Verifying symbolic computation in the HolPy theorem prover

Bohua Zhan

Institute of Software, Chinese Academy of Sciences

Outline

1. Introduction to HolPy

   Joint with Zhenyan Ji, Wenfan Zhou, Chaozhu Xiang, Jie Hou, Wenhui Sun (ICFEM'19)

2. Verification of definite integrals

   Joint with Runqing Xu, Liming Li (CADE'21)

3. Recent extensions

   Joint with Yuheng Fan, Weiqiang Xiong

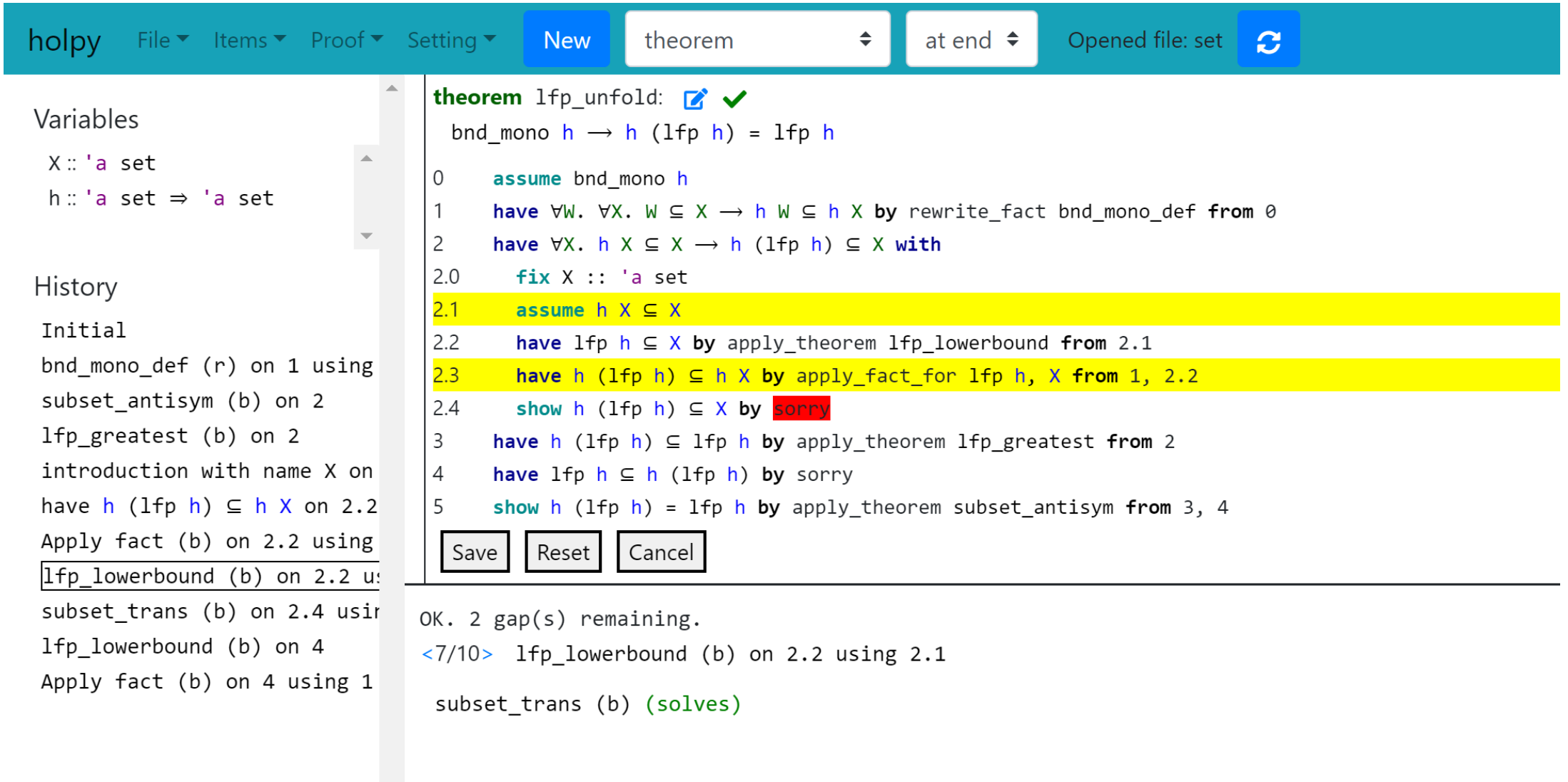
4. Discussion and future work

Demo!

Introduction to HolPy

- Development started in October 2018.
- Implemented in Python, with user interface in JavaScript.
- Based on higher-order logic (similar to Isabelle/HOL, HOL Light, etc).
- Python API for writing proof automation.
- Free hand to try out new designs for user interfaces.

User interface for HolPy (topic for another time …)



Python API for proof automation

- Allow users to extend proof automation by writing Python code (correctness still guaranteed by checking proof terms).
- Define Python classes for conversions, macros, tactics, etc.
- Support multiple styles for constructing proofs (top-down or bottom-up, functional or imperative).

Python API example

```
class norm_add_atom_1(Conv):
    """Normalize expression of the form (a_1 + ... + a_n) + a."""
    def get_proof_term(self, t: Term) -> ProofTerm:
        pt = refl(t)
        if t.arg1.is_zero():
            return pt.on_rhs(rewr_conv("nat_plus_def_1"))
        elif t.arg.is_zero():
            return pt.on_rhs(rewr_conv("add_0_right"))
        elif t.arg1.is_plus():
            if compare_atom(t.arg1.arg, t.arg) > 0:
                return pt.on_rhs(swap_add_r(), arg1_conv(norm_add_atom_1()))
            else:
                return pt
        else:
            if compare_atom(t.arg1, t.arg) > 0:
                return pt.on_rhs(rewr_conv("add_comm"))
            else:
                return pt
```

Case analysis on input term

Apply existing theorem

Apply existing conversions

Verification of definite integrals: motivation

- **Symbolic computation:** relatively limited language compared to full mathematics.
- Nevertheless, has wide application area:
  - Mathematical proofs
  - Calculations in science and engineering
- **Goal:** let users perform computations step-by-step in a familiar CAS-like setting, with results checkable by conversion to lower-level logical systems.

Potential applications in verifying …

If we let F be an exponential signal, the resulting response satisfies

$$M_t s^2\, p - mls^2\, \theta + cs\, p = F,$$
$$J_t s^2\, \theta - mls^2\, p + \gamma\, s\, \theta - mgl\, \theta = 0,$$

where all signals are exponential signals. The resulting transfer functions for the position of the cart and the orientation of the pendulum are given by solving for p and θ in terms of F to obtain

$$H_{\theta F} = \frac{mls}{(M_t J_t - m^2 l^2)s^4 + (\gamma M_t + cJ_t)s^2 + (c\gamma - M_t mgl)s - mglc},$$
$$H_{pF} = \frac{J_t s^2 + \gamma s - mgl}{(M_t J_t - m^2 l^2)s^4 + (\gamma M_t + cJ_t)s^3 + (c\gamma - M_t mgl)s^2 - mglcs},$$

where each of the coefficients is positive. The pole zero diagrams for these two transfer functions are shown in Figure 8.5 using the parameters from Example 6.7.

If we assume the damping is small and set $c = 0$ and $\gamma = 0$, we obtain

$$H_{\theta F} = \frac{ml}{(M_t J_t - m^2 l^2)s^2 - M_t mgl},$$
$$H_{pF} = \frac{J_t s^2 - mgl}{s^2\big((M_t J_t - m^2 l^2)s^2 - M_t mgl\big)}.$$

This gives nonzero poles and zeros at

$$p = \pm\sqrt{\frac{mgl\,M_t}{M_t J_t - m^2 l^2}} \approx \pm 2.68, \qquad z = \pm\sqrt{\frac{mgl}{J_t}} \approx \pm 2.09.$$

2.457

1. $$\int \frac{(A + B\cosh x)\,dx}{\cosh x\,(a + b\cosh x)} = \frac{1}{a}\left[A\,\operatorname{arctan\,sinh}\, x - (Ab - Ba)\int \frac{dx}{a + b\cosh x}\right]$$

(see **2.443** 3)

2.458

1. $$\int \frac{dx}{a + b\sinh^2 x}$$
$$= \frac{1}{\sqrt{a(b-a)}}\arctan\left(\sqrt{\frac{b}{a} - 1}\tanh x\right) \qquad \left[\frac{b}{a} > 1\right]$$
$$= \frac{1}{\sqrt{a(a-b)}}\operatorname{arctanh}\left(\sqrt{1 - \frac{b}{a}}\tanh x\right) \qquad \left[0 < \frac{b}{a} < 1 \text{ or } \frac{b}{a} < 0 \text{ and } \sinh^2 x < -\frac{a}{b}\right]$$
$$= \frac{1}{\sqrt{a(a-b)}}\operatorname{arccoth}\left(\sqrt{1 - \frac{b}{a}}\tanh x\right) \qquad \left[\frac{b}{a} < 0 \text{ and } \sinh^2 x > -\frac{a}{b}\right]$$

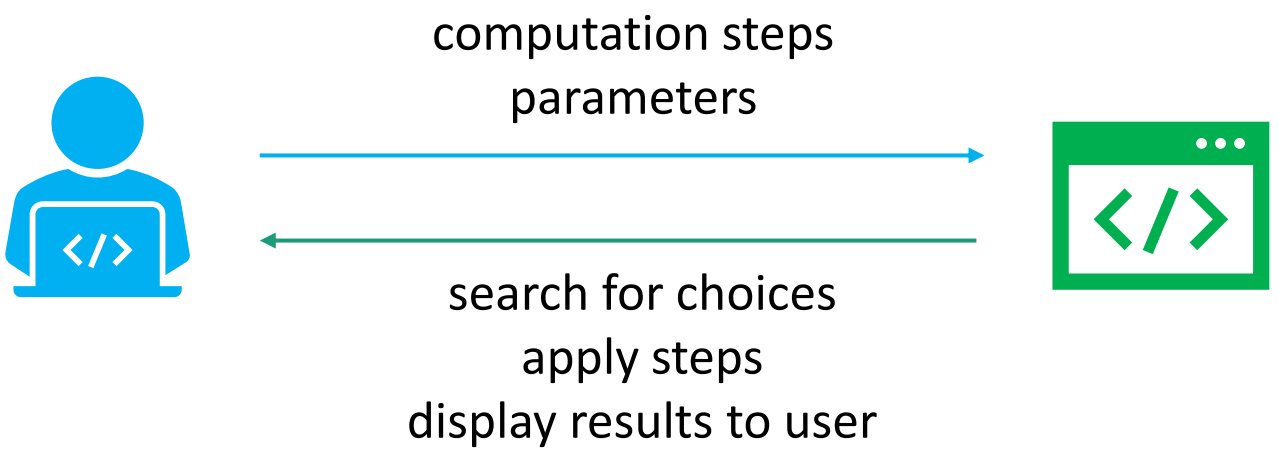
MZ 195

2. $$\int \frac{dx}{a + b\cosh^2 x}$$
$$= \frac{1}{\sqrt{-a(a+b)}}\arctan\left(\sqrt{-\left(1 + \frac{b}{a}\right)}\coth x\right) \qquad \left[\frac{b}{a} < -1\right]$$
$$= \frac{1}{\sqrt{a(a+b)}}\operatorname{arctanh}\left(\sqrt{1 + \frac{b}{a}}\coth x\right) \qquad \left[-1 < \frac{b}{a} < 0 \text{ and } \cosh^2 x > -\frac{a}{b}\right]$$
$$= \frac{1}{\sqrt{a(a+b)}}\operatorname{arccoth}\left(\sqrt{1 + \frac{b}{a}}\coth x\right) \qquad \left[\frac{b}{a} > 0 \text{ or } -1 < \frac{b}{a} < 0 \text{ and } \cosh^2 x < -\frac{a}{b}\right]$$

MZ 202

Basic idea

- Users specify steps of computation and possible parameters.
- Computer suggests choices, applies steps and show results to the user.



Example: integration by parts

$$\int_0^{\pi/2} \cos x e^{2x}\, dx$$

$$= (e^{2x} \sin x|_{x=0}^{\pi/2}) - (\int_0^{\pi/2} 2e^{2x} \sin x\, dx) \quad \text{(integrate by parts } u \to e^{2x}, v \to \sin x\text{)}$$

$$= -2(\int_0^{\pi/2} e^{2x} \sin x\, dx) + e^{\pi} \quad \text{(full simplify)}$$

Existing computation

Integrate by parts on: $e^{2x} \sin x$

Choose u and v such that $u \cdot \mathrm{d}v$ is the integrand.

$u =$

exp(2*x)

e^{2x}

$v =$

-cos(x)

$-\cos x$

User selects “integration by parts” from the menu

Query for expressions u and v.

Example: trigonometric identity

$$\int_{1/\sqrt{2}}^{1} \frac{\sqrt{1-x^2}}{x^2}\,dx$$

$$= \int_{\pi/4}^{\pi/2} \frac{\sqrt{1-\sin^2(u)}}{\sin^2(u)} \cos u\,du \quad \text{(inverse substitution for } \sin u\text{)}$$

$$= \int_{\pi/4}^{\pi/2} \frac{\cos u\sqrt{-\sin^2(u)+1}}{\sin^2(u)}\,du \quad \text{(full simplify)}$$

$$= \int_{\pi/4}^{\pi/2} \frac{\cos u\sqrt{-(1-\cos^2(u))+1}}{\sin^2(u)}\,du \quad \text{(rewrite } \sin^2(u) \text{ to } 1-\cos^2(u) \text{ using identity)}$$

$$= \int_{\pi/4}^{\pi/2} \frac{\cos^2(u)}{\sin^2(u)}\,du \quad \text{(full simplify)}$$

Select subexpression:

```
INT u:[pi / 4,pi / 2]. cos(u) ^ 2 / sin(u) ^ 2
```

$$\cos^2(u)$$
$$= -\sin^2(u) + 1$$
$$= \tfrac{1}{2}(\cos(2u) + 1)$$

Existing computation

↓

User selects “apply identity”

↓

User selects subexpression

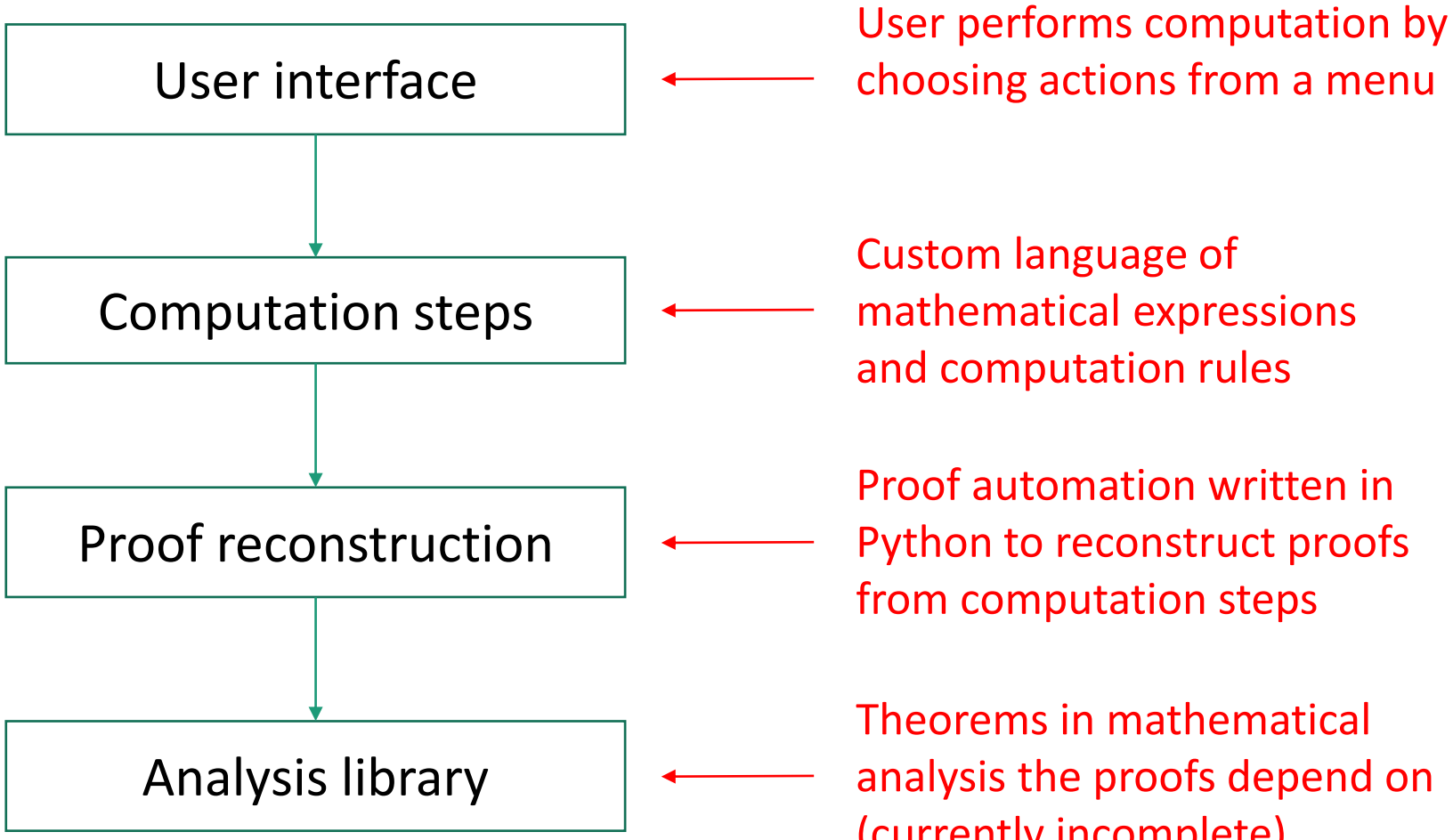
↓

Computer provides choices

Proof reconstruction

- From a sequence of computation steps, *automatically* reconstruct proofs in higher-order logic. This is possible since all necessary information is already available.
- **Main tasks:**
  - Proofs for simplification/rewriting of expressions.
  - Proofs for inequality checking.
  - Applying integration theorems, checking side conditions such as continuity, integrability, …
- Implemented in Python using the API for proof automation.
- Underlying mathematical library: still under construction.

Summary: overall architecture



Example #1

- An easy-to-make mistake:

$$\int_0^{\pi} \sqrt{1+\cos(2x)}\,dx = \int_0^{\pi} \sqrt{1+(2\cos^2 x - 1)}\,dx$$

$$= \int_0^{\pi} \sqrt{2}\cancel{\cos x\,dx}$$

$$= \cancel{\sqrt{2}(\sin\pi - \sin 0) = 0}$$

$\sqrt{\cos^2 x} \neq \cos x$

???

- Correct version:

$$\int_0^{\pi} \sqrt{1+\cos(2x)}\,dx = \int_0^{\pi} \sqrt{1+(2\cos^2 x - 1)}\,dx$$

$$= \int_0^{\pi} \sqrt{2}|\cos x|\,dx$$

$$= \int_0^{\pi/2} \sqrt{2}\cos x\,dx + \int_{\pi/2}^{\pi} -\sqrt{2}\cos x\,dx = 2\sqrt{2}$$

Example #2

- A problem that is difficult even for Mathematica:

$$\int_0^{\pi/100} \frac{\sin(20x)+\sin(19x)}{\cos(20x)+\cos(19x)}\,dx$$

$$= \int_0^{\pi/100} \frac{2\cos(\frac{x}{2})\sin(\frac{39x}{2})}{2\cos(\frac{x}{2})\cos(\frac{39x}{2})}\,dx$$

$$= \int_0^{\pi/100} \frac{\sin(\frac{39x}{2})}{\cos(\frac{39x}{2})}\,dx = \int_{\cos(39\pi/200)}^{1} \frac{2}{39u}\,du$$

$$= -\frac{2\log(\cos(\frac{39\pi}{200}))}{39}$$

Limitations

The above work is limited in the following ways:

- *Extensibility:* all functions and identities satisfied by them are hard-coded.
- *Reusability:* calculations are independent from each other, no reuse of results.
- *Range of application:* definite integrals on finite intervals only.

Extensions

The first major extension we made is to introduce a system for managing definitions, identities, and proofs:

- Define new functions and state identities.
- Manage conditions of identities.
- Proof by induction.

More and more like regular ITP systems…

New rules

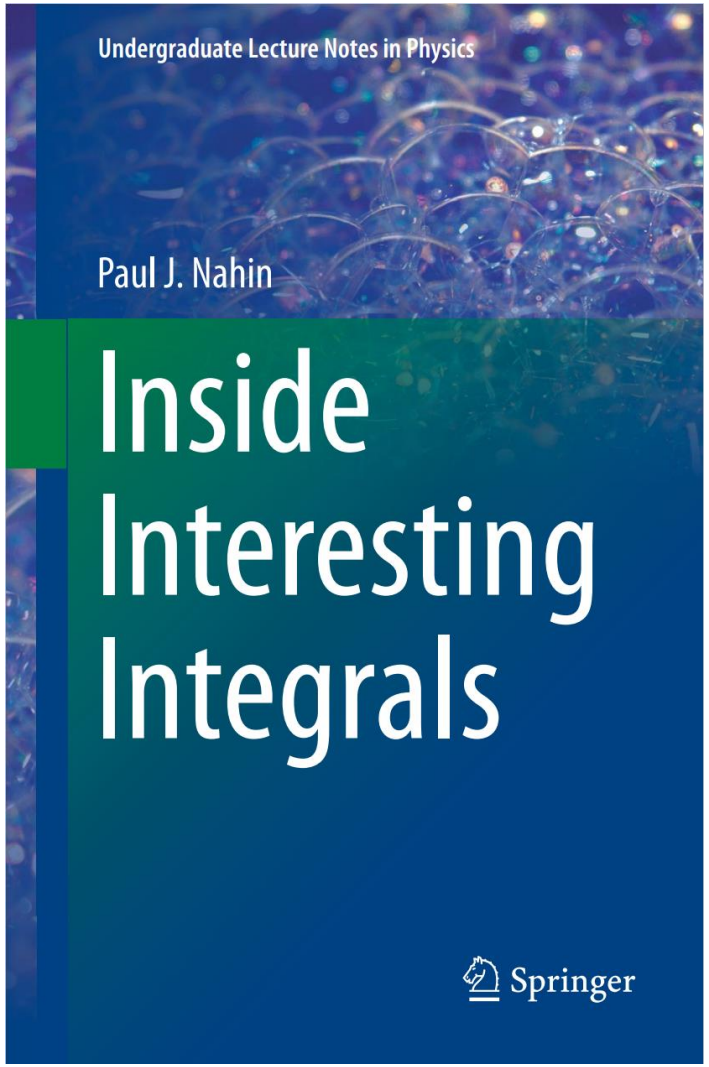
We also added rules for dealing with:

- Improper integrals and limits.
- Indefinite integrals, and working with the "$+ C$".
- Series expansion and evaluations.
- Swapping integral and derivative, integral and sums.

Currently: 27 rules to choose from in the menu.

Examples

- We tried our tool on examples from the book “Inside Interesting Integrals”.
- Over 50 integrals from the book are performed, covering topics like:
  - Partial fraction decomposition.
  - Differentiating under integral sign.
  - Use of power series in integration.
  - Gamma and Beta functions.
- The following are *not* included:
  - Complex numbers and contour integrals.



Example #3

Using power series of log function (from Section 5.2):

$$\int_0^1 \frac{\log(1+x)}{x}\,dx = \int_0^1 \frac{\sum_{n=0}^{\infty} \frac{(-1)^n x^{n+1}}{n+1}}{x}\,dx \qquad \text{(expand power series)}$$

$$= \int_0^1 \sum_{n=0}^{\infty} \frac{(-1)^n x^n}{n+1}\,dx$$

$$= \sum_{n=0}^{\infty} \int_0^1 \frac{(-1)^n x^n}{n+1}\,dx \qquad \text{(exchange integral and sum)}$$

$$= \sum_{n=0}^{\infty} \frac{(-1)^n}{(n+1)^2} \qquad \text{(evaluate integral term by term)}$$

$$= \frac{\pi^2}{12} \qquad \text{(evaluate series)}$$

Example #4

Exchanging derivative and integral (from Section 3.4):

To evaluate

$$I(a) = \int_0^1 \frac{x^a - 1}{\log x}\, dx$$

First differentiate with respect to a: (exchange deriv. and integral)

$$\frac{d}{da} I(a) = \frac{d}{da} \int_0^1 \frac{x^a - 1}{\log x}\, dx = \int_0^1 \frac{d}{da} \frac{x^a - 1}{\log x}\, dx = \int_0^1 x^a\, dx = \frac{1}{a+1}$$

Integrating both sides, we get (indefinite integral with $+C$)

$$I(a) = \int \frac{1}{a+1} = \log(a+1) + C$$

(solving for value of C)

By taking $a = 0$ we get $I(a) = 0$, so $C = 0$. This means $I(a) = \log(a+1)$.

Example #5:

Reasoning about the Γ-function (Section 4.1).

First, we show a standard identity:

$$\begin{aligned}\Gamma(n+1) &= \int_0^\infty e^{-x}x^n\,dx \\ &= -e^{-x}x^n|_0^\infty + \int_0^\infty ne^{-x}x^{n-1}\,dx \qquad \text{(integration by parts with } \infty\text{)} \\ &= n\int_0^\infty e^{-x}x^{n-1}\,dx = n\Gamma(n)\end{aligned}$$

Hence we obtain the following by induction:

$$\Gamma(n+1) = n! \qquad \text{(proof by induction)}$$

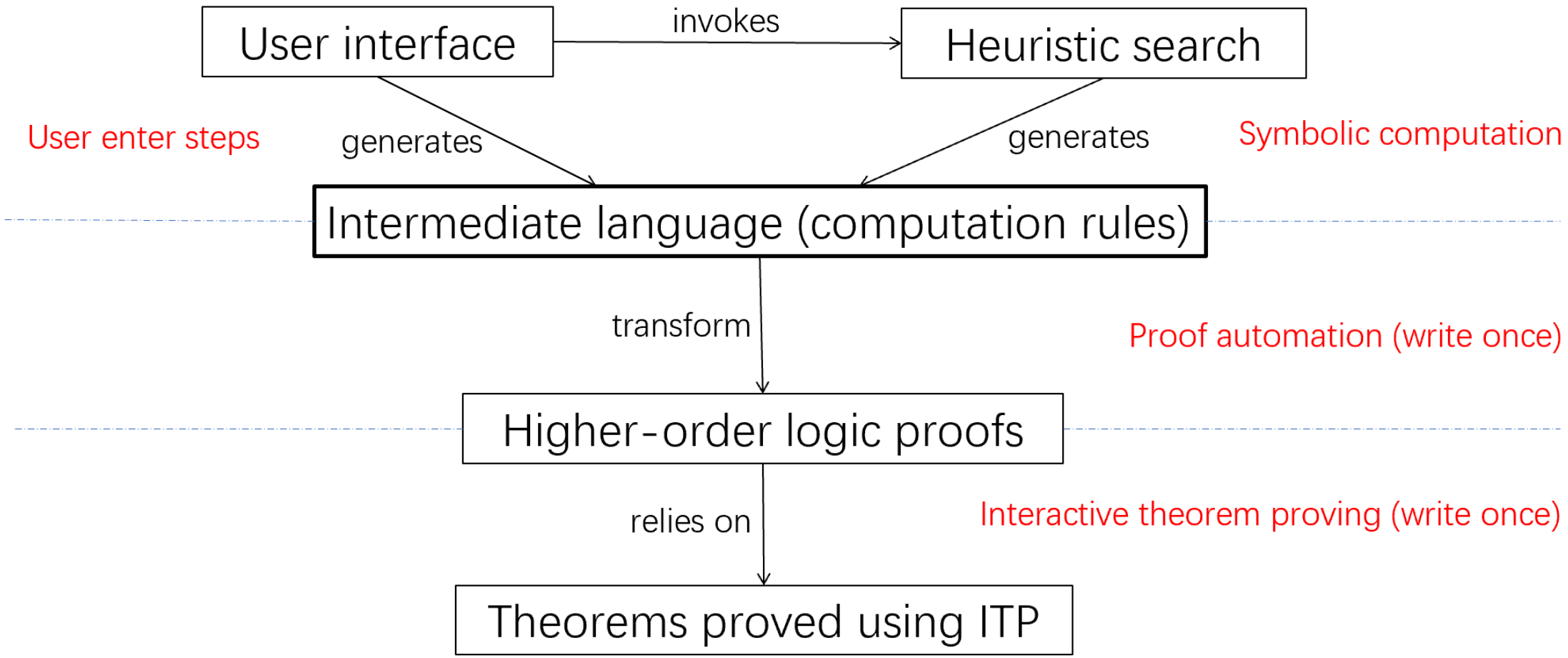
As an application of Γ function, we compute

$$\int_0^\infty e^{-x^3}\,dx = \frac{1}{3}\int_0^\infty e^{-y}y^{-2/3}\,dy = \frac{1}{3}\Gamma\left(\frac{1}{3}\right) = \Gamma\left(\frac{4}{3}\right) \qquad \text{(substitution, fold definition)}$$

Summary

- “Interactive Theorem Prover” designed specially for symbolic computation.
- User interface displays formulas in LaTeX form.
- CAS-like automation to help users with simple tasks.
- Generate higher-order logic proofs that can be checked independently (currently implemented only for simple cases).

Summary: division of labor



Summary: relation with CAS

- Permits more manual control of the computation.
- Many components of computer algebra systems are used here, currently only with simple implementations:
  - Simplification and normalization.
  - Pattern matching.
  - Checking conditions.
  - Taking limits.
  - Solving equations.
- Need proof-generating versions of these procedures!

Future work

- Extensions:
  - Linear algebra
  - Complex analysis
  - Probability
- Application to verifying symbolic computation that appear in mathematical proofs, engineering, machine learning algorithms …
- Applications in education?
- We can try to build tools that are immediately useful, without achieving full rigor in the ITP sense.

Future work

- To achieve full rigor, we need:
  - Proof reconstruction for all CAS-procedures.
  - Library of mathematical theorems covering the foundational rules.
- Integration within an interactive theorem prover:
  - Switch between general-purpose and symbolic computation modes. (similar to math texts).
- Available as open source, and we welcome contributions!

https://github.com/bzhan/holpy · https://gitee.com/bhzhan/holpy