)} \ ? \ [\![cl\ (A_i)]\!] .$$

Results - CF

Moran's Theorem

Aczel

[redacted]

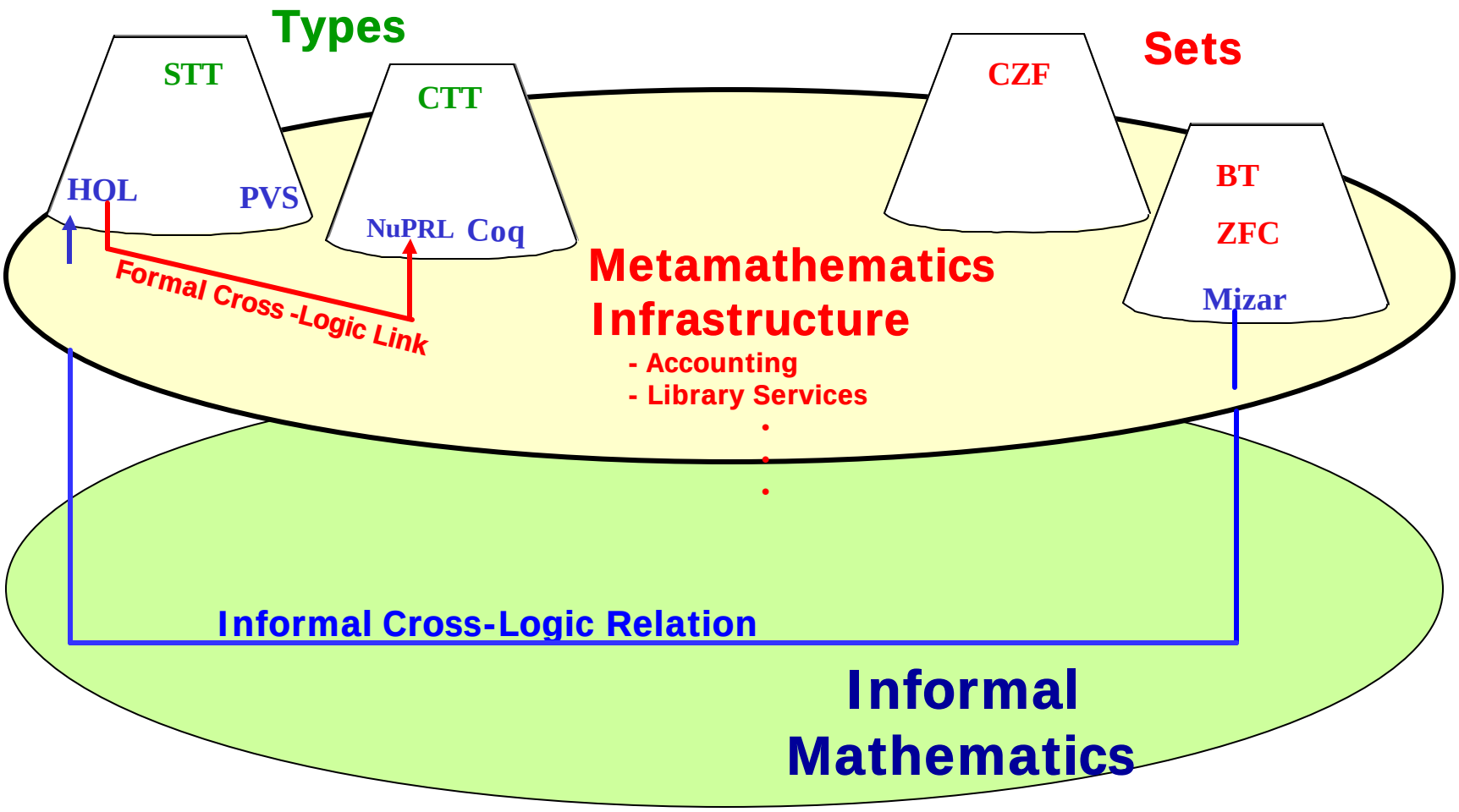
CCTT

CZFC

[redacted]

Howe

H(Acz(M)) @ M



Metamathematical results linking two theories can be accomplished in two ways: an informal one, where the link is established semantically on paper; and a formal proof-theoretic one, which describes mechanisms for translating proofs, etc. The metamathematical infrastructure provides mechanisms to account for both, and the library services to establish the links.

Plan of the Talk

- FDL Role in System Verification
- Features of an FDL
- Theoretical Basis for Sharing
- Conclusion

Here are the elements of our vision for an FDL of the future:

- 100K formally proved theorems in an interconnected web of formal alg math results by 2010, accessible to both people and provers
- Large amounts of algorithmic knowledge in the collection, including verified reference algorithms and reference systems with verified components
- Routinely used in system design, implementation, verification, extension, documentation, and maintenance
- Content accessible from National Science Digital Library (NSDL)
- Proof servers available on the grid for interacting with the FDL at many levels of service, including executing algorithms and proofs
- Library services available from the Web
- Significant discoveries enabled in CS, Mathematics, and other sciences and engineering areas