Is truth futureproof?

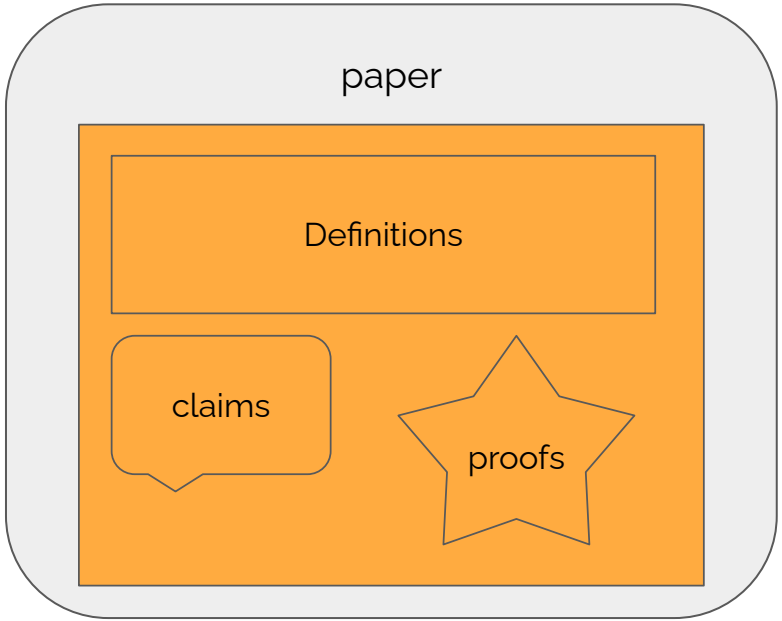
On the possible futures of mechanized proofs

Chris Martens
Emma Tosch
Elan Semenova
Cynthia Li

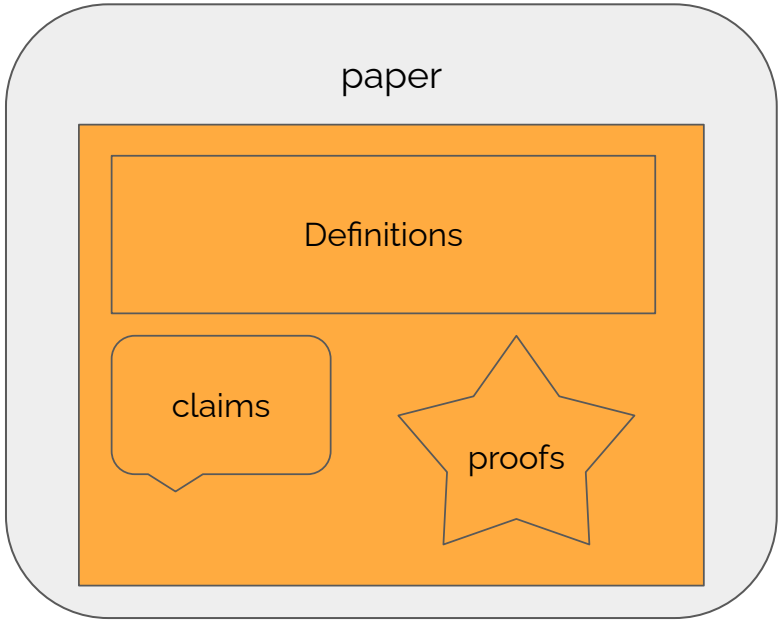
March 10, 2026
PLATEAU, Pittsburgh, PA

Context

Context: recording mathematical knowledge for future benefit



Knowledge artifacts



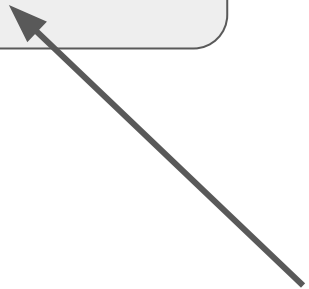
Available for:

- Peer review
- Distribution
- Independent validation
- Training future generations
- A part of human culture!

SCS Technical Report library

Mechanizing mathematical knowledge

Mechanizing mathematical knowledge



aka “formalizing”

Proof assistants

Show of hands:

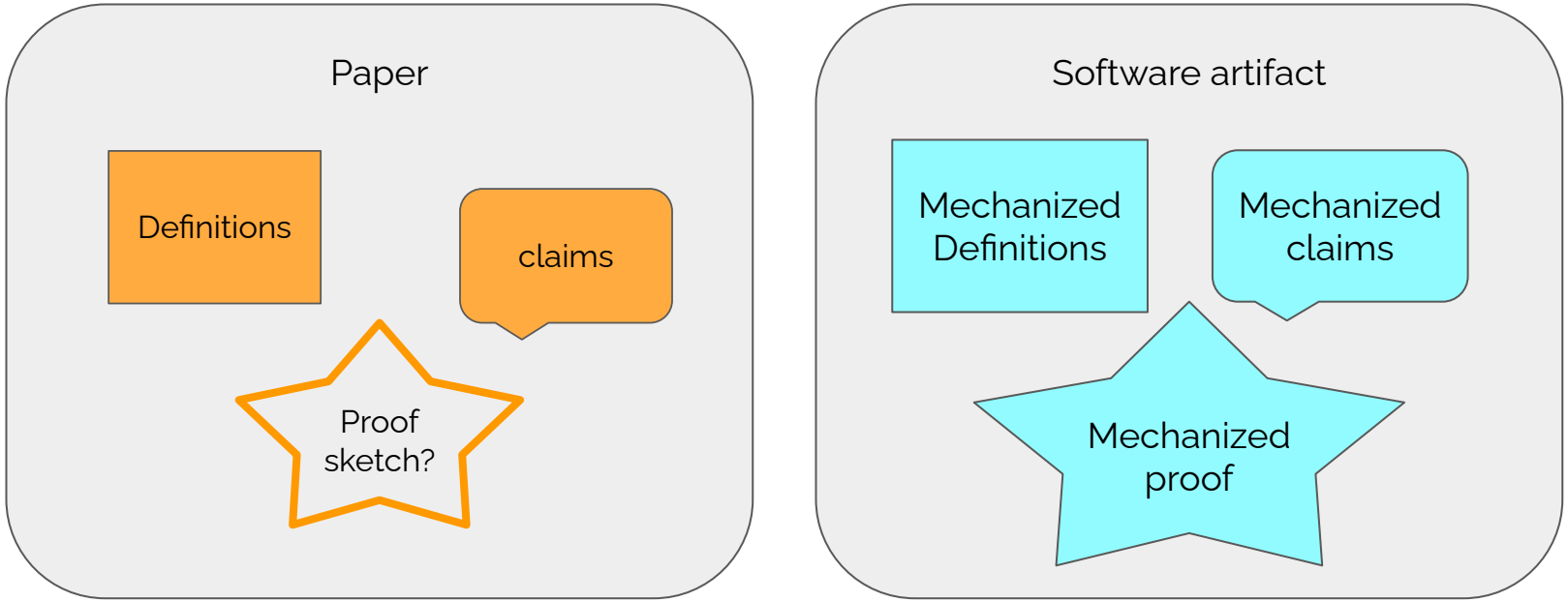
Who has used a proof assistant for research, education, or entertainment?

Proof assistants

Proof assistance

- **Scaffolding** (e.g. case enumeration)
- **Track proof state**: assumptions, obligations
- **Automation** (search/synthesis of proof)
- **Checking validity** incrementally
- **Independent verification** after-the-fact

Mechanized knowledge artifacts



Show of hands:

Have you ever tried to read or run someone else's mechanized development from more than 5 years ago?

Show of hands:

Have you ever tried to read or run someone else's mechanized development from more than 5 years ago?

Keep your hands up if you succeeded.

Provocation:

When a mechanized proof breaks,
is the theorem still true?

More careful question:

How does mechanization present distinct **opportunities** and **challenges** for **stewarding mathematical knowledge** across generations?

Outline

1. **Why do we mechanize?**
   Articulate distinct social practices of proof and where they are in tension

2. **How can we future-proof mechanized proofs?**
   Describe challenges, current approaches, and visions for the future

Why do we mechanize?

Anecdote: attitudes toward automation

On replacing 60-line Lean tactic script with “grind” (automation):

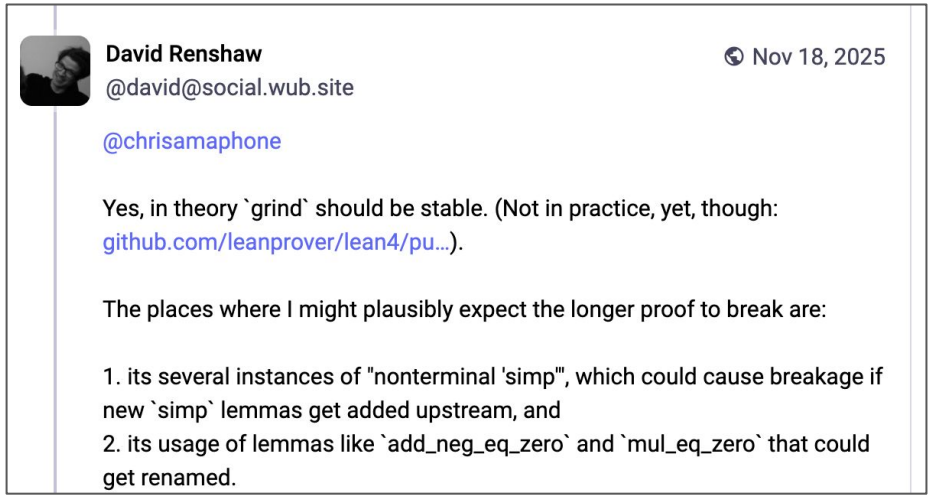
Is this better?

It’s easier to maintain!

https://hci.social/@chrisamaphone/115571191055573371

Anecdote: attitudes toward automation

On replacing 60-line Lean tactic script with “grind” (automation):



David Renshaw
@david@social.wub.site

Nov 18, 2025

@chrisamaphone

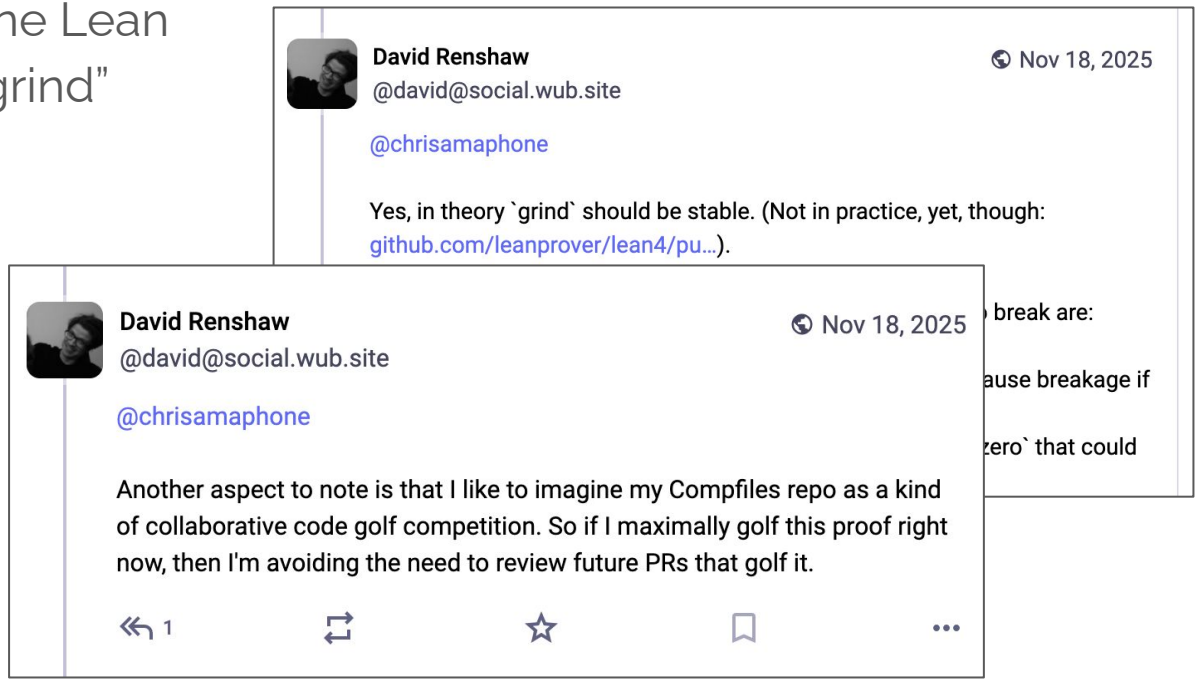
Yes, in theory `grind` should be stable. (Not in practice, yet, though: github.com/leanprover/lean4/pu…).

The places where I might plausibly expect the longer proof to break are:

1. its several instances of "nonterminal 'simp'", which could cause breakage if new `simp` lemmas get added upstream, and
2. its usage of lemmas like `add_neg_eq_zero` and `mul_eq_zero` that could get renamed.

Anecdote: attitudes toward automation

On replacing 60-line Lean tactic script with “grind” (automation):



Anecdote: attitudes toward automation

2024: Collaborative project to mechanize Viazovska et al. (2022) sphere-packing results
2026: ~200,000 line "autoformalization" published by independent group

"The formalization, on its own, is close to worthless, since **the correctness of Viazovska's result was never in doubt**. The participants embraced the project, rather, as a way of revisiting those results and **better understanding them**, and of **building libraries and infrastructure to support future work.**"

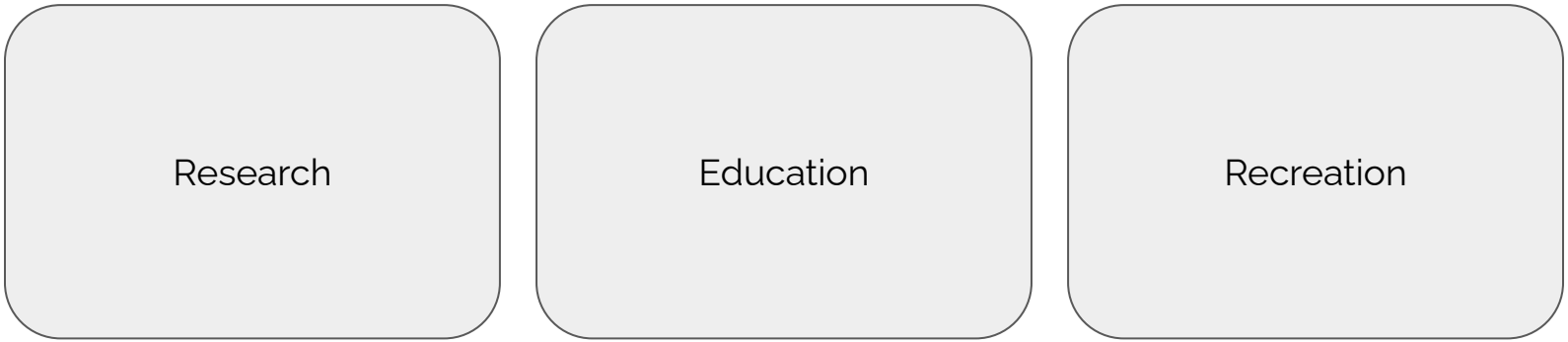
Jeremy Avigad
"Mathematicians in the Age of AI"
https://arxiv.org/pdf/2603.03684

Anecdote: proof assistant as tool for thought

> “Four of my PhD students formalized their theses in Agda [...] but it feels wrong to say they formalized their theses. In reality, **Agda worked as a blackboard to develop and test their thoughts.**
>
> [...] And it is in blackboards that computer scientists/mathematicians **come up with their ideas** that eventually become preprints and published papers.”

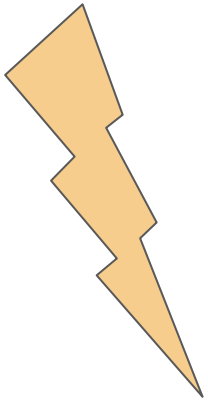
https://mathstodon.xyz/@MartinEscardo/116167281653220814

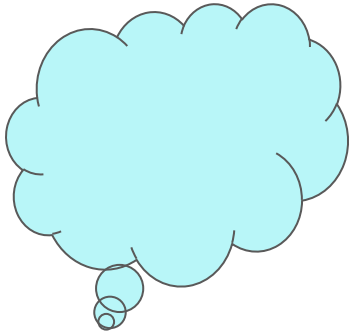
Contexts



Proof as social process***es***

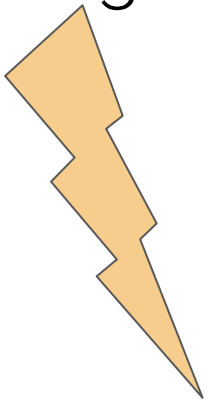
Convincing and *explaining*



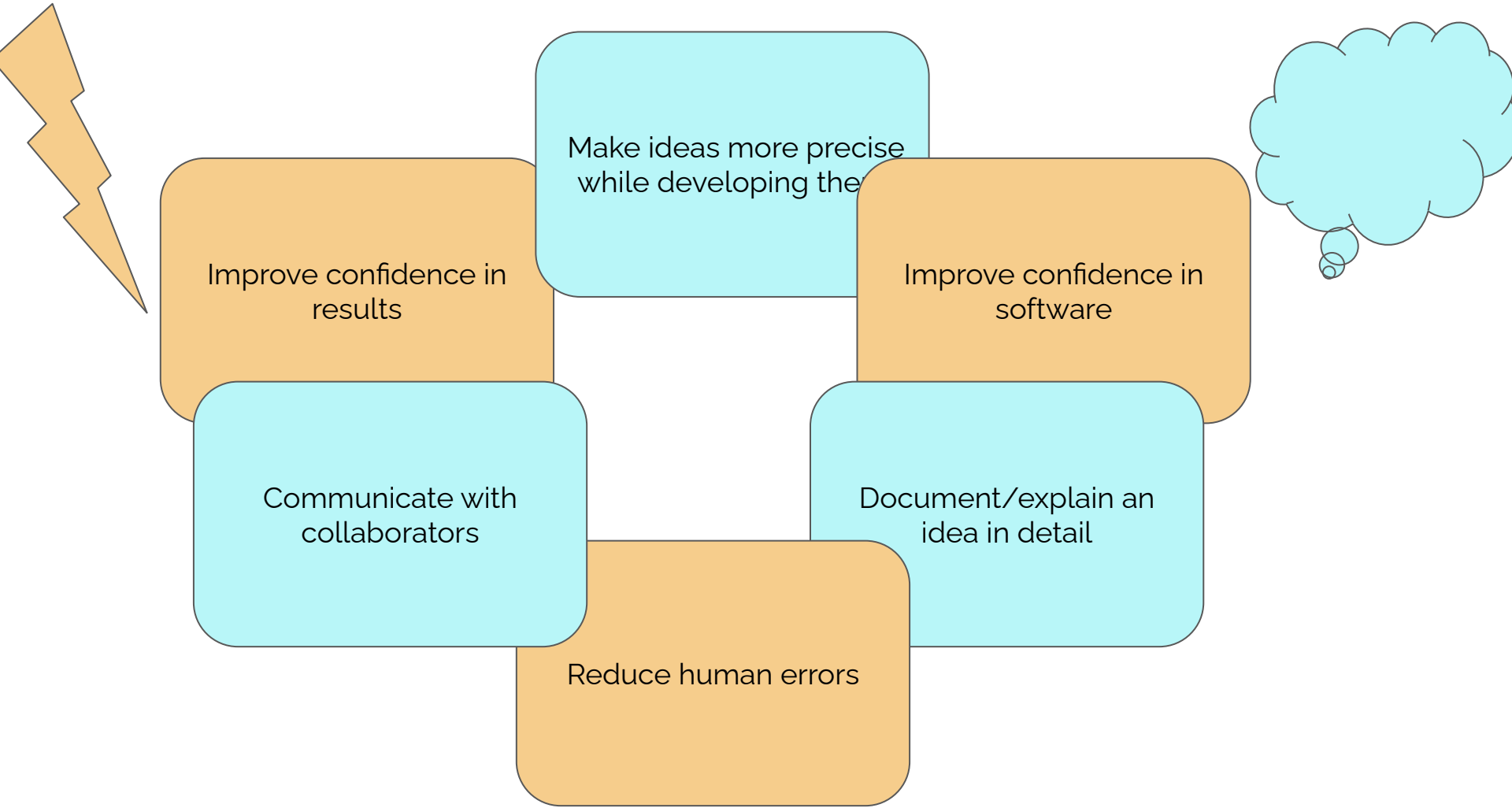


Convincing and explaining

Improve confidence **that** something is true



Understand **why** something is true


Make ideas more precise while developing the
Improve confidence in results
Improve confidence in software
Communicate with collaborators
Document/explain an idea in detail
Reduce human errors

Discussion question

Why do **you** mechanize your (meta)theory?

Do you gravitate towards one of convincing/explaining as a default mode?

What does it mean to future-proof mechanized proofs?

Collected desiderata

(Covered in the paper)

- **Peer review and curation** of proof artifacts
- **Archival storage**
- **Indexing** for search, citation, retrieval
- **Running** proofs (running ITPs to check old proofs)
- **Comprehending** proofs (including stepping/interacting)
- **Adapting** old proofs to new contexts
    - Research project, classroom, different ITP tool, etc.

Collected desiderata

(Covered in the paper)

- **Peer review and curation** of proof artifacts
- **Archival storage**
- **Indexing** for search, citation, retrieval
- **Running** proofs (running ITPs to check old proofs)
- **Comprehending** proofs (including stepping/interacting)
- **Adapting** old proofs to new contexts
    - Research project, classroom, different ITP tool, etc.

Collected desiderata

(Covered in the paper)

- **Peer review and curation** of proof artifacts
- **Archival storage**
- **Indexing** for search, citation, retrieval
- **Running** proofs (running ITPs to check old proofs)
- **Comprehending** proofs (including stepping/interacting)
- **Adapting** old proofs to new contexts
  - Research project, classroom, different ITP tool, etc.

Discussion question

What kinds of “proof artifacts” we should store?

What is a stored (archival) proof object?

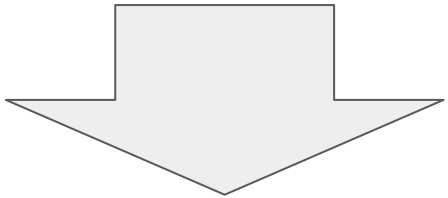
Direct proof term? Tactic script? (shown: Lean)

```
1 def distrib (A B C : Prop) : A ∧ (B ∨ C) → (A ∧ B) ∨ (A ∧ C)
2   := fun assm =>
3         match (And.right assm) with
4         | Or.inl b => Or.inl (And.intro (And.left assm) b)
5         | Or.inr c => Or.inr (And.intro (And.left assm) c)
6
7
```

```
1 def distrib (A B C : Prop) : A ∧ (B ∨ C) → (A ∧ B) ∨ (A ∧ C)
2   := by
3       intro assm
4       apply Or.elim (And.right assm)
5       . intro b_hyp
6         apply Or.inl
7         apply And.intro
8         . apply And.left assm
9         . exact b_hyp
10      . intro c_hyp
11        apply Or.inr
12        apply And.intro
13        . apply And.left assm
14        . exact c_hyp
```

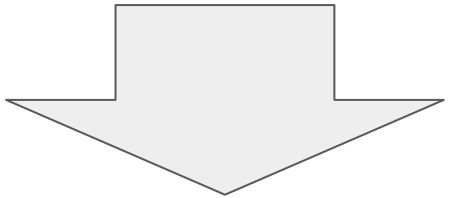
Compiled “core language” term?

```
1 def distrib (A B C : Prop) : A ∧ (B ∨ C) → (A ∧ B) ∨ (A ∧ C)
2   := fun assm =>
3         match (And.right assm) with
4         | Or.inl b => Or.inl (And.intro (And.left assm) b)
5         | Or.inr c => Or.inr (And.intro (And.left assm) c)
6
7
```



Lean [kernel]

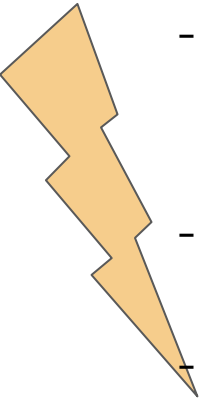
```
1 open import Data.Product
2 open import Data.Sum
3
4 distribute : {A B C : Set} → A × (B ⊎ C) → (A × B) ⊎ (A × C)
5 distribute (a , inj₁ b) = inj₁ (a , b)
6 distribute (a , inj₂ c) = inj₂ (a , c)
```



???

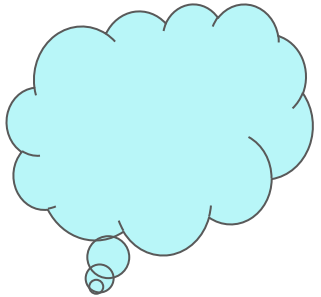
(see https://jesper.cx/posts/agda-core.html)

Needs for future-proofing



Convincing:

- **Maintain checkability**
  Limit dependencies/use of unstable features
- **Maintain links to human-readable claims**
- **Maintain soundness of proof assistant**
  - Small “trusted core”

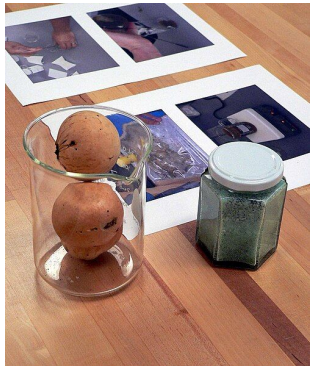


Explaining:

- **Maintain reusable libraries of abstractions** decoupled from specific developments
- **Maintain links to human-readable explanations**
- **Maintain legibility of proofs**
  - Structure of argument
  - Re-runnable interactive process

Archiving media

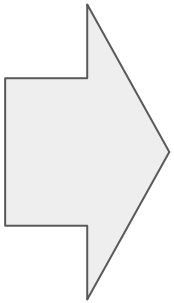
Print/text media:
Paper rots
Ink fades



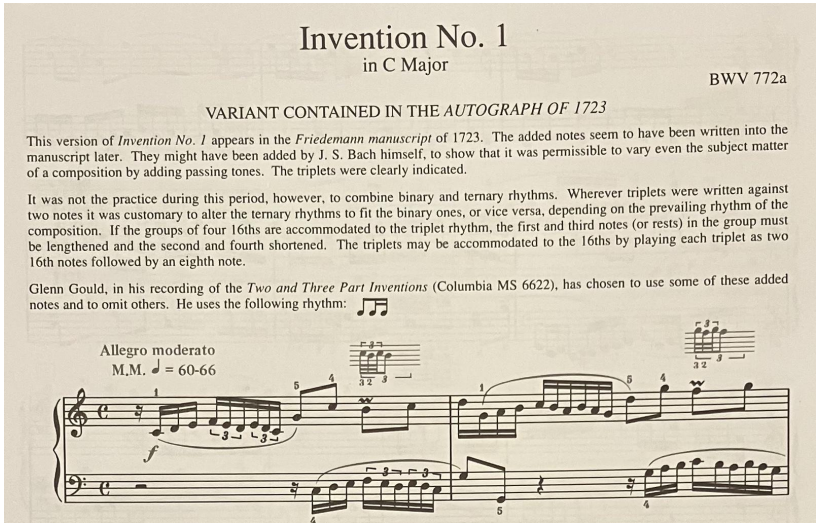
https://en.wikipedia.org/wiki/Iron_gall_ink

Lightfast inks
Printing press
Manual transcription
Stewardship/care

Informal notation; Ambiguity



Formalization;
Standardization;
Transcription;
Annotation;
Stewardship/care



Invention No. 1
in C Major

BWV 772a

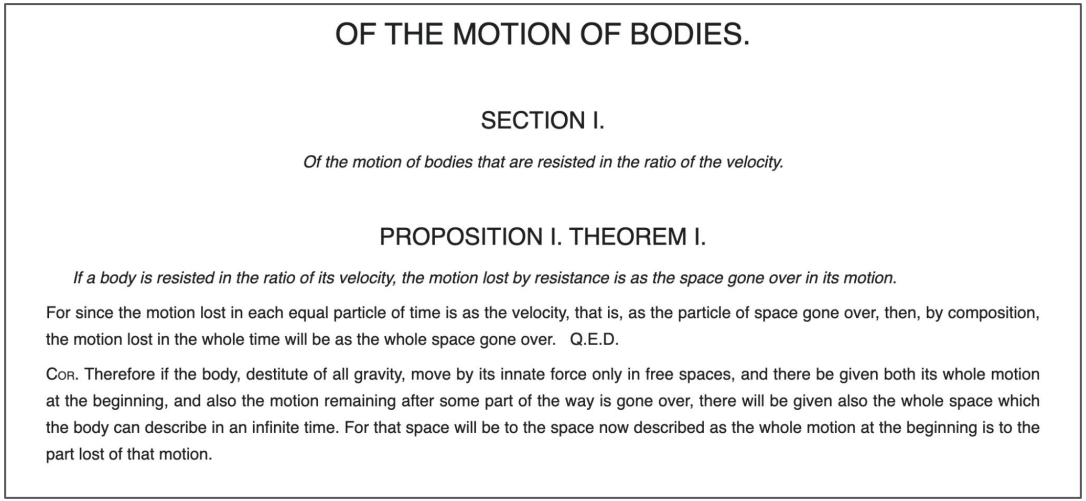
VARIANT CONTAINED IN THE *AUTOGRAPH OF 1723*

This version of *Invention No. 1* appears in the *Friedemann manuscript* of 1723. The added notes seem to have been written into the manuscript later. They might have been added by J. S. Bach himself, to show that it was permissible to vary even the subject matter of a composition by adding passing tones. The triplets were clearly indicated.

It was not the practice during this period, however, to combine binary and ternary rhythms. Wherever triplets were written against two notes it was customary to alter the ternary rhythms to fit the binary ones, or vice versa, depending on the prevailing rhythm of the composition. If the groups of four 16ths are accommodated to the triplet rhythm, the first and third notes (or rests) in the group must be lengthened and the second and fourth shortened. The triplets may be accommodated to the 16ths by playing each triplet as two 16th notes followed by an eighth note.

Glenn Gould, in his recording of the *Two and Three Part Inventions* (Columbia MS 6622), has chosen to use some of these added notes and to omit others. He uses the following rhythm:

Mathematical notation as technology



OF THE MOTION OF BODIES.

SECTION I.

Of the motion of bodies that are resisted in the ratio of the velocity.

PROPOSITION I. THEOREM I.

If a body is resisted in the ratio of its velocity, the motion lost by resistance is as the space gone over in its motion.

For since the motion lost in each equal particle of time is as the velocity, that is, as the particle of space gone over, then, by composition, the motion lost in the whole time will be as the whole space gone over. Q.E.D.

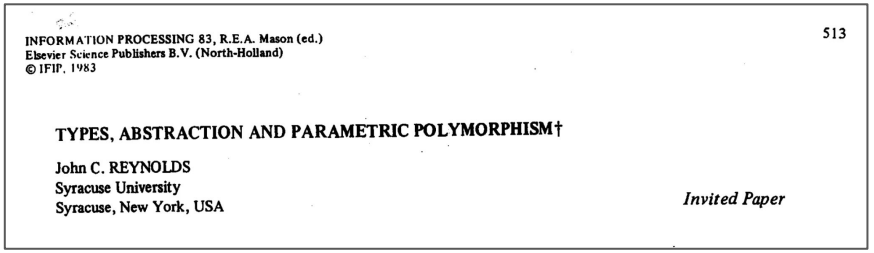
Cor. Therefore if the body, destitute of all gravity, move by its innate force only in free spaces, and there be given both its whole motion at the beginning, and also the motion remaining after some part of the way is gone over, there will be given also the whole space which the body can describe in an infinite time. For that space will be to the space now described as the whole motion at the beginning is to the part lost of that motion.

https://en.wikisource.org/wiki/The_Mathematical_Principles_of_Natural_Philosophy_(1846)/BookII-I

Digitization:

Typesetting languages

Interoperable standards and formats (unicode; pdf; pandoc)

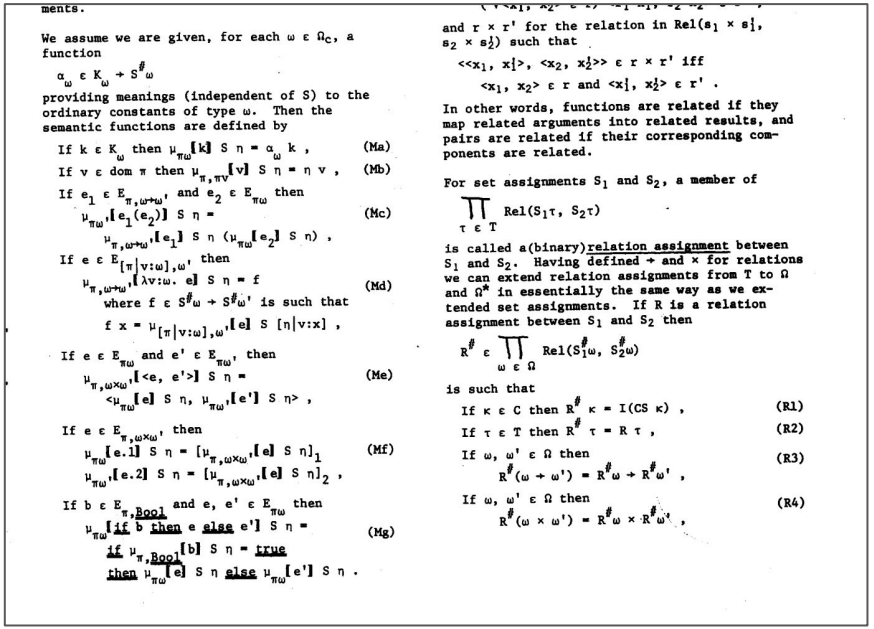


INFORMATION PROCESSING 83, R.E.A. Mason (ed.)
Elsevier Science Publishers B.V. (North-Holland)

TYPES, ABSTRACTION AND PARAMETRIC POLYMORPHISM†

John C. REYNOLDS
Syracuse University
Syracuse, New York, USA

Invited Paper



ments.

We assume we are given, for each $\omega \in \Omega_c$, a function

$$\alpha_\omega \in K_\omega \to S^\#\omega$$

providing meanings (independent of S) to the ordinary constants of type ω. Then the semantic functions are defined by

$$\text{If } k \in K_\omega \text{ then } \mu_{\pi\omega}[\![k]\!]\, S\, \eta = \alpha_\omega\, k\,, \tag{Ma}$$

$$\text{If } v \in \text{dom } \pi \text{ then } \mu_{\pi,\pi v}[\![v]\!]\, S\, \eta = \eta\, v\,, \tag{Mb}$$

$$\text{If } e_1 \in E_{\pi,\omega\to\omega'} \text{ and } e_2 \in E_{\pi\omega} \text{ then } \mu_{\pi\omega'}[\![e_1(e_2)]\!]\, S\, \eta = \mu_{\pi,\omega\to\omega'}[\![e_1]\!]\, S\, \eta\, (\mu_{\pi\omega}[\![e_2]\!]\, S\, \eta)\,, \tag{Mc}$$

$$\text{If } e \in E_{[\pi|v:\omega],\omega'} \text{ then } \mu_{\pi,\omega\to\omega'}[\![\lambda v{:}\omega.\ e]\!]\, S\, \eta = f \text{ where } f \in S^\#\omega \to S^\#\omega' \text{ is such that } f\, x = \mu_{[\pi|v:\omega],\omega'}[\![e]\!]\, S\, [\eta|v:x]\,, \tag{Md}$$

$$\text{If } e \in E_{\pi\omega} \text{ and } e' \in E_{\pi\omega'} \text{ then } \mu_{\pi,\omega\times\omega'}[\![\langle e, e'\rangle]\!]\, S\, \eta = \langle \mu_{\pi\omega}[\![e]\!]\, S\, \eta,\ \mu_{\pi\omega'}[\![e']\!]\, S\, \eta\rangle\,, \tag{Me}$$

$$\text{If } e \in E_{\pi,\omega\times\omega'} \text{ then } \mu_{\pi\omega}[\![e.1]\!]\, S\, \eta = [\mu_{\pi,\omega\times\omega'}[\![e]\!]\, S\, \eta]_1 \quad \mu_{\pi\omega'}[\![e.2]\!]\, S\, \eta = [\mu_{\pi,\omega\times\omega'}[\![e]\!]\, S\, \eta]_2\,, \tag{Mf}$$

$$\text{If } b \in E_{\pi,\underline{\text{Bool}}} \text{ and } e, e' \in E_{\pi\omega} \text{ then } \mu_{\pi\omega}[\![\underline{\text{if}}\ b\ \underline{\text{then}}\ e\ \underline{\text{else}}\ e']\!]\, S\, \eta = \underline{\text{if}}\ \mu_{\pi,\underline{\text{Bool}}}[\![b]\!]\, S\, \eta = \underline{\text{true}}\ \underline{\text{then}}\ \mu_{\pi\omega}[\![e]\!]\, S\, \eta\ \underline{\text{else}}\ \mu_{\pi\omega}[\![e']\!]\, S\, \eta\,. \tag{Mg}$$

and $r \times r'$ for the relation in $\text{Rel}(s_1 \times s_1', s_2 \times s_2')$ such that

$$\langle\langle x_1, x_1'\rangle, \langle x_2, x_2'\rangle\rangle \in r \times r' \text{ iff } \langle x_1, x_2\rangle \in r \text{ and } \langle x_1', x_2'\rangle \in r'\,.$$

In other words, functions are related if they map related arguments into related results, and pairs are related if their corresponding components are related.

For set assignments S_1 and S_2, a member of

$$\prod_{\tau \in T} \text{Rel}(S_1\tau, S_2\tau)$$

is called a (binary) relation assignment between S_1 and S_2. Having defined → and × for relations we can extend relation assignments from T to Ω and Ω^* in essentially the same way as we extended set assignments. If R is a relation assignment between S_1 and S_2 then

$$R^\# \in \prod_{\omega \in \Omega} \text{Rel}(S_1^\#\omega, S_2^\#\omega)$$

is such that

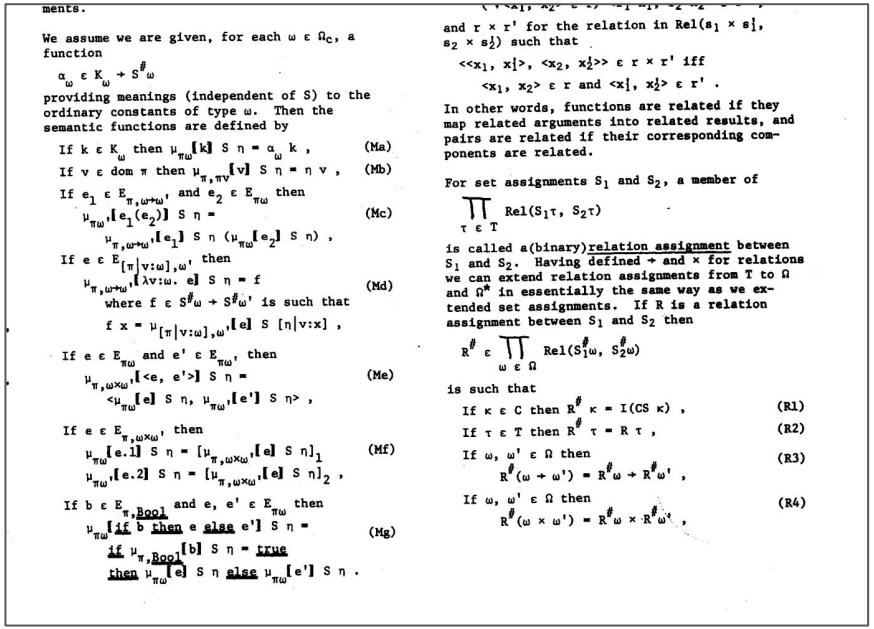
$$\text{If } \kappa \in C \text{ then } R^\#\kappa = I(CS\,\kappa)\,, \tag{R1}$$

$$\text{If } \tau \in T \text{ then } R^\#\tau = R\,\tau\,, \tag{R2}$$

$$\text{If } \omega, \omega' \in \Omega \text{ then } R^\#(\omega \to \omega') = R^\#\omega \to R^\#\omega'\,, \tag{R3}$$

$$\text{If } \omega, \omega' \in \Omega \text{ then } R^\#(\omega \times \omega') = R^\#\omega \times R^\#\omega'\,, \tag{R4}$$

Stewardship, care, transcription



ments.

We assume we are given, for each $\omega \in \Omega_C$, a function

$$\alpha_\omega \in K_\omega \to S^\#\omega$$

providing meanings (independent of S) to the ordinary constants of type ω. Then the semantic functions are defined by

$$\text{If } k \in K_\omega \text{ then } \mu_{\pi\omega}[\![k]\!]\ S\ \eta = \alpha_\omega\ k, \tag{Ma}$$

$$\text{If } v \in \text{dom } \pi \text{ then } \mu_{\pi,\pi v}[\![v]\!]\ S\ \eta = \eta\ v, \tag{Mb}$$

$$\text{If } e_1 \in E_{\pi,\omega\to\omega'} \text{ and } e_2 \in E_{\pi\omega} \text{ then} \quad \mu_{\pi\omega'}[\![e_1(e_2)]\!]\ S\ \eta = \mu_{\pi,\omega\to\omega'}[\![e_1]\!]\ S\ \eta\ (\mu_{\pi\omega}[\![e_2]\!]\ S\ \eta), \tag{Mc}$$

$$\text{If } e \in E_{[\pi|v:\omega],\omega'} \text{ then} \quad \mu_{\pi,\omega\to\omega'}[\![\lambda v{:}\omega.\ e]\!]\ S\ \eta = f \text{ where } f \in S^\#\omega \to S^\#\omega' \text{ is such that } f\ x = \mu_{[\pi|v:\omega],\omega'}[\![e]\!]\ S\ [\eta|v:x], \tag{Md}$$

$$\text{If } e \in E_{\pi\omega} \text{ and } e' \in E_{\pi\omega'} \text{ then} \quad \mu_{\pi,\omega\times\omega'}[\![\langle e, e'\rangle]\!]\ S\ \eta = \langle \mu_{\pi,\omega}[\![e]\!]\ S\ \eta, \mu_{\pi,\omega'}[\![e']\!]\ S\ \eta\rangle, \tag{Me}$$

$$\text{If } e \in E_{\pi,\omega\times\omega'} \text{ then} \quad \mu_{\pi\omega}[\![e.1]\!]\ S\ \eta = [\mu_{\pi,\omega\times\omega'}[\![e]\!]\ S\ \eta]_1 \quad \mu_{\pi\omega}[\![e.2]\!]\ S\ \eta = [\mu_{\pi,\omega\times\omega'}[\![e]\!]\ S\ \eta]_2, \tag{Mf}$$

$$\text{If } b \in E_{\pi,\text{Bool}} \text{ and } e, e' \in E_{\pi\omega} \text{ then} \quad \mu_{\pi\omega}[\![\text{if } b \text{ then } e \text{ else } e']\!]\ S\ \eta = \text{if } \mu_{\pi,\text{Bool}}[\![b]\!]\ S\ \eta = \text{true} \text{ then } \mu_{\pi,\omega}[\![e]\!]\ S\ \eta \text{ else } \mu_{\pi,\omega}[\![e']\!]\ S\ \eta. \tag{Mg}$$

and $r \times r'$ for the relation in $\text{Rel}(s_1 \times s_1', s_2 \times s_2')$ such that

$$\langle\langle x_1, x_1'\rangle, \langle x_2, x_2'\rangle\rangle \in r \times r' \text{ iff } \langle x_1, x_2\rangle \in r \text{ and } \langle x_1', x_2'\rangle \in r'.$$

In other words, functions are related if they map related arguments into related results, and pairs are related if their corresponding components are related.

For set assignments S_1 and S_2, a member of

$$\prod_{\tau \in T} \text{Rel}(S_1\tau, S_2\tau)$$

is called a (binary) relation assignment between S_1 and S_2. Having defined $\to$ and $\times$ for relations we can extend relation assignments from T to Ω and Ω^* in essentially the same way as we extended set assignments. If R is a relation assignment between S_1 and S_2 then

$$R^\# \in \prod_{\omega \in \Omega} \text{Rel}(S_1^\#\omega, S_2^\#\omega)$$

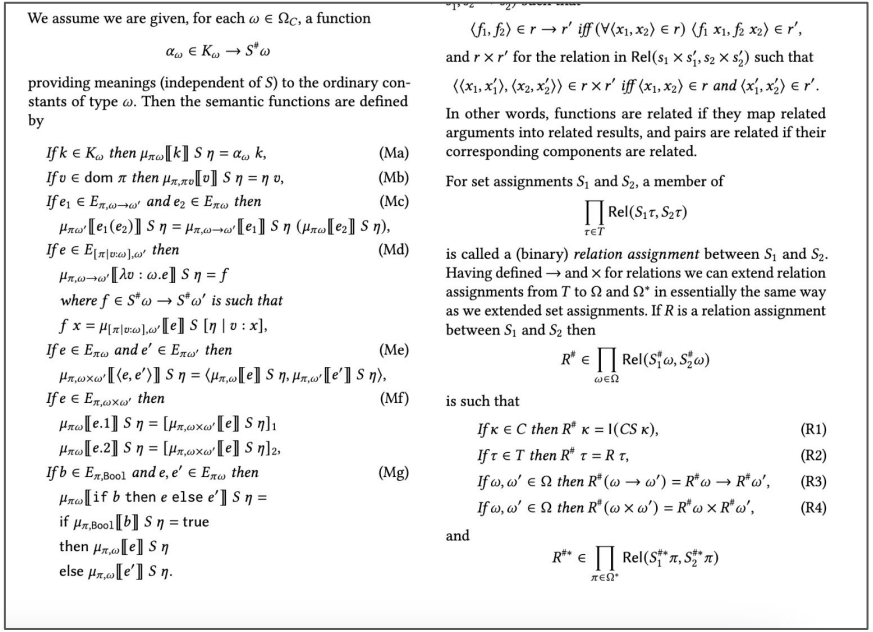
is such that

$$\text{If } \kappa \in C \text{ then } R^\#\ \kappa = \text{I}(CS\ \kappa), \tag{R1}$$

$$\text{If } \tau \in T \text{ then } R^\#\ \tau = R\ \tau, \tag{R2}$$

$$\text{If } \omega, \omega' \in \Omega \text{ then } R^\#(\omega \to \omega') = R^\#\omega \to R^\#\omega', \tag{R3}$$

$$\text{If } \omega, \omega' \in \Omega \text{ then } R^\#(\omega \times \omega') = R^\#\omega \times R^\#\omega', \tag{R4}$$



We assume we are given, for each $\omega \in \Omega_C$, a function

$$\alpha_\omega \in K_\omega \to S^\#\omega$$

providing meanings (independent of S) to the ordinary constants of type ω. Then the semantic functions are defined by

$$\text{If } k \in K_\omega \text{ then } \mu_{\pi\omega}[\![k]\!]\ S\ \eta = \alpha_\omega\ k, \tag{Ma}$$

$$\text{If } v \in \text{dom } \pi \text{ then } \mu_{\pi,\pi v}[\![v]\!]\ S\ \eta = \eta\ v, \tag{Mb}$$

$$\text{If } e_1 \in E_{\pi,\omega\to\omega'} \text{ and } e_2 \in E_{\pi\omega} \text{ then} \quad \mu_{\pi\omega'}[\![e_1(e_2)]\!]\ S\ \eta = \mu_{\pi,\omega\to\omega'}[\![e_1]\!]\ S\ \eta\ (\mu_{\pi\omega}[\![e_2]\!]\ S\ \eta), \tag{Mc}$$

$$\text{If } e \in E_{[\pi|v:\omega],\omega'} \text{ then} \quad \mu_{\pi,\omega\to\omega'}[\![\lambda v : \omega.e]\!]\ S\ \eta = f \text{ where } f \in S^\#\omega \to S^\#\omega' \text{ is such that } f\ x = \mu_{[\pi|v:\omega],\omega'}[\![e]\!]\ S\ [\eta \mid v : x], \tag{Md}$$

$$\text{If } e \in E_{\pi\omega} \text{ and } e' \in E_{\pi\omega'} \text{ then} \quad \mu_{\pi,\omega\times\omega'}[\![\langle e, e'\rangle]\!]\ S\ \eta = \langle \mu_{\pi,\omega}[\![e]\!]\ S\ \eta, \mu_{\pi,\omega'}[\![e']\!]\ S\ \eta\rangle, \tag{Me}$$

$$\text{If } e \in E_{\pi,\omega\times\omega'} \text{ then} \quad \mu_{\pi\omega}[\![e.1]\!]\ S\ \eta = [\mu_{\pi,\omega\times\omega'}[\![e]\!]\ S\ \eta]_1 \quad \mu_{\pi\omega}[\![e.2]\!]\ S\ \eta = [\mu_{\pi,\omega\times\omega'}[\![e]\!]\ S\ \eta]_2, \tag{Mf}$$

$$\text{If } b \in E_{\pi,\text{Bool}} \text{ and } e, e' \in E_{\pi\omega} \text{ then} \quad \mu_{\pi\omega}[\![\text{if } b \text{ then } e \text{ else } e']\!]\ S\ \eta = \text{if } \mu_{\pi,\text{Bool}}[\![b]\!]\ S\ \eta = \text{true} \text{ then } \mu_{\pi,\omega}[\![e]\!]\ S\ \eta \text{ else } \mu_{\pi,\omega}[\![e']\!]\ S\ \eta. \tag{Mg}$$

$$\langle f_1, f_2\rangle \in r \to r' \text{ iff } (\forall \langle x_1, x_2\rangle \in r)\ \langle f_1\ x_1, f_2\ x_2\rangle \in r',$$

and $r \times r'$ for the relation in $\text{Rel}(s_1 \times s_1', s_2 \times s_2')$ such that

$$\langle\langle x_1, x_1'\rangle, \langle x_2, x_2'\rangle\rangle \in r \times r' \text{ iff } \langle x_1, x_2\rangle \in r \text{ and } \langle x_1', x_2'\rangle \in r'.$$

In other words, functions are related if they map related arguments into related results, and pairs are related if their corresponding components are related.

For set assignments S_1 and S_2, a member of

$$\prod_{\tau \in T} \text{Rel}(S_1\tau, S_2\tau)$$

is called a (binary) *relation assignment* between S_1 and S_2. Having defined $\to$ and $\times$ for relations we can extend relation assignments from T to Ω and Ω^* in essentially the same way as we extended set assignments. If R is a relation assignment between S_1 and S_2 then

$$R^\# \in \prod_{\omega \in \Omega} \text{Rel}(S_1^\#\omega, S_2^\#\omega)$$

is such that

$$\text{If } \kappa \in C \text{ then } R^\#\ \kappa = \text{I}(CS\ \kappa), \tag{R1}$$

$$\text{If } \tau \in T \text{ then } R^\#\ \tau = R\ \tau, \tag{R2}$$

$$\text{If } \omega, \omega' \in \Omega \text{ then } R^\#(\omega \to \omega') = R^\#\omega \to R^\#\omega', \tag{R3}$$

$$\text{If } \omega, \omega' \in \Omega \text{ then } R^\#(\omega \times \omega') = R^\#\omega \times R^\#\omega', \tag{R4}$$

and

$$R^{\#*} \in \prod_{\pi \in \Omega^*} \text{Rel}(S_1^{\#*}\pi, S_2^{\#*}\pi)$$

https://github.com/ionathanch/parapoly

Formalism = Adherence to Form

Are mechanizations actually *formal*?

Discussion question

If formal interoperability is impossible, how should we **share knowledge across different ITP communities** with overlapping mechanization goals?

Rosetta stones, manual comparison efforts a la POPLmark?

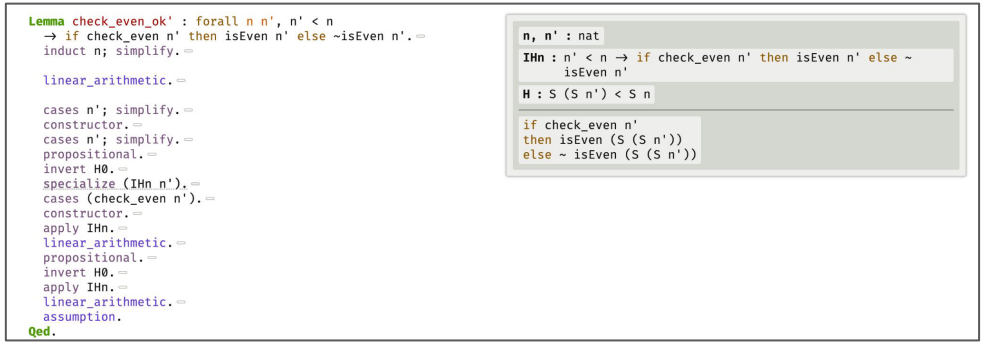
Collected desiderata

- **Peer review and curation** of proof artifacts
- **Archival storage**
- **Indexing** for search, citation, retrieval
- **Running stored proofs**
- **Comprehending** proofs
- **Adapting** old proofs to new contexts
  - Including portability across proof languages/tools

Interactivity

Replay/stepping through mechanized proofs

- Alectryon (Pit-Claudel)
- Various Lean tools



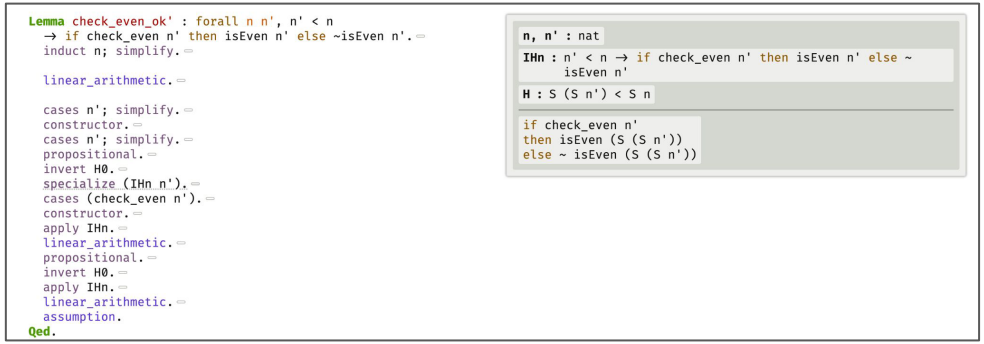
```
Lemma check_even_ok' : forall n n', n' < n
  → if check_even n' then isEven n' else ~isEven n'.
  induct n; simplify.

  linear_arithmetic.

  cases n'; simplify.
  constructor.
  cases n'; simplify.
  propositional.
  invert H0.
  specialize (IHn n').
  cases (check_even n').
  constructor.
  apply IHn.
  linear_arithmetic.
  propositional.
  invert H0.
  apply IHn.
  linear_arithmetic.
  assumption.
Qed.
```

```
n, n' : nat
IHn : n' < n → if check_even n' then isEven n' else ~
      isEven n'
H : S (S n') < S n
---
if check_even n'
then isEven (S (S n'))
else ~ isEven (S (S n'))
```

Interactivity

Replay/stepping through mechanized proofs

- Alectryon (Pit-Claudel)
- Various Lean tools



```
Lemma check_even_ok' : forall n n', n' < n
  → if check_even n' then isEven n' else ~isEven n'.
  induct n; simplify.

  linear_arithmetic.

  cases n'; simplify.
  constructor.
  cases n'; simplify.
  propositional.
  invert H0.
  specialize (IHn n').
  cases (check_even n').
  constructor.
  apply IHn.
  linear_arithmetic.
  propositional.
  invert H0.
  apply IHn.
  linear_arithmetic.
  assumption.
Qed.
```

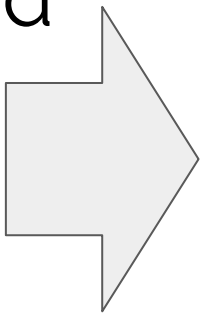
```
n, n' : nat
IHn : n' < n → if check_even n' then isEven n' else ~
      isEven n'
H : S (S n') < S n
---
if check_even n'
then isEven (S (S n'))
else ~ isEven (S (S n'))
```

Can we make these artifacts archival?

“New media”

(artists + technologists causing problems)

Time-based media:
film, animation, sound



Media players,
Digitization

“New media”

(artists + technologists causing problems)

10PRINT ?

Thermal printer orchestra ?

Archival:

- Hardware/media player preservation
- Emulators?
- Digital recordings?
- Specs to recreate?

Interactivity

- Games: playtraces, playable snippets for game citation

https://gisst.dev/

Coda: responsible stewardship

Who carries out stewardship?

- Proof assistant developers/maintainers
- Proof repository/library curators/peer reviewers
- Proof/artifact repository maintainers (e.g., universities, ACM)
- Proof authors

Contexts: academia, open source, private companies

When ITP versions update

Responsibilities of **proof repository maintainers**?

- Retain old versions?
- Update old proofs?
- Snapshot whole environment into a Docker image?

  (Pros/cons of all options for both convincing and explaining)

Responsibilities of **ITP maintainers**?

- Stable kernels?
- Backwards compatibility?
- Version managers?

Responsibilities of **proof author communities**?

- Documenting conventions, best practices?
- Accounting for goals and tradeoffs between desiderata (e.g. convincing and explaining)?

Discussion

In the paper

Coverage of current/past practices:

1. Artifact Evaluation
2. ITP-specific repositories/libraries/docs/search features
3. Interactive proof replay/inspection
4. Cross-ITP comparisons (e.g. POPLmark)
5. Proof maintenance and repair tools

Discussion questions

1. Have you ever tried to **read, run, or reuse** an “old” mechanized development for your own designs? How did that go?
2. Why do **you** mechanize your theory? Do you gravitate more towards convincing or explaining as default modes?
3. What “**knowledge artifacts**” should we attempt to store/preserve in order to support diverse social processes of proof?
4. How can we share knowledge **across different ITP communities** with overlapping mechanization goals?
5. Who bears what responsibility for **stewardship**?

These slides:

Corresponding authors

Chris Martens
contextadventure@gmail.com
https://www.khoury.northeastern.edu/~cmartens/
https://hci.social/@chrisamaphone

Emma Tosch
e.tosch@northeastern.edu
https://emmatosch.com/index.html

Paper pdf:

http://khoury.northeastern.edu/~cmartens/papers/plateau26-itfp.pdf