DXXXXXXXXXXXXXX
XIXXXXXXXXXXXXX
XXAXXXXXXXXXXXX
XXXGXXXXXXXXXXX
XXXXOXXXXXXXXXX
XXXXXNXXXXXXXXX
XXXXXXAXXXXXXXX
XXXXXXXLXXXXXXX
XXXXXXXXIXXXXXX
XXXXXXXXXSXXXXX
XXXXXXXXXXAXXXX
XXXXXXXXXXXTXXX
XXXXXXXXXXXXIXX
XXXXXXXXXXXXXOX
XXXXXXXXXXXXXXN

Diagonalisation

The Road to Infinities, Truth and Gödel

Rasmus Bakken — Mahin Hossain

University of Oxford

Michaelmas 2025

Lecture 2 Summary

- Introduce the formal language $\mathcal{L}$.
- Show how $\mathcal{L}$ can express statements in arithmetic.
- Motivate Gödel-numbering as a precise way to achieve self-reference.
- We outline a Gödel-encoding scheme for symbols, sentences, and proofs, and illustrate it through worked examples.

Quote

"There is a difference between a thing and talking about a thing."

— Kurt Gödel

Roadmap

The Symbols of $\mathcal{L}$

- Alphabet: $0, S, +, *, exp, (,), \neg, \rightarrow, \forall, =, \#, v_i$ (for each $i \in \mathbb{N}$).

The Symbols of $\mathcal{L}$

- Alphabet: $0, S, +, *, exp, (,), \neg, \rightarrow, \forall, =, \#, v_i$ (for each $i \in \mathbb{N}$).
- An *expression* is any finite string of symbols (not necessarily sensible).

The Symbols of $\mathcal{L}$

- Alphabet: $0, S, +, *, exp, (,), \neg, \rightarrow, \forall, =, \#, v_i$ (for each $i \in \mathbb{N}$).
- An *expression* is any finite string of symbols (not necessarily sensible).
- We need rules to pick out well-formed expressions (terms, formulae, sentences).

Some $\mathcal{L}$-expressions

(i) $S(0) + S(0) = S(S(0))$

(ii) $\forall \neg v_8)$

(iii) $v_2 v_3 + + =$

Terms (syntax that denotes numbers)

- Intuition: a term is supposed to pick out a number.

Terms (syntax that denotes numbers)

- Intuition: a term is supposed to pick out a number.
- Basic terms: 0 and variables v_i.

Terms (syntax that denotes numbers)

- Intuition: a term is supposed to pick out a number.
- Basic terms: 0 and variables v_i.
- If t is a term, then $S(t)$ is a term.

Terms (syntax that denotes numbers)

- Intuition: a term is supposed to pick out a number.
- Basic terms: 0 and variables v_i.
- If t is a term, then $S(t)$ is a term.
- If x, y are terms, then $x + y$, $x * y$, and $exp(x, y)$ are terms.

Terms (syntax that denotes numbers)

- Intuition: a term is supposed to pick out a number.
- Basic terms: 0 and variables v_i.
- If t is a term, then $S(t)$ is a term.
- If x, y are terms, then $x + y$, $x * y$, and $exp(x, y)$ are terms.
- Terms are built recursively from the above rules.

Numerals

- *Numerals* are terms formed by iterating S on 0: $0, S(0), S(S(0)), \ldots$

Numerals

- *Numerals* are terms formed by iterating S on 0: $0, S(0), S(S(0)), \ldots$
- Shorthand: $\overline{n}$ denotes the numeral with n occurrences of S.

  Example: $\overline{0} = 0$, $\overline{1} = S(0)$, $\overline{3} = S(S(S(0)))$.

Formulae and sentences

- Intuition: a formula expresses a statement about numbers.

Formulae and sentences

- Intuition: a formula expresses a statement about numbers.
- Atomic formula: if σ, τ are terms then $\sigma = \tau$ is a formula.

Formulae and sentences

- Intuition: a formula expresses a statement about numbers.
- Atomic formula: if σ, τ are terms then $\sigma = \tau$ is a formula.
- If φ, ψ are formulae and $i \in \mathbb{N}$ then:
  - $\neg\varphi$ is a formula,
  - $(\varphi \to \psi)$ is a formula,
  - $\forall v_i\, \varphi$ is a formula.

Formulae and sentences

- Intuition: a formula expresses a statement about numbers.
- Atomic formula: if σ, τ are terms then $\sigma = \tau$ is a formula.
- If φ, ψ are formulae and $i \in \mathbb{N}$ then:
  - $\neg\varphi$ is a formula,
  - $(\varphi \to \psi)$ is a formula,
  - $\forall v_i\, \varphi$ is a formula.
- What about $\varphi \wedge \psi$, $\varphi \vee \psi$ and $\exists v_i \varphi$?

Examples of formulae

- $S(v_5) \neq 0$.
- $S(v_2) = S(v_3) \rightarrow v_2 = v_3$.
- $(\overline{2} + \overline{2} = \overline{4})$.
- $\forall v_1 (\forall v_2 \forall v_3 ((v_1 = v_2 * v_3) \rightarrow (v_3 = \overline{1} \vee v_3 = v_1)) \rightarrow (\exists v_4 \forall v_5 \forall v_6 ((v_4 = v_5 * v_6) \rightarrow (v_6 = \overline{1} \vee v_6 = v_4)) \wedge \exists v_7 (v_7 \neq 0 \wedge v_1 + v_7 = v_4)))$

Roadmap

How can $\mathcal{L}$ talk about itself?

- $\mathcal{L}$ can only talk about numbers. To make it talk about itself, we have to code expressions of $\mathcal{L}$ as numbers.

How can $\mathcal{L}$ talk about itself?

- $\mathcal{L}$ can only talk about numbers. To make it talk about itself, we have to code expressions of $\mathcal{L}$ as numbers.
- We want an injective function $\ulcorner \cdot \urcorner$ that maps φ to it's *Gödel code*.

How can $\mathcal{L}$ talk about itself?

- $\mathcal{L}$ can only talk about numbers. To make it talk about itself, we have to code expressions of $\mathcal{L}$ as numbers.
- We want an injective function $\ulcorner \cdot \urcorner$ that maps φ to it's *Gödel code*.
- Two step plan:
  - For each symbol x of $\mathcal{L}$, define $\ulcorner x \urcorner$ (easy).
  - Find out how to encode a sequence of numbers as one number (difficult).

Step 1: Assign codes to symbols

0	S	$+$	$*$	exp	$($	$)$	$\neg$	$\rightarrow$	$\forall$	$=$	$\#$
1	3	5	7	9	11	13	15	17	19	21	23

Step 1: Assign codes to symbols

0	S	$+$	$*$	exp	$($	$)$	$\neg$	$\rightarrow$	$\forall$	$=$	$\#$
1	3	5	7	9	11	13	15	17	19	21	23

Variables: $\ulcorner v_i \urcorner = 2(i+1)$ (so $\ulcorner v_0 \urcorner = 2$, $\ulcorner v_1 \urcorner = 4, \ldots$)

Taking Stock

- Let's consider $\ulcorner 0 = 0 \urcorner$.

Taking Stock

- Let's consider $\ulcorner 0 = 0 \urcorner$.
- We know that:
  - $\ulcorner 0 \urcorner = 1$ and
  - $\ulcorner = \urcorner = 21$.

Taking Stock

- Let's consider $\ulcorner 0 = 0 \urcorner$.
- We know that:
  - $\ulcorner 0 \urcorner = 1$ and
  - $\ulcorner = \urcorner = 21$.
- Hence, we can think of $\ulcorner 0 = 0 \urcorner$ as $\langle \ulcorner 0 \urcorner, \ulcorner = \urcorner, \ulcorner 0 \urcorner \rangle = \langle 1, 21, 1 \rangle$

Taking Stock

- Let's consider $\ulcorner 0 = 0 \urcorner$.
- We know that:
    - $\ulcorner 0 \urcorner = 1$ and
    - $\ulcorner = \urcorner = 21$.
- Hence, we can think of $\ulcorner 0 = 0 \urcorner$ as $\langle \ulcorner 0 \urcorner, \ulcorner = \urcorner, \ulcorner 0 \urcorner \rangle = \langle 1, 21, 1 \rangle$
- Question: How to encode $\langle 1, 21, 1 \rangle$ as one number?

From tuples to numbers

- Loosely speaking, we want an injective function $\mathcal{F}; \langle x_1, \ldots, x_k \rangle \mapsto n$.

From tuples to numbers

- Loosely speaking, we want an injective function $\mathcal{F}; \langle x_1, \ldots, x_k \rangle \mapsto n$.
- Injectivity ensures that if $\langle x_1, \ldots, x_k \rangle \neq \langle y_1, \ldots, y_l \rangle$ then $\mathcal{F}(\langle x_1, \ldots, x_k \rangle) \neq \mathcal{F}(\langle y_1, \ldots, y_l \rangle)$.

From tuples to numbers

- Loosely speaking, we want an injective function $\mathcal{F}; \langle x_1, \ldots, x_k \rangle \mapsto n$.
- Injectivity ensures that if $\langle x_1, \ldots, x_k \rangle \neq \langle y_1, \ldots, y_l \rangle$ then $\mathcal{F}(\langle x_1, \ldots, x_k \rangle) \neq \mathcal{F}(\langle y_1, \ldots, y_l \rangle)$.
- This means that if φ and ψ are two distinct formulae, then they get different tuples and hence different Gödel codes.

Some Number Theory

- A *prime number* n is a natural number larger than 1 such that it is only divisible by 1 and itself.

Some Number Theory

- A *prime number* n is a natural number larger than 1 such that it is only divisible by 1 and itself.
- A *composite* number is a natural number larger than 1 that is not a prime number.

Some Number Theory

- A *prime number* n is a natural number larger than 1 such that it is only divisible by 1 and itself.
- A *composite* number is a natural number larger than 1 that is not a prime number.
- 4 is a composite number that is built up of prime numbers 2 and 2.

Some Number Theory

- A *prime number* n is a natural number larger than 1 such that it is only divisible by 1 and itself.
- A *composite* number is a natural number larger than 1 that is not a prime number.
- 4 is a composite number that is built up of prime numbers 2 and 2.
- $18 = 9 * 2 = 3^2 * 2$
- A *prime composition* of a number n is a product of prime numbers raised to positive exponents in decreasing order evaluates to n.

Some Number Theory

- A *prime number* n is a natural number larger than 1 such that it is only divisible by 1 and itself.
- A *composite* number is a natural number larger than 1 that is not a prime number.
- 4 is a composite number that is built up of prime numbers 2 and 2.
- $18 = 9 * 2 = 3^2 * 2$
- A *prime composition* of a number n is a product of prime numbers raised to positive exponents in decreasing order evaluates to n.
- The prime composition of 18 is $3^2 * 2$, and it is unique.

The Fundamental Theorem of Arithmetic

Theorem (The Fundamental Theorem of Arithmetic)

Every natural number greater than 1 *has a unique prime composition.*

Proof.

Trust me. □

Encoding tuples by prime powers

- Let π_i be the i-th prime number. I.e., $\pi_1 = 2,\ \pi_2 = 3,\ \pi_3 = 5$ and so on.

Encoding tuples by prime powers

- Let π_i be the i-th prime number. I.e., $\pi_1 = 2,\ \pi_2 = 3,\ \pi_3 = 5$ and so on.
- We then say that $\mathcal{F}(\langle n_1, \ldots, n_k \rangle) = \pi_1^{n_1} \pi_2^{n_2} \cdots \pi_k^{n_k}$.

Encoding tuples by prime powers

- Let π_i be the i-th prime number. I.e., $\pi_1 = 2,\ \pi_2 = 3,\ \pi_3 = 5$ and so on.
- We then say that $\mathcal{F}(\langle n_1, \ldots, n_k \rangle) = \pi_1^{n_1} \pi_2^{n_2} \cdots \pi_k^{n_k}$.
- By the Fundamental Theorem of Arithmetic this encoding is *injective*.

The Gödel Code

Definition

Let φ be a $\mathcal{L}$-formula with symbols $\psi_1, \ldots, \psi_n$. Then we define *the Gödel Code of* φ, denoted $\ulcorner\varphi\urcorner$, as $\mathcal{F}(\langle\ulcorner\psi_1\urcorner, \ldots, \ulcorner\psi_n\urcorner\rangle)$.

Short example

- Consider the formula $0 = 0$.

Short example

- Consider the formula $0 = 0$.
- Tuple: $\langle 1, 21, 1 \rangle$.

Short example

- Consider the formula $0 = 0$.
- Tuple: $\langle 1, 21, 1 \rangle$.
- Encoding the tuple:

$$2^1 \cdot 3^{21} \cdot 5^1 \;=\; 104603532030.$$

Short example

- Consider the formula $0 = 0$.
- Tuple: $\langle 1, 21, 1 \rangle$.
- Encoding the tuple:
$$2^1 \cdot 3^{21} \cdot 5^1 \;=\; 104603532030.$$
- Thus $\ulcorner 0 = 0 \urcorner = \mathcal{F}(\langle \ulcorner 0 \urcorner, \ulcorner = \urcorner, \ulcorner 0 \urcorner \rangle) = \mathcal{F}(\langle 1, 21, 1 \rangle) = 2^1 \cdot 3^{21} \cdot 5^1 \;=\; 104603532030.$

Longer example

- Sentence: $\forall v_5(\neg S(v_5) = 0)$.

Longer example

- Sentence: $\forall v_5(\neg S(v_5) = 0)$.
- Tuple:
  - $\langle \ulcorner\forall\urcorner, \ulcorner v_5\urcorner, \ulcorner(\urcorner, \ulcorner\neg\urcorner, \ulcorner S\urcorner, \ulcorner(\urcorner, \ulcorner v_5\urcorner, \ulcorner)\urcorner, \ulcorner=\urcorner, \ulcorner 0\urcorner, \ulcorner)\urcorner \rangle$

Longer example

- Sentence: $\forall v_5(\neg S(v_5) = 0)$.
- Tuple:
  - $\langle \ulcorner\forall\urcorner, \ulcorner v_5\urcorner, \ulcorner(\urcorner, \ulcorner\neg\urcorner, \ulcorner S\urcorner, \ulcorner(\urcorner, \ulcorner v_5\urcorner, \ulcorner)\urcorner, \ulcorner=\urcorner, \ulcorner 0\urcorner, \ulcorner)\urcorner\rangle$
  - $\langle 19, 12, 11, 15, 3, 11, 12, 13, 21, 1, 13\rangle$

Longer example

- Sentence: $\forall v_5(\neg S(v_5) = 0)$.
- Tuple:
  - $\langle \ulcorner\forall\urcorner, \ulcorner v_5\urcorner, \ulcorner(\urcorner, \ulcorner\neg\urcorner, \ulcorner S\urcorner, \ulcorner(\urcorner, \ulcorner v_5\urcorner, \ulcorner)\urcorner, \ulcorner=\urcorner, \ulcorner 0\urcorner, \ulcorner)\urcorner \rangle$
  - $\langle 19, 12, 11, 15, 3, 11, 12, 13, 21, 1, 13 \rangle$
- Encoding the tuple:

$$2^{19} \cdot 3^{12} \cdot 5^{11} \cdot 7^{15} \cdot 11^{3} \cdot 13^{11} \cdot 17^{12} \cdot 19^{13} \cdot 23^{21} \cdot 29^{1} \cdot 31^{13}.$$

Longer example pt.2

$$
\begin{aligned}
\ulcorner \forall v_5 (\neg S(v_5) = 0) \urcorner &= \\
\mathcal{F}(\langle \ulcorner \forall \urcorner, \ulcorner v_5 \urcorner, \ulcorner (\urcorner, \ulcorner \neg \urcorner, \ulcorner S \urcorner, \ulcorner (\urcorner, \ulcorner v_5 \urcorner, \ulcorner) \urcorner, \ulcorner = \urcorner, \ulcorner 0 \urcorner, \ulcorner) \urcorner \rangle) &= \\
\mathcal{F}(\langle 19, 12, 11, 15, 3, 11, 12, 13, 21, 1, 13 \rangle) &= \\
2^{19} * 3^{12} * 5^{11} * 7^{15} * 11^{3} * 13^{11} * 17^{12} * 19^{13} * 23^{21} * 29^{1} * 31^{13}
\end{aligned}
$$

Encoding proofs

- A proof is a finite sequence of formulas $\varphi_1, \varphi_2, \ldots, \varphi_k$.

Encoding proofs

- A proof is a finite sequence of formulas $\varphi_1, \varphi_2, \ldots, \varphi_k$.
- Write the whole proof as the expression $\#\varphi_1\#\varphi_2\#\ldots\#\varphi_k\#$ (note $\#$ does not occur inside formulas).

Encoding proofs

- A proof is a finite sequence of formulas $\varphi_1, \varphi_2, \ldots, \varphi_k$.
- Write the whole proof as the expression $\#\varphi_1\#\varphi_2\#\ldots\#\varphi_k\#$ (note $\#$ does not occur inside formulas).
- This is an expression of $\mathcal{L}$, so it has a Gödel code in the same manner.

Takeaways

- We defined a formal language $\mathcal{L}$ for arithmetic (symbols, terms, formulae, sentences).
- We assigned codes to symbols and used prime-power encodings to map tuples of codes to unique natural numbers.
- Both sentences and proofs can be given Gödel codes — the key technical device for letting arithmetic speak about arithmetic.