Idea: the Amsterdam Theorem Exchange

dr. Hans-Dieter A. Hiep (hdh@drheap.nl)

Declarative.Amsterdam
Friday, November 7th, 2025

Outline

1. Recall: mathematical logic
2. What are solvable problems?
3. What is a theorem exchange?
4. Valuable problems: specialization and arbitrage
5. Dominating all cryptocurrencies

1/5. Mathematical logic

Syntax

- vocabulary/signature Σ
  constants $1, 2, 3, \ldots$
  operators $+, \times$
  relations $\leq$

- variables
  terms $x + y$
  statements $\forall x \exists y (x + y = 0)$

- finitary proofs
  syntactic consequence $\Gamma \vdash \varphi$

Semantics

- model/structure $\mathcal{M}$
  domain/universe
  interpretation

- valuation $\rho(x)$
  denotation $[\![t]\!]_\rho$
  satisfaction $\mathcal{M} \models \varphi$

- infinitary intuitions
  semantic consequence $\Gamma \models \varphi$

Soundness and completeness (Gödel 1930)

1/5. Mathematical logic

Syntax

- vocabulary/signature Σ
  constants $1, 2, 3, \ldots$
  operators $+, \times$
  relations $\leq$

- variables
  terms $x + y$
  statements $\forall x \exists y (x + y = 0)$

- finitary proofs
  syntactic consequence $\Gamma \vdash \varphi$

Semantics

- model/structure $\mathcal{M}$
  domain/universe
  interpretation

- valuation $\rho(x)$
  denotation $[\![t]\!]_\rho$
  satisfaction $\mathcal{M} \models \varphi$

- infinitary intuitions
  semantic consequence $\Gamma \models \varphi$

Soundness and completeness (Gödel 1930)

1/5. Mathematical logic

Syntax

- vocabulary/signature Σ
  constants $1, 2, 3, \ldots$
  operators $+, \times$
  relations $\leq$

- variables
  terms $x + y$
  statements $\forall x \exists y (x + y = 0)$

- finitary proofs
  syntactic consequence $\Gamma \vdash \varphi$

Semantics

- model/structure $\mathcal{M}$
  domain/universe
  interpretation

- valuation $\rho(x)$
  denotation $[\![t]\!]_\rho$
  satisfaction $\mathcal{M} \models \varphi$

- infinitary intuitions
  semantic consequence $\Gamma \models \varphi$

Soundness and completeness (Gödel 1930)

2/5. What are solvable problems?

- two formal systems: (A) for syntax, (B) for semantics for which soundness and completeness holds

- a problem φ in theory Γ is **solvable** if either:
  - (A) shows $\Gamma \vdash \varphi$, (construction of proof)
  - (B) shows $\Gamma \not\models \varphi$ (construction of counter-example)

- in a complete theory Γ, every problem φ is **decidable**: $\Gamma \vdash \varphi$ or $\Gamma \vdash \neg\varphi$

Incompleteness (Gödel 1931)
In a consistent, sufficiently expressive theory Γ (i.e. can do elementary arithmetic), $\mathrm{Con}(\Gamma)$ can not be decided by Γ.

But a counter-example $\Gamma \not\models \mathrm{Con}(\Gamma)$ can sometimes be constructed!

2/5. What are solvable problems?

- two formal systems: (A) for syntax, (B) for semantics
  for which soundness and completeness holds

- a problem φ in theory Γ is **solvable** if either:
  - (A) shows $\Gamma \vdash \varphi$, (construction of proof)
  - (B) shows $\Gamma \not\models \varphi$ (construction of counter-example)

- in a complete theory Γ, every problem φ is **decidable**:
  $\Gamma \vdash \varphi$ or $\Gamma \vdash \neg\varphi$

Incompleteness (Gödel 1931)
In a consistent, sufficiently expressive theory Γ (i.e. can do elementary arithmetic), $\mathrm{Con}(\Gamma)$ can not be decided by Γ.

But a counter-example $\Gamma \not\models \mathrm{Con}(\Gamma)$ can sometimes be constructed!

2/5. What are solvable problems?

- two formal systems: (A) for syntax, (B) for semantics
  for which soundness and completeness holds

- a problem φ in theory Γ is **solvable** if either:
  - (A) shows $\Gamma \vdash \varphi$, (construction of proof)
  - (B) shows $\Gamma \not\models \varphi$ (construction of counter-example)

- in a complete theory Γ, every problem φ is **decidable**:
  $\Gamma \vdash \varphi$ or $\Gamma \vdash \neg\varphi$

Incompleteness (Gödel 1931)
In a consistent, sufficiently expressive theory Γ (i.e. can do elementary arithmetic), $\mathrm{Con}(\Gamma)$ can not be decided by Γ.

But a counter-example $\Gamma \not\models \mathrm{Con}(\Gamma)$ can sometimes be constructed!

2/5. What are solvable problems?

- two formal systems: (A) for syntax, (B) for semantics for which soundness and completeness holds
- a problem φ in theory Γ is **solvable** if either:
  - (A) shows $\Gamma \vdash \varphi$, (construction of proof)
  - (B) shows $\Gamma \not\models \varphi$ (construction of counter-example)
- in a complete theory Γ, every problem φ is **decidable**: $\Gamma \vdash \varphi$ or $\Gamma \vdash \neg\varphi$

Incompleteness (Gödel 1931)
In a consistent, sufficiently expressive theory Γ (i.e. can do elementary arithmetic), $\mathrm{Con}(\Gamma)$ can not be decided by Γ.

But a counter-example $\Gamma \not\models \mathrm{Con}(\Gamma)$ can sometimes be constructed!

2/5. What are solvable problems?

- two formal systems: (A) for syntax, (B) for semantics
  for which soundness and completeness holds

- a problem φ in theory Γ is **solvable** if either:
  - (A) shows $\Gamma \vdash \varphi$, (construction of proof)
  - (B) shows $\Gamma \not\models \varphi$ (construction of counter-example)

- in a complete theory Γ, every problem φ is **decidable**:
  $\Gamma \vdash \varphi$ or $\Gamma \vdash \neg\varphi$

Incompleteness (Gödel 1931)
In a consistent, sufficiently expressive theory Γ (i.e. can do elementary arithmetic), $\mathrm{Con}(\Gamma)$ can not be decided by Γ.

But a counter-example $\Gamma \not\models \mathrm{Con}(\Gamma)$ can sometimes be constructed!

3/5. What is a theorem exchange?

A problem is **solved** if proof or counter-example is constructed.

A problem is **outstanding** if not yet solved.

Imagine a system that:

- collects outstanding problems
- assigns monetary values ($, £, €) to outstanding problems
- at any time, solution and money can be exchanged
- concurrent solvers are motivated by earning money
- problems and their solutions may need *perfect secrecy*
- public money $\rightarrow$ public solutions

IMPORTANT: Separation of power.

Holy trinity: Specification — Implementation — Verification

3/5. What is a theorem exchange?

A problem is **solved** if proof or counter-example is constructed.

A problem is **outstanding** if not yet solved.

Imagine a system that:

- collects outstanding problems
- assigns monetary values ($,£,€) to outstanding problems
- at any time, solution and money can be exchanged
- concurrent solvers are motivated by earning money
- problems and their solutions may need *perfect secrecy*
- public money $\rightarrow$ public solutions

IMPORTANT: Separation of power.

Holy trinity: Specification — Implementation — Verification

3/5. What is a theorem exchange?

A problem is **solved** if proof or counter-example is constructed.

A problem is **outstanding** if not yet solved.

Imagine a system that:

- collects outstanding problems
- assigns monetary values ($,£,€) to outstanding problems
- at any time, solution and money can be exchanged
- concurrent solvers are motivated by earning money
- problems and their solutions may need *perfect secrecy*
- public money $\rightarrow$ public solutions

IMPORTANT: Separation of power.

Holy trinity: Specification — Implementation — Verification

3/5. What is a theorem exchange?

A problem is **solved** if proof or counter-example is constructed.

A problem is **outstanding** if not yet solved.

Imagine a system that:

- collects outstanding problems
- assigns monetary values (\$,£,€) to outstanding problems
- at any time, solution and money can be exchanged
- concurrent solvers are motivated by earning money
- problems and their solutions may need *perfect secrecy*
- public money → public solutions

IMPORTANT: Separation of power.

Holy trinity: Specification — Implementation — Verification

3/5. What is a theorem exchange?

A problem is **solved** if proof or counter-example is constructed.

A problem is **outstanding** if not yet solved.

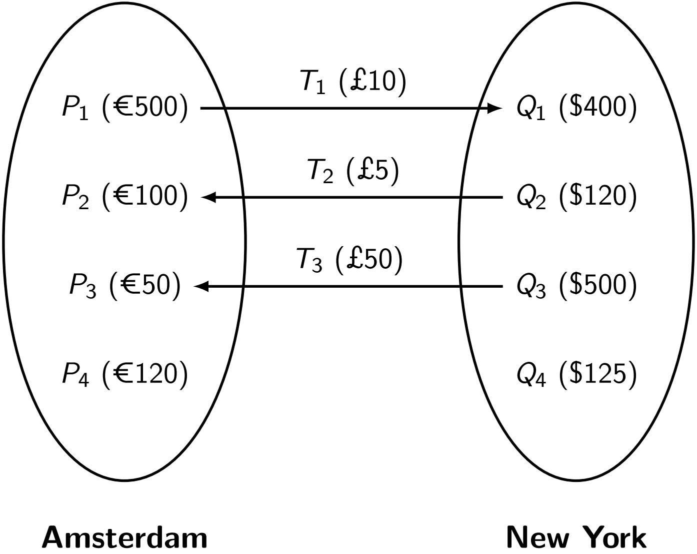
Imagine a system that:

- collects outstanding problems
- assigns monetary values (\$,£,€) to outstanding problems
- at any time, solution and money can be exchanged
- concurrent solvers are motivated by earning money
- problems and their solutions may need *perfect secrecy*
- public money $\rightarrow$ public solutions

IMPORTANT: Separation of power.

Holy trinity: Specification — Implementation — Verification

4/5. Valuable problems: specialization and arbitrage



5/5. Dominating all cryptocurrencies

Example problems:

- Reversing cryptographic functions on specific output
- Primality testing (NP and co-NP) and factoring
- Optimization and logistic problems
- Verifying semiconductor designs, gateware, firmware, ...
- Finding zero-day exploits
- etc.

Introducing the **Amsterdam Theorem Exchange**:

- a theorem exchange built on top of the SCION internet
- setting up not-for-profit, infrastructure at CWI
- transaction overhead (1%) funds fundamental research

5/5. Dominating all cryptocurrencies

Example problems:

- Reversing cryptographic functions on specific output
- Primality testing (NP and co-NP) and factoring
- Optimization and logistic problems
- Verifying semiconductor designs, gateware, firmware, ...
- Finding zero-day exploits
- etc.

Introducing the **Amsterdam Theorem Exchange**:

- a theorem exchange built on top of the SCION internet
- setting up not-for-profit, infrastructure at CWI
- transaction overhead (1%) funds fundamental research