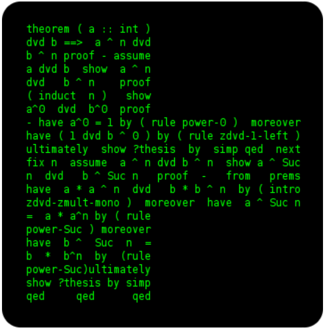
vdash.org:
a formalized math wiki



```
theorem ( a :: int )
dvd b ==>  a ^ n dvd
b ^ n proof - assume
a dvd b  show  a ^ n
dvd   b ^ n    proof
( induct  n )   show
a^0  dvd  b^0  proof
- have a^0 = 1 by ( rule power-0 )  moreover
have ( 1 dvd b ^ 0 ) by ( rule zdvd-1-left )
ultimately  show ?thesis  by  simp qed  next
fix n  assume  a ^ n dvd b ^ n  show a ^ Suc
n  dvd   b ^ Suc n   proof  -   from   prems
have  a * a ^ n  dvd   b * b ^ n  by ( intro
zdvd-zmult-mono )  moreover  have  a ^ Suc n
=  a * a^n by ( rule
power-Suc ) moreover
have  b ^  Suc  n  =
b  *  b^n  by  (rule
power-Suc)ultimately
show ?thesis by simp
qed     qed      qed
```

Cameron Freer
freer@mit.edu

MIT Dept. of Mathematics

MIT E-Club
August 11, 2008

1. What is vdash?

2. Why?

3. Who?
    - Perspectives and communities

4. How?
    - Implementation
    - Screenshot
    - Future Developments
    - Contribute

“vdash?”

The name

\vdash is $\vdash$ in LaTeX.
$S \vdash \varphi$ iff φ can be proved using assumptions from the set S.

“vdash?”

The name

\vdash is $\vdash$ in LaTeX.
$S \vdash \varphi$ iff φ can be proved using assumptions from the set S.

What is vdash?

- vdash is a library of formally verified mathematics that anyone can edit.

“vdash?”

The name

\vdash is $\vdash$ in LaTeX.
$S \vdash \varphi$ iff φ can be proved using assumptions from the set S.

What is vdash?

- vdash is a library of formally verified mathematics that anyone can edit.
- Roughly: vdash = computer proof assistant + library of formalized mathematics + web interface

“vdash?”

The name

\vdash is $\vdash$ in LaTeX.
$S \vdash \varphi$ iff φ can be proved using assumptions from the set S.

What is vdash?

- vdash is a library of formally verified mathematics that anyone can edit.
- Roughly: vdash = computer proof assistant + library of formalized mathematics + web interface
- a math wiki that can be verified all the way down to the most basic results

“vdash?”

The name

\vdash is $\vdash$ in LaTeX.
$S \vdash \varphi$ iff φ can be proved using assumptions from the set S.

What is vdash?

- vdash is a library of formally verified mathematics that anyone can edit.
- Roughly: vdash = computer proof assistant + library of formalized mathematics + web interface
- a math wiki that can be verified all the way down to the most basic results
- one approach to developing the “Math Commons”

The goal

Discovering new mathematics is hard; but all mathematics, once already known, is mechanically verifiable, in principle.

The goal

Discovering new mathematics is hard; but all mathematics, once already known, is mechanically verifiable, in principle.

In practice, this is a pain, and currently only worth the effort in certain special circumstances.

The goal

Discovering new mathematics is hard; but all mathematics, once already known, is mechanically verifiable, in principle.

In practice, this is a pain, and currently only worth the effort in certain special circumstances.

The idea of vdash is to harness the collaborative power of a wiki, while retaining the absolute certainty of a mechanical verifier.

“I demand satisfaction!”

"I demand satisfaction!"

Imagine that Wikipedia could instantly verify every contribution:
"Sorry, this site allows only *true* statements."

“I demand satisfaction!”

Imagine that Wikipedia could instantly verify every contribution:
“Sorry, this site allows only *true* statements.”

(In practice, probably allow any contribution, but indicate level of verification visually – e.g., with different colors of links.)

Why *Formalized* Mathematics?

Right now

- Certainty: growing crisis due to enormous proofs

Why *Formalized* Mathematics?

Right now

- Certainty: growing crisis due to enormous proofs
- Explanation: all the details are there

Why *Formalized* Mathematics?

Right now

- Certainty: growing crisis due to enormous proofs
- Explanation: all the details are there
- Reusability: prove it once, then it becomes modular

Why *Formalized* Mathematics?

Right now

- Certainty: growing crisis due to enormous proofs
- Explanation: all the details are there
- Reusability: prove it once, then it becomes modular
- Instant verification of new results: peer-review still useful for assessing quality, though not as essential for checking

Why *Formalized* Mathematics?

Right now

- Certainty: growing crisis due to enormous proofs
- Explanation: all the details are there
- Reusability: prove it once, then it becomes modular
- Instant verification of new results: peer-review still useful for assessing quality, though not as essential for checking

Much later

- Knowledge base for robot mathematicians

> *"We are not scanning all those books to be read by people," explained one of my hosts after my talk. "We are scanning them to be read by an AI."*
>
> *– George Dyson, on his visit to Google, 2005.*

Why *Formalized* Mathematics?

Right now

- Certainty: growing crisis due to enormous proofs
- Explanation: all the details are there
- Reusability: prove it once, then it becomes modular
- Instant verification of new results: peer-review still useful for assessing quality, though not as essential for checking

Much later

- Knowledge base for robot mathematicians

> *"We are not scanning all those books to be read by people," explained one of my hosts after my talk. "We are scanning them to be read by an AI."*
> *– George Dyson, on his visit to Google, 2005.*

- Tool/interface for human + computer symbiont mathematicians

Biggest stumbling block: intermediate results

Why does LaTeX work?

Most mathematicians now use LaTeX to typeset their mathematics because it is easier than sending texts to secretaries and rechecking. Currently, using proof assistants is much harder than writing informally.

So try to develop libraries, tools, and interfaces to a point where it is easier to formally verify one's own results.

Biggest stumbling block: intermediate results

Why does LaTeX work?

Most mathematicians now use LaTeX to typeset their mathematics because it is easier than sending texts to secretaries and rechecking. Currently, using proof assistants is much harder than writing informally.

So try to develop libraries, tools, and interfaces to a point where it is easier to formally verify one's own results.

Two main cases where formalization occurs today

- carefully curated bottom-up libraries
- vertical projects – do what it takes to get the big result

There are plenty of frameworks for the stuff in between, but it still is just tons of work.

Why a *wiki?*

- Tons more contributors.

Why a *wiki?*

- Tons more contributors.
- Attack from all sides at once: top-down, fill in middle, scrape web to create stubs.

Why a *wiki?*

- Tons more contributors.
- Attack from all sides at once: top-down, fill in middle, scrape web to create stubs.
- See what works, rather than setting up elaborate frameworks first.

Why a *wiki?*

- Tons more contributors.
- Attack from all sides at once: top-down, fill in middle, scrape web to create stubs.
- See what works, rather than setting up elaborate frameworks first.
- It'll be messy, but the formal verification provides a sanity check.

Why a *wiki?*

- Tons more contributors.
- Attack from all sides at once: top-down, fill in middle, scrape web to create stubs.
- See what works, rather than setting up elaborate frameworks first.
- It'll be messy, but the formal verification provides a sanity check.

Hopefully, it is possible to build a community that produces way more formalized mathematics than with existing methods (which are careful, but too slow).

(Over time, ever larger "cores" could be cleaned up and submitted to existing careful forums.)

The Math Commons

Freely available research

- Preprint archives (arXiv, CiteSeer)
- Open access journals

The Math Commons

Freely available research

- Preprint archives (arXiv, CiteSeer)
- Open access journals

Open content movements

- Science Commons
    - shared building-blocks for other collaborative sciences
- Creative Commons
- Free Culture

The *Formal* Math Encyclopedia that anyone can edit

- Wikipedia
- PlanetMath
- MathWorld

Better search through formalization

- Mathematics Knowledge Management
- Semantic Web

Existing formalization efforts

- Isabelle: Archive of Formal Proofs, IsarMathLib
- Mizar: Journal of Formalized Mathematics
- Coq: Coq Contribs, Hypertextual Electronic Library of Mathematics
- HOL: HOL Light examples

Implementation of prototype (somewhat tentative)

Isabelle/Isar:

- free software
- small trusted core
- human readable – looks like real mathematics

Implementation of prototype (somewhat tentative)

Isabelle/Isar:

- free software
- small trusted core
- human readable – looks like real mathematics

import IsarMathLib

Implementation of prototype (somewhat tentative)

Isabelle/Isar:

- free software
- small trusted core
- human readable – looks like real mathematics

import IsarMathLib

MediaWiki

Implementation of prototype (somewhat tentative)

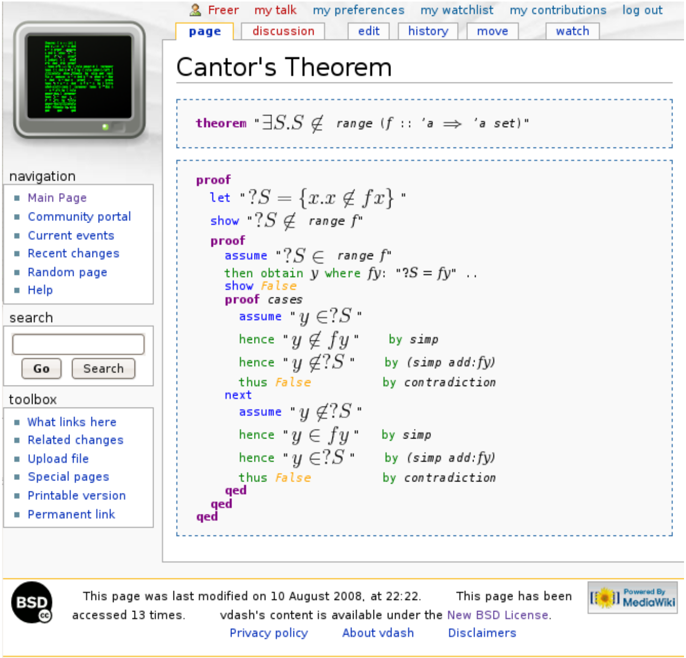
Isabelle/Isar:

- free software
- small trusted core
- human readable – looks like real mathematics

import IsarMathLib

MediaWiki

contributions under new BSD license



Cantor's Theorem

```
theorem "∃S. S ∉ range (f :: 'a ⇒ 'a set)"
```

```
proof
  let "?S = {x. x ∉ f x} "
  show "?S ∉ range f"
  proof
    assume "?S ∈ range f"
    then obtain y where fy: "?S = f y" ..
    show False
    proof cases
      assume "y ∈ ?S"
      hence "y ∉ f y"    by simp
      hence "y ∉ ?S"     by (simp add: fy)
      thus False         by contradiction
    next
      assume "y ∉ ?S"
      hence "y ∈ f y"    by simp
      hence "y ∈ ?S"     by (simp add: fy)
      thus False         by contradiction
    qed
  qed
qed
```

This page was last modified on 10 August 2008, at 22:22. This page has been accessed 13 times. vdash's content is available under the New BSD License.

Privacy policy About vdash Disclaimers

Future features (not that the main site is ready yet!)

Scrape web for mathematics

- arXMLiv, CiteSeer, other preprints
- open-source math textbooks

Future features (not that the main site is ready yet!)

Scrape web for mathematics

- arXMLiv, CiteSeer, other preprints
- open-source math textbooks

Other proof assistants/libraries

- Coq/HELM, Mizar/MML, many others

Future features (not that the main site is ready yet!)

Scrape web for mathematics

- arXMLiv, CiteSeer, other preprints
- open-source math textbooks

Other proof assistants/libraries

- Coq/HELM, Mizar/MML, many others

semantic metadata and search

- OMDoc/OpenMath/MathML

Future features (not that the main site is ready yet!)

Scrape web for mathematics

- arXMLiv, CiteSeer, other preprints
- open-source math textbooks

Other proof assistants/libraries

- Coq/HELM, Mizar/MML, many others

semantic metadata and search

- OMDoc/OpenMath/MathML

clientside AJAX interface

- interactive verification
- better GUI
- reduce server load

Extensions and spinoffs

Education

Build open math textbooks, any proof in which can be “unfolded” all the way to the bottom

Extensions and spinoffs

Education

Build open math textbooks, any proof in which can be "unfolded" all the way to the bottom

General public

The Theorome Project: visualize and browse interconnections within mathematics

Extensions and spinoffs

Education

Build open math textbooks, any proof in which can be “unfolded” all the way to the bottom

General public

The Theorome Project: visualize and browse interconnections within mathematics

Certified calculations and visualizations

- Computer algebra systems
- Graphing calculators and other technical graphics

New Domains

Formal science and engineering

- Computer Science: algorithms, programming languages, protocols
- Mathematical Physics

New Domains

Formal science and engineering

- Computer Science: algorithms, programming languages, protocols
- Mathematical Physics

Industrial applications

- software and hardware verification
- more modular code

Doing the same thing with new content, or reusing building-blocks?

How you can help

Contribute math to the wiki

- formalize the statement of a theorem
- fill in proof details informally
- formalize details
- correct errors; make it prettier/idiomatic

How you can help

Contribute math to the wiki

- formalize the statement of a theorem
- fill in proof details informally
- formalize details
- correct errors; make it prettier/idiomatic

Back-end design

- versioning and dependencies of math code itself
- integrate bugtracker with Talk pages
- semantic metadata

How you can help

Human interface

- interactive proof assistant via AJAX
- better search
- browse interconnections between results

How you can help

Human interface

- interactive proof assistant via AJAX
- better search
- browse interconnections between results

Interface to other projects

- web scrapers and internal bots
  - create vdash stubs
  - link elsewhere from vdash pages
- link from existing content to vdash pages

How you can help

Human interface

- interactive proof assistant via AJAX
- better search
- browse interconnections between results

Interface to other projects

- web scrapers and internal bots
  - create vdash stubs
  - link elsewhere from vdash pages
- link from existing content to vdash pages

Challenges

- Organize challenge projects and competitions to build up content and get people excited.

Feedback

Advice/ideas/suggestions: `freer@mit.edu`

Thanks!