数理の世界 (数学の考え方) — ゲーデルの不完全性定理

ゲーデル数化と第2不完全性定理， (第 XIII 回の講義)

Sakaé Fuchino (渕野 昌)

Dept. of Computer Sciences
Kobe University
(神戸大学大学院 システム情報学研究科)

http://kurt.scitec.kobe-u.ac.jp/~fuchino/

講義関連資料: http://kurt.scitec.kobe-u.ac.jp/~fuchino/kobe/

(28. Januar 2013 (01:38 JST) version)

神戸大学 2012 年後期の講義
於 K302 教室， 月曜 8:50 – 10:20

January 21, 2013

This presentation is typeset by pLaTeX with beamer class.

ゲーデル数化 (1/7)

- ゲーデル数化 (Gödel numbering) とは，ゲーデルが不完全性定理の証明で導入した，記号列を数にコードしするテクニックのことを言う．
- 記号列に関する述語や言明を，ペアノの算術 (PA を公理系とする形式的体系 — 第 V 回の講義を参照) での論理式や文に翻訳することが目標である．

PA の言語 L_{PA} には定数記号や関数記号として，“ 0 ”, “ S ”, “ $+$ ”, “ $\cdot$ ”だけしか用意されていない．

ゲーデル数化 (1/7)

- ゲーデル数化 (Gödel numbering) とは，ゲーデルが不完全性定理の証明で導入した，記号列を数にコードしするテクニックのことを言う．

- 記号列に関する述語や言明を，ペアノの算術 (PA を公理系とする形式的体系 — 第 V 回の講義を参照) での論理式や文に翻訳することが目標である．

PA の言語 L_{PA} には定数記号や関数記号として，“ 0 ”, “ S ”, “ $+$ ”, “ $\cdot$ ”だけしか用意されていない．

ゲーデル数化 (1/7)

- ゲーデル数化 (Gödel numbering) とは，ゲーデルが不完全性定理の証明で導入した，記号列を数にコードしするテクニックのことを言う．
- 記号列に関する述語や言明を，ペアノの算術 (PA を公理系とする形式的体系 — 第 V 回の講義を参照) での論理式や文に翻訳することが目標である．

PA の言語 L_{PA} には定数記号や関数記号として，“ 0 ”, “ S ”, “ $+$ ”, “ $\cdot$ ”だけしか用意されていない．

ゲーデル数化 (1/7)

- ゲーデル数化 (Gödel numbering) とは，ゲーデルが不完全性定理の証明で導入した，記号列を数にコードするテクニックのことを言う．
- 記号列に関する述語や言明を，ペアノの算術 (PA を公理系とする形式的体系 — 第 V 回の講義を参照) での論理式や文に翻訳することが目標である．

PA の言語 L_{PA} には定数記号や関数記号として，“ 0 ”, “ S ”, “ $+$ ”, “ $\cdot$ ”だけしか用意されていない．

ゲーデル数化 (2/7)

PA の言語 L_{PA} には定数記号や関数記号として，“ 0 ”, “ S ”, “ $+$ ”, “ $\cdot$ ”だけしか用意されていない.

- しかし，L_{PA}-項 $\underbrace{S(S(\cdots S}_{n\text{-回}}(0)\cdots))$ を数 n の表記として使える．数 n に対応するこの項を n の **数表記** とよび $\underline{n}$ と表すことにする.
- PA では指数関数を定義する論理式を作ることができる．つまり，L_{PA} の論理式 $\varphi = \varphi(x, y, z)$ で，任意の数 ℓ, m, n に対して，

$$\mathrm{PA} \vdash \varphi(\underline{\ell}, \underline{m}, \underline{n}) \quad \Leftrightarrow \quad \ell = m^n$$

となるようなものが作れる (これはかなり複雑な論理式になるし，その論理式が上の性質を持つことを示すには,「中国の剰余定理」という数論の定理が必要になる．)

- 以下では，$\underline{\ell} \equiv \underline{m}^{\underline{n}}$ の形の表現を，$\varphi(\underline{\ell}, \underline{m}, \underline{n})$ という形の論理式の略記のことと思って，自由に使うことにする.

ゲーデル数化 (2/7)

PA の言語 L_{PA} には定数記号や関数記号として，" 0 ", " S ", " $+$ ", " $\cdot$ "だけしか用意されていない．

- しかし，L_{PA}-項 $\underbrace{S(S(\cdots S}_{n\text{-回}}(0)\cdots))$ を数 n の表記として使える．数 n に対応するこの項を n の **数表記** とよび $\underline{n}$ と表すことにする．
- PA では指数関数を定義する論理式を作ることができる．つまり，L_{PA} の論理式 $\varphi = \varphi(x, y, z)$ で，任意の数 ℓ, m, n に対して，

$$\mathrm{PA} \vdash \varphi(\underline{\ell}, \underline{m}, \underline{n}) \quad \Leftrightarrow \quad \ell = m^n$$

となるようなものが作れる (これはかなり複雑な論理式になるし，その論理式が上の性質を持つことを示すには,「中国の剰余定理」という数論の定理が必要になる．)

- 以下では，$\underline{\ell} = \underline{m}^{\underline{n}}$ の形の表現を，$\varphi(\underline{\ell}, \underline{m}, \underline{n})$ という形の論理式の略記のことと思って，自由に使うことにする．

ゲーデル数化 (2/7)

PA の言語 L_{PA} には定数記号や関数記号として，“ 0 ”, “ S ”, “ $+$ ”, “ $\cdot$ ”だけしか用意されていない．

- しかし，L_{PA}-項 $\underbrace{S(S(\cdots S}_{n\text{-回}}(0)\cdots))$ を数 n の表記として使える．数 n に対応するこの項を n の **数表記** とよび $\underline{n}$ と表すことにする．
- PA では指数関数を定義する論理式を作ることができる．つまり，L_{PA} の論理式 $\varphi = \varphi(x, y, z)$ で，任意の数 ℓ, m, n に対して，

$$\mathrm{PA} \vdash \varphi(\underline{\ell}, \underline{m}, \underline{n}) \quad \Leftrightarrow \quad \ell = m^n$$

となるようなものが作れる（これはかなり複雑な論理式になるし，その論理式が上の性質を持つことを示すには,「中国の剰余定理」という数論の定理が必要になる．）

- 以下では，$\underline{\ell} \equiv \underline{m}^{\underline{n}}$ の形の表現を，$\varphi(\underline{\ell}, \underline{m}, \underline{n})$ という形の論理式の略記のことと思って，自由に使うことにする．

ゲーデル数化 (2/7)

PA の言語 L_{PA} には定数記号や関数記号として，“ 0 ”, “ S ”, “ $+$ ”, “ $\cdot$ ”だけしか用意されていない.

▷ しかし，L_{PA}-項 $\underbrace{S(S(\cdots S}_{n\text{-回}}(0)\cdots))$ を数 n の表記として使える．数 n に対応するこの項を n の **数表記** とよび $\underline{n}$ と表すことにする.

▷ PA では指数関数を定義する論理式を作ることができる．つまり，L_{PA} の論理式 $\varphi = \varphi(x, y, z)$ で，任意の数 ℓ, m, n に対して，

$$\mathrm{PA} \vdash \varphi(\underline{\ell}, \underline{m}, \underline{n}) \quad \Leftrightarrow \quad \ell = m^n$$

となるようなものが作れる (これはかなり複雑な論理式になるし，その論理式が上の性質を持つことを示すには,「中国の剰余定理」という数論の定理が必要になる．)

▷ 以下では，$\underline{\ell} \equiv \underline{m}^{\underline{n}}$ の形の表現を，$\varphi(\underline{\ell}, \underline{m}, \underline{n})$ という形の論理式の略記のことと思って，自由に使うことにする.

ゲーデル数化 (2/7)

PA の言語 L_{PA} には定数記号や関数記号として，“ 0 ”, “ S ”, “ $+$ ”, “ $\cdot$ ”だけしか用意されていない．

▷ しかし，L_{PA}-項 $\underbrace{S(S(\cdots S}_{n\text{-回}}(0)\cdots))$ を数 n の表記として使える．数 n に対応するこの項を n の **数表記** とよび $\underline{n}$ と表すことにする．

▷ PA では指数関数を定義する論理式を作ることができる．つまり，L_{PA} の論理式 $\varphi = \varphi(x, y, z)$ で，任意の数 ℓ, m, n に対して，

$$\mathrm{PA} \vdash \varphi(\underline{\ell}, \underline{m}, \underline{n}) \quad \Leftrightarrow \quad \ell = m^n$$

となるようなものが作れる (これはかなり複雑な論理式になるし，その論理式が上の性質を持つことを示すには,「中国の剰余定理」という数論の定理が必要になる．)

▶ 以下では，$\underline{\ell} \equiv \underline{m}^{\underline{n}}$ の形の表現を，$\varphi(\underline{\ell}, \underline{m}, \underline{n})$ という形の論理式の略記のことと思って，自由に使うことにする．

ゲーデル数化 (3/7)

- 記号 s や記号列 $\overline{t}$ や記号列の列 $\mathcal{S}$ を，それらをコードする数 $\#(s)$, $\#(\overline{t})$, $\#(\mathcal{S})$ に規則的に対応させる方法のことを **ゲーデル数化** (Gödel numbering) とよぶ.
- $\#(s)$, $\#(\overline{t})$, $\#(\mathcal{S})$ の数表記を，$\ulcorner s \urcorner$, $\ulcorner \overline{t} \urcorner$, $\ulcorner \mathcal{S} \urcorner$ とあらわすことにする.
- ゲーデル数化は色々のやり方が考えられるが，たとえば，次のようにして実現することができる.
- まず，使う記号を記号のカテゴリーごとに一列にならべておく．たとえば，論理記号として使う有限個の記号を $\neg, \wedge, \vee, \ldots$ とならべ，変数記号として使うことにする (無限個の) 記号を $x_0, x_1, x_2, \ldots$ とならべ,... と続ける.
- 記号 s が m 番目のカテゴリーの n 番目の記号のとき，この記号のゲーデル数を $2^0 \cdot 3^m \cdot 5^n$ とする．$2, 3, 5$ は最初の 3 つの素数であることに注意する.

ゲーデル数化 (3/7)

- 記号 s や記号列 $\overline{t}$ や記号列の列 $\mathcal{S}$ を，それらをコードする数 $\#(s)$, $\#(\overline{t})$, $\#(\mathcal{S})$ に規則的に対応させる方法のことを **ゲーデル数化** (Gödel numbering) とよぶ.
- $\#(s)$, $\#(\overline{t})$, $\#(\mathcal{S})$ の数表記を，$\ulcorner s \urcorner$, $\ulcorner \overline{t} \urcorner$, $\ulcorner \mathcal{S} \urcorner$ とあらわすことにする.
- ゲーデル数化は色々のやり方が考えられるが，たとえば，次のようにして実現することができる.
- まず，使う記号を記号のカテゴリーごとに一列にならべておく．たとえば，論理記号として使う有限個の記号を $\neg$, $\wedge$, $\vee$,... とならべ，変数記号として使うことにする (無限個の) 記号を x_0, x_1, x_2,... とならべ,... と続ける.
- 記号 s が m 番目のカテゴリーの n 番目の記号のとき，この記号のゲーデル数を $2^0 \cdot 3^m \cdot 5^n$ とする．2, 3, 5 は最初の 3 つの素数であることに注意する.

ゲーデル数化 (3/7)

- 記号 s や記号列 $\bar{t}$ や記号列の列 $\mathcal{S}$ を，それらをコードする数 $\#(s)$, $\#(\bar{t})$, $\#(\mathcal{S})$ に規則的に対応させる方法のことを **ゲーデル数化** (Gödel numbering) とよぶ.
- $\#(s)$, $\#(\bar{t})$, $\#(\mathcal{S})$ の数表記を，$\ulcorner s \urcorner$, $\ulcorner \bar{t} \urcorner$, $\ulcorner \mathcal{S} \urcorner$ とあらわすことにする.
- ゲーデル数化は色々のやり方が考えられるが，たとえば，次のようにして実現することができる.
- まず，使う記号を記号のカテゴリーごとに一列にならべておく．たとえば，論理記号として使う有限個の記号を $\neg, \wedge, \vee, \ldots$ とならべ，変数記号として使うことにする (無限個の) 記号を $x_0, x_1, x_2, \ldots$ とならべ,... と続ける.
- 記号 s が m 番目のカテゴリーの n 番目の記号のとき，この記号のゲーデル数を $2^0 \cdot 3^m \cdot 5^n$ とする．2, 3, 5 は最初の 3 つの素数であることに注意する.

ゲーデル数化 (3/7)

- 記号 s や記号列 $\overline{t}$ や記号列の列 $\mathcal{S}$ を，それらをコードする数 $\#(s)$, $\#(\overline{t})$, $\#(\mathcal{S})$ に規則的に対応させる方法のことを **ゲーデル数化** (Gödel numbering) とよぶ．
- $\#(s)$, $\#(\overline{t})$, $\#(\mathcal{S})$ の数表記を，$\ulcorner s \urcorner$, $\ulcorner \overline{t} \urcorner$, $\ulcorner \mathcal{S} \urcorner$ とあらわすことにする．
- ゲーデル数化は色々のやり方が考えられるが，たとえば，次のようにして実現することができる．
- まず，使う記号を記号のカテゴリーごとに一列にならべておく．たとえば，論理記号として使う有限個の記号を $\neg$, $\wedge$, $\vee$,... とならべ，変数記号として使うことにする (無限個の) 記号を x_0, x_1, x_2,... とならべ,...と続ける．
- 記号 s が m 番目のカテゴリーの n 番目の記号のとき，この記号のゲーデル数を $2^0 \cdot 3^m \cdot 5^n$ とする．2, 3, 5 は最初の 3 つの素数であることに注意する．

ゲーデル数化 (3/7)

- 記号 s や記号列 $\overline{t}$ や記号列の列 $\mathcal{S}$ を，それらをコードする数 $\#(s)$, $\#(\overline{t})$, $\#(\mathcal{S})$ に規則的に対応させる方法のことを **ゲーデル数化** (Gödel numbering) とよぶ.
- $\#(s)$, $\#(\overline{t})$, $\#(\mathcal{S})$ の数表記を，$\ulcorner s \urcorner$, $\ulcorner \overline{t} \urcorner$, $\ulcorner \mathcal{S} \urcorner$ とあらわすことにする.
- ゲーデル数化は色々のやり方が考えられるが，たとえば，次のようにして実現することができる.
- まず，使う記号を記号のカテゴリーごとに一列にならべておく．たとえば，論理記号として使う有限個の記号を $\neg$, $\wedge$, $\vee$,... とならべ，変数記号として使うことにする (無限個の) 記号を x_0, x_1, x_2,... とならべ,... と続ける.
- 記号 s が m 番目のカテゴリーの n 番目の記号のとき，この記号のゲーデル数を $2^0 \cdot 3^m \cdot 5^n$ とする．2, 3, 5 は最初の 3 つの素数であることに注意する.

ゲーデル数化 (3/7)

- 記号 s や記号列 $\overline{t}$ や記号列の列 $\mathcal{S}$ を，それらをコードする数 $\#(s)$, $\#(\overline{t})$, $\#(\mathcal{S})$ に規則的に対応させる方法のことを **ゲーデル数化** (Gödel numbering) とよぶ.
- $\#(s)$, $\#(\overline{t})$, $\#(\mathcal{S})$ の数表記を，$\ulcorner s \urcorner$, $\ulcorner \overline{t} \urcorner$, $\ulcorner \mathcal{S} \urcorner$ とあらわすことにする.
- ゲーデル数化は色々のやり方が考えられるが，たとえば，次のようにして実現することができる.
- まず，使う記号を記号のカテゴリーごとに一列にならべておく．たとえば，論理記号として使う有限個の記号を $\neg$, $\wedge$, $\vee$,... とならべ，変数記号として使うことにする (無限個の) 記号を x_0, x_1, x_2,... とならべ,... と続ける.
- 記号 s が m 番目のカテゴリーの n 番目の記号のとき，この記号のゲーデル数を $2^0 \cdot 3^m \cdot 5^n$ とする．2, 3, 5 は最初の 3 つの素数であることに注意する.

ゲーデル数化 (4/7)

- 記号 s が m 番目のカテゴリーの n 番目の記号のとき，この記号のゲーデル数 $\#(s)$ を $2^0 \cdot 3^m \cdot 5^n$ とする． 2, 3, 5 は最初の 3 つの素数であることに注意する．
- たとえば，変数記号が 2 番目のカテゴリーの記号としてならべられているときには，"数 k は $2^0 \cdot 3^2 \cdot 5^n$ という形の数に分解される" ということを表現する L_{PA} の論理式 $\varphi = \varphi(k, n)$ を考えると，この論理式は，"k は n 番目の変数記号のコードである" と主張する論理式になっているとみなすことができる．
- 記号列 $s_0 s_1 \cdots s_{n-1}$ に対し，このゲーデル数 $\#(s_0 s_1 \cdots s_{n-1})$ を，数 $2^1 \cdot 3^{\#(s_0)} 5^{\#(s_1)} \ldots {p_n}^{\#(s_{n-1})}$ のこととする．ただし， p_n は n 番目の素数とする． $p_0 = 2$, $p_1 = 3$, $p_2 = 5$, $p_3 = 7, \ldots$ である．ゲーデル数の因数の最初の 2^1 は，この数が記号列をコードしていることを表現するために使われている．

ゲーデル数化 (4/7)

- 記号 s が m 番目のカテゴリーの n 番目の記号のとき，この記号のゲーデル数 $\#(s)$ を $2^0 \cdot 3^m \cdot 5^n$ とする． $2, 3, 5$ は最初の 3 つの素数であることに注意する．
- たとえば，変数記号が 2 番目のカテゴリーの記号としてならべられているときには，“数 k は $2^0 \cdot 3^2 \cdot 5^n$ という形の数に分解される” ということを表現する L_{PA} の論理式 $\varphi = \varphi(k, n)$ を考えると，この論理式は，“k は n 番目の変数記号のコードである” と主張する論理式になっているとみなすことができる．
- 記号列 $s_0 s_1 \cdots s_{n-1}$ に対し，このゲーデル数 $\#(s_0 s_1 \cdots s_{n-1})$ を，数 $2^1 \cdot 3^{\#(s_0)} 5^{\#(s_1)} \ldots {p_n}^{\#(s_{n-1})}$ のこととする．ただし， p_n は n 番目の素数とする． $p_0 = 2$, $p_1 = 3$, $p_2 = 5$, $p_3 = 7, \ldots$ である．ゲーデル数の因数の最初の 2^1 は，この数が記号列をコードしていることを表現するために使われている．

ゲーデル数化 (4/7)

- 記号 s が m 番目のカテゴリーの n 番目の記号のとき，この記号のゲーデル数 $\#(s)$ を $2^0 \cdot 3^m \cdot 5^n$ とする． $2, 3, 5$ は最初の 3 つの素数であることに注意する．
- たとえば，変数記号が 2 番目のカテゴリーの記号としてならべられているときには，“数 k は $2^0 \cdot 3^2 \cdot 5^n$ という形の数に分解される”ということを表現する L_{PA} の論理式 $\varphi = \varphi(k, n)$ を考えると，この論理式は，“k は n 番目の変数記号のコードである”と主張する論理式になっているとみなすことができる．
- 記号列 $s_0 s_1 \cdots s_{n-1}$ に対し，このゲーデル数 $\#(s_0 s_1 \cdots s_{n-1})$ を，数 $2^1 \cdot 3^{\#(s_0)} 5^{\#(s_1)} \ldots {p_n}^{\#(s_{n-1})}$ のこととする．ただし， p_n は n 番目の素数とする． $p_0 = 2$, $p_1 = 3$, $p_2 = 5$, $p_3 = 7, \ldots$ である．ゲーデル数の因数の最初の 2^1 は，この数が記号列をコードしていることを表現するために使われている．

ゲーデル数化 (4/7)

- 記号 s が m 番目のカテゴリーの n 番目の記号のとき，この記号のゲーデル数 $\#(s)$ を $2^0 \cdot 3^m \cdot 5^n$ とする． $2, 3, 5$ は最初の 3 つの素数であることに注意する．
- たとえば，変数記号が 2 番目のカテゴリーの記号としてならべられているときには，"数 k は $2^0 \cdot 3^2 \cdot 5^n$ という形の数に分解される" ということを表現する L_{PA} の論理式 $\varphi = \varphi(k, n)$ を考えると，この論理式は，"k は n 番目の変数記号のコードである" と主張する論理式になっているとみなすことができる．
- 記号列 $s_0 s_1 \cdots s_{n-1}$ に対し，このゲーデル数 $\#(s_0 s_1 \cdots s_{n-1})$ を，数 $2^1 \cdot 3^{\#(s_0)} 5^{\#(s_1)} \ldots {p_n}^{\#(s_{n-1})}$ のこととする．ただし， p_n は n 番目の素数とする． $p_0 = 2$, $p_1 = 3$, $p_2 = 5$, $p_3 = 7, \ldots$ である．ゲーデル数の因数の最初の 2^1 は，この数が記号列をコードしていることを表現するために使われている．

ゲーデル数化 (4/7)

- 記号 s が m 番目のカテゴリーの n 番目の記号のとき，この記号のゲーデル数 $\#(s)$ を $2^0 \cdot 3^m \cdot 5^n$ とする． 2, 3, 5 は最初の 3 つの素数であることに注意する．
- たとえば，変数記号が 2 番目のカテゴリーの記号としてならべられているときには，“数 k は $2^0 \cdot 3^2 \cdot 5^n$ という形の数に分解される” ということを表現する L_{PA} の論理式 $\varphi = \varphi(k, n)$ を考えると，この論理式は，“k は n 番目の変数記号のコードである” と主張する論理式になっているとみなすことができる．
- 記号列 $s_0 s_1 \cdots s_{n-1}$ に対し，このゲーデル数 $\#(s_0 s_1 \cdots s_{n-1})$ を，数 $2^1 \cdot 3^{\#(s_0)} 5^{\#(s_1)} \ldots {p_n}^{\#(s_{n-1})}$ のこととする．ただし， p_n は n 番目の素数とする． $p_0 = 2$, $p_1 = 3$, $p_2 = 5$, $p_3 = 7, \ldots$ である． ゲーデル数の因数の最初の 2^1 は，この数が記号列をコードしていることを表現するために使われている．

ゲーデル数化 (4/7)

- 記号 s が m 番目のカテゴリーの n 番目の記号のとき，この記号のゲーデル数 $\#(s)$ を $2^0 \cdot 3^m \cdot 5^n$ とする． $2, 3, 5$ は最初の 3 つの素数であることに注意する．
- たとえば，変数記号が 2 番目のカテゴリーの記号としてならべられているときには，"数 k は $2^0 \cdot 3^2 \cdot 5^n$ という形の数に分解される" ということを表現する L_{PA} の論理式 $\varphi = \varphi(k, n)$ を考えると，この論理式は，"k は n 番目の変数記号のコードである" と主張する論理式になっているとみなすことができる．
- 記号列 $s_0 s_1 \cdots s_{n-1}$ に対し，このゲーデル数 $\#(s_0 s_1 \cdots s_{n-1})$ を，数 $2^1 \cdot 3^{\#(s_0)} 5^{\#(s_1)} \ldots {p_n}^{\#(s_{n-1})}$ のこととする．ただし， p_n は n 番目の素数とする． $p_0 = 2$, $p_1 = 3$, $p_2 = 5$, $p_3 = 7$,... である．ゲーデル数の因数の最初の 2^1 は，この数が記号列をコードしていることを表現するために使われている．

ゲーデル数化 (4/7)

- 記号 s が m 番目のカテゴリーの n 番目の記号のとき，この記号のゲーデル数 $\#(s)$ を $2^0 \cdot 3^m \cdot 5^n$ とする． $2, 3, 5$ は最初の 3 つの素数であることに注意する．
- たとえば，変数記号が 2 番目のカテゴリーの記号としてならべられているときには，“数 k は $2^0 \cdot 3^2 \cdot 5^n$ という形の数に分解される” ということを表現する L_{PA} の論理式 $\varphi = \varphi(k, n)$ を考えると，この論理式は，“k は n 番目の変数記号のコードである” と主張する論理式になっているとみなすことができる．
- 記号列 $s_0 s_1 \cdots s_{n-1}$ に対し，このゲーデル数 $\#(s_0 s_1 \cdots s_{n-1})$ を，数 $2^1 \cdot 3^{\#(s_0)} 5^{\#(s_1)} \ldots {p_n}^{\#(s_{n-1})}$ のこととする．ただし， p_n は n 番目の素数とする． $p_0 = 2$, $p_1 = 3$, $p_2 = 5$, $p_3 = 7, \ldots$ である．ゲーデル数の因数の最初の 2^1 は，この数が記号列をコードしていることを表現するために使われている．

ゲーデル数化 (5/7)

- 記号列 $s_0 s_1 \cdots s_{n-1}$ に対し，このゲーデル数 $\#(s_0 s_1 \cdots s_{n-1})$ を，数 $2^1 \cdot 3^{\#(s_0)} 5^{\#(s_1)} \ldots p_n{}^{\#(s_{n-1})}$ のこととする．ただし， p_n は n 番目の素数とする． $p_0 = 2$, $p_1 = 3$, $p_2 = 5$, $p_3 = 7, \ldots$ である．ゲーデル数の因数の最初の 2^1 は，この数が記号列をコードしていることを表現するために使われている．
- たとえば，“k は記号列をコードしている” は，

  “k を因数分解したときの 2 の指数は 1 で，因数の全体は素数列の最初の部分になっていて 2 以外の因数の指数はすべて記号をコードする数である”

  を表す L_{PA}-論理式によって表現できる．
- もう少し複雑なトリックがいくつか必要になるが，同様に次を表現する L_{PA}-論理式を作ることができる:

  “k は L_{PA}-項となっている記号列をコードしている”

  “k は L_{PA}-論理式となっている記号列をコードしている”

ゲーデル数化 (5/7)

- 記号列 $s_0 s_1 \cdots s_{n-1}$ に対し，このゲーデル数 $\#(s_0 s_1 \cdots s_{n-1})$ を，数 $2^1 \cdot 3^{\#(s_0)} 5^{\#(s_1)} \dots {p_n}^{\#(s_{n-1})}$ のこととする．ただし， p_n は n 番目の素数とする． $p_0 = 2$, $p_1 = 3$, $p_2 = 5$, $p_3 = 7, \dots$ である．ゲーデル数の因数の最初の 2^1 は，この数が記号列をコードしていることを表現するために使われている．
- たとえば，“k は記号列をコードしている” は，

  “k を因数分解したときの 2 の指数は 1 で，因数の全体は素数列の最初の部分になっていて 2 以外の因数の指数はすべて記号をコードする数である”

  を表す L_{PA}-論理式によって表現できる．
- もう少し複雑なトリックがいくつか必要になるが，同様に次を表現する L_{PA}-論理式を作ることができる:

  “k は L_{PA}-項となっている記号列をコードしている”

  “k は L_{PA}-論理式となっている記号列をコードしている”

ゲーデル数化 (5/7)

- 記号列 $s_0 s_1 \cdots s_{n-1}$ に対し，このゲーデル数 $\#(s_0 s_1 \cdots s_{n-1})$ を，数 $2^1 \cdot 3^{\#(s_0)} 5^{\#(s_1)} \dots p_n{}^{\#(s_{n-1})}$ のこととする．ただし，p_n は n 番目の素数とする．$p_0 = 2$, $p_1 = 3$, $p_2 = 5$, $p_3 = 7, \dots$ である．ゲーデル数の因数の最初の 2^1 は，この数が記号列をコードしていることを表現するために使われている．

- たとえば，“k は記号列をコードしている” は，

  “k を因数分解したときの 2 の指数は 1 で，因数の全体は素数列の最初の部分になっていて 2 以外の因数の指数はすべて記号をコードする数である”

  を表す L_{PA}-論理式によって表現できる．

- もう少し複雑なトリックがいくつか必要になるが，同様に次を表現する L_{PA}-論理式を作ることができる:

  “k は L_{PA}-項となっている記号列をコードしている”

  “k は L_{PA}-論理式となっている記号列をコードしている”

ゲーデル数化 (5/7)

- 記号列 $s_0 s_1 \cdots s_{n-1}$ に対し，このゲーデル数 $\#(s_0 s_1 \cdots s_{n-1})$ を，数 $2^1 \cdot 3^{\#(s_0)} 5^{\#(s_1)} \ldots p_n{}^{\#(s_{n-1})}$ のこととする．ただし，p_n は n 番目の素数とする．$p_0 = 2$, $p_1 = 3$, $p_2 = 5$, $p_3 = 7, \ldots$ である．ゲーデル数の因数の最初の 2^1 は，この数が記号列をコードしていることを表現するために使われている．
- たとえば，“k は記号列をコードしている” は，

  “k を因数分解したときの 2 の指数は 1 で，因数の全体は素数列の最初の部分になっていて 2 以外の因数の指数はすべて記号をコードする数である”

  を表す L_{PA}-論理式によって表現できる．
- もう少し複雑なトリックがいくつか必要になるが，同様に次を表現する L_{PA}-論理式を作ることができる:

  “k は L_{PA}-項となっている記号列をコードしている”

  “k は L_{PA}-論理式となっている記号列をコードしている”

ゲーデル数化 (5/7)

- 記号列 $s_0 s_1 \cdots s_{n-1}$ に対し，このゲーデル数 $\#(s_0 s_1 \cdots s_{n-1})$ を，数 $2^1 \cdot 3^{\#(s_0)} 5^{\#(s_1)} \ldots p_n{}^{\#(s_{n-1})}$ のこととする．ただし，p_n は n 番目の素数とする．$p_0 = 2$, $p_1 = 3$, $p_2 = 5$, $p_3 = 7, \ldots$ である．ゲーデル数の因数の最初の 2^1 は，この数が記号列をコードしていることを表現するために使われている．
- たとえば，“k は記号列をコードしている” は，

  > “k を因数分解したときの 2 の指数は 1 で，因数の全体は素数列の最初の部分になっていて 2 以外の因数の指数はすべて記号をコードする数である”

  を表す L_{PA}-論理式によって表現できる．
- もう少し複雑なトリックがいくつか必要になるが，同様に次を表現する L_{PA}-論理式を作ることができる:

  “k は L_{PA}-項となっている記号列をコードしている”

  “k は L_{PA}-論理式となっている記号列をコードしている”

ゲーデル数化 (6/7)

- もう少し複雑なトリックがいくつか必要になるが，同様に次を表現する L_{PA}-論理式を作ることができる:

  “k は L_{PA}-項となっている記号列をコードしている”

  “k は L_{PA}-論理式となっている記号列をコードしている”

  “k は自由変数 $x_1, \ldots, x_n$ を持つ論理式をコードしていて，$\ell_1, \ldots, \ell_n$ は L_{PA}-項のコードで，m は，k のコードする論理式の変数 $x_1, \ldots, x_n$ に $\ell_1, \ldots, \ell_n$ を代入して得られる L_{PA}-論理式をコードしている”

- $\mathcal{S}$ が記号列 $\overline{t}_0, \overline{t}_1, \ldots, \overline{t}_{n-1}$ の列のとき，$\#(\mathcal{S})$ を $2^2 \cdot 3^{\#(\overline{t}_0)} \cdot 5^{\#(\overline{t}_1)} \cdots {p_n}^{\#(\overline{t}_{n-1})}$ のこととする．

- “k は記号列の列をコードする数である” を表現する L_{PA}-論理式を作ることができる．

- “k は記号列の列をコードする数で，ℓ は L_{PA}-論理式をコードする数で， k のコードする記号列の列は ℓ のコードする論理式の PA からの証明である” を表現する L_{PA} 論理式を作ることができる．

ゲーデル数化 (6/7)

- もう少し複雑なトリックがいくつか必要になるが，同様に次を表現する L_{PA}-論理式を作ることができる:
  "k は L_{PA}-項となっている記号列をコードしている"
  "k は L_{PA}-論理式となっている記号列をコードしている"
  "k は自由変数 $x_1, \ldots, x_n$ を持つ論理式をコードしていて，$\ell_1, \ldots, \ell_n$ は L_{PA}-項のコードで，m は，k のコードする論理式の変数 $x_1, \ldots, x_n$ に $\ell_1, \ldots, \ell_n$ を代入して得られる L_{PA}-論理式をコードしている"
- $\mathcal{S}$ が記号列 $\overline{t}_0, \overline{t}_1, \ldots, \overline{t}_{n-1}$ の列のとき，$\#(\mathcal{S})$ を $2^2 \cdot 3^{\#(\overline{t}_0)} \cdot 5^{\#(\overline{t}_1)} \cdots p_n{}^{\#(\overline{t}_{n-1})}$ のこととする.
- "k は記号列の列をコードする数である" を表現する L_{PA}-論理式を作ることができる.
- "k は記号列の列をコードする数で，ℓ は L_{PA}-論理式をコードする数で，k のコードする記号列の列は ℓ のコードする論理式の PA からの証明である" を表現する L_{PA} 論理式を作ることができる.

ゲーデル数化 (6/7)

- ▷ もう少し複雑なトリックがいくつか必要になるが，同様に次を表現する L_{PA}-論理式を作ることができる:

  “k は L_{PA}-項となっている記号列をコードしている”

  “k は L_{PA}-論理式となっている記号列をコードしている”

  “k は自由変数 $x_1, \dots, x_n$ を持つ論理式をコードしていて，$\ell_1, \dots, \ell_n$ は L_{PA}-項のコードで，m は，k のコードする論理式の変数 $x_1, \dots, x_n$ に $\ell_1, \dots, \ell_n$ を代入して得られる L_{PA}-論理式をコードしている”

- ▶ $\mathcal{S}$ が記号列 $\overline{t}_0, \overline{t}_1, \dots, \overline{t}_{n-1}$ の列のとき，$\#(\mathcal{S})$ を $2^2 \cdot 3^{\#(\overline{t}_0)} \cdot 5^{\#(\overline{t}_1)} \cdots {p_n}^{\#(\overline{t}_{n-1})}$ のこととする．

- ▷ “k は記号列の列をコードする数である” を表現する L_{PA}-論理式を作ることができる．

- ▷ “k は記号列の列をコードする数で，ℓ は L_{PA}-論理式をコードする数で，k のコードする記号列の列は ℓ のコードする論理式の PA からの証明である” を表現する L_{PA} 論理式を作ることができる．

ゲーデル数化 (6/7)

- ▷ もう少し複雑なトリックがいくつか必要になるが，同様に次を表現する L_{PA}-論理式を作ることができる:

  “k は L_{PA}-項となっている記号列をコードしている”

  “k は L_{PA}-論理式となっている記号列をコードしている”

  “k は自由変数 $x_1, \ldots, x_n$ を持つ論理式をコードしていて，$\ell_1, \ldots, \ell_n$ は L_{PA}-項のコードで，m は，k のコードする論理式の変数 $x_1, \ldots, x_n$ に $\ell_1, \ldots, \ell_n$ を代入して得られる L_{PA}-論理式をコードしている”

- ▶ $\mathcal{S}$ が記号列 $\overline{t}_0, \overline{t}_1, \ldots, \overline{t}_{n-1}$ の列のとき，$\#(\mathcal{S})$ を $2^2 \cdot 3^{\#(\overline{t}_0)} \cdot 5^{\#(\overline{t}_1)} \cdots {p_n}^{\#(\overline{t}_{n-1})}$ のこととする．

- ▷ “k は記号列の列をコードする数である” を表現する L_{PA}-論理式を作ることができる．

- ▷ “k は記号列の列をコードする数で，ℓ は L_{PA}-論理式をコードする数で，k のコードする記号列の列は ℓ のコードする論理式の PA からの証明である” を表現する L_{PA} 論理式を作ることができる．

ゲーデル数化 (6/7)

▷ もう少し複雑なトリックがいくつか必要になるが，同様に次を表現する L_{PA}-論理式を作ることができる:

"k は L_{PA}-項となっている記号列をコードしている"

"k は L_{PA}-論理式となっている記号列をコードしている"

"k は自由変数 $x_1, \ldots, x_n$ を持つ論理式をコードしていて，$\ell_1, \ldots, \ell_n$ は L_{PA}-項のコードで，m は，k のコードする論理式の変数 $x_1, \ldots, x_n$ に $\ell_1, \ldots, \ell_n$ を代入して得られる L_{PA}-論理式をコードしている"

▶ $\mathcal{S}$ が記号列 $\bar{t}_0, \bar{t}_1, \ldots, \bar{t}_{n-1}$ の列のとき，$\#(\mathcal{S})$ を $2^2 \cdot 3^{\#(\bar{t}_0)} \cdot 5^{\#(\bar{t}_1)} \cdots {p_n}^{\#(\bar{t}_{n-1})}$ のこととする.

▷ "k は記号列の列をコードする数である" を表現する L_{PA}-論理式を作ることができる.

▷ "k は記号列の列をコードする数で，ℓ は L_{PA}-論理式をコードする数で，k のコードする記号列の列は ℓ のコードする論理式の PA からの証明である" を表現する L_{PA} 論理式を作ることができる.

ゲーデル数化 (7/7)

- "k は記号列の列をコードする数で，ℓ は L_{PA}-論理式をコードする数で，k のコードする記号列の列は ℓ のコードする論理式の PA からの証明である" を表現する L_{PA}-論理式を作ることができる．
- 上のような L_{PA}-論理式を，$proof_{\mathrm{PA}}(k, \ell)$ と書くことにする．この論理式は次のような意味で，上の "..." をよく表現するものとしてとることができる:

  PA $\vdash^P \varphi$ なら，PA $\vdash proof_{\mathrm{PA}}(\ulcorner P \urcorner, \ulcorner \varphi \urcorner)$ である．逆に PA $\vdash^P \varphi$ でないなら，PA $\vdash \neg proof_{\mathrm{PA}}(\ulcorner P \urcorner, \ulcorner \varphi \urcorner)$ である．

- 上に用意したことと Diagonal Lemma を用いると，第 1 不完全性定理が証明できることを前回見た．Diagonal Lemma の証明は講義の資料としてあげた，執筆中の本の，原稿の一部を参照．

ゲーデル数化 (7/7)

▷ "k は記号列の列をコードする数で，ℓ は L_{PA}-論理式をコードする数で， k のコードする記号列の列は ℓ のコードする論理式の PA からの証明である" を表現する L_{PA}-論理式を作ることができる.

▶ 上のような L_{PA}-論理式を，$\mathit{proof}_{\mathrm{PA}}(k, \ell)$ と書くことにする．この論理式は次のような意味で，上の "…" をよく表現するものとしてとることができる:

> PA $\vdash^P \varphi$ なら，PA $\vdash \mathit{proof}_{\mathrm{PA}}(\ulcorner P \urcorner, \ulcorner \varphi \urcorner)$ である．逆に PA $\vdash^P \varphi$ でないなら，PA $\vdash \neg \mathit{proof}_{\mathrm{PA}}(\ulcorner P \urcorner, \ulcorner \varphi \urcorner)$ である.

▶ 上に用意したことと Diagonal Lemma を用いると，第 1 不完全性定理が証明できることを前回見た．Diagonal Lemma の証明は講義の資料としてあげた，執筆中の本の，原稿の一部を参照.

ゲーデル数化 (7/7)

- “k は記号列の列をコードする数で，ℓ は L_{PA}-論理式をコードする数で， k のコードする記号列の列は ℓ のコードする論理式の PA からの証明である” を表現する L_{PA}-論理式を作ることができる．
- 上のような L_{PA}-論理式を，$\mathit{proof}_{\mathrm{PA}}(k, \ell)$ と書くことにする．この論理式は次のような意味で，上の “...” をよく表現するものとしてとることができる:

> $\mathrm{PA} \vdash^{P} \varphi$ なら，$\mathrm{PA} \vdash \mathit{proof}_{\mathrm{PA}}(\ulcorner P \urcorner, \ulcorner \varphi \urcorner)$ である．逆に $\mathrm{PA} \vdash^{P} \varphi$ でないなら，$\mathrm{PA} \vdash \neg \mathit{proof}_{\mathrm{PA}}(\ulcorner P \urcorner, \ulcorner \varphi \urcorner)$ である．

- 上に用意したことと Diagonal Lemma を用いると，第 1 不完全性定理が証明できることを前回見た．Diagonal Lemma の証明は講義の資料としてあげた，執筆中の本の，原稿の一部を参照．

ゲーデル数化 (7/7)

- “k は記号列の列をコードする数で，ℓ は L_{PA}-論理式をコードする数で， k のコードする記号列の列は ℓ のコードする論理式の PA からの証明である” を表現する L_{PA}-論理式を作ることができる．
- 上のような L_{PA}-論理式を，$proof_{\mathsf{PA}}(k,\ell)$ と書くことにする．この論理式は次のような意味で，上の “...” をよく表現するものとしてとることができる:

  PA $\vdash^P \varphi$ なら，PA $\vdash proof_{\mathsf{PA}}(\ulcorner P \urcorner, \ulcorner \varphi \urcorner)$ である．逆に PA $\vdash^P \varphi$ でないなら，PA $\vdash \neg proof_{\mathsf{PA}}(\ulcorner P \urcorner, \ulcorner \varphi \urcorner)$ である．

- 上に用意したことと Diagonal Lemma を用いると，第 1 不完全性定理が証明できることを前回見た．Diagonal Lemma の証明は講義の資料としてあげた，執筆中の本の，原稿の一部を参照．

第２不完全性定理

- 上で導入した $proof_{\mathrm{PA}}(k,\ell)$ と同様の論理式を，具体的に書くことのできる理論 T に対しても $proof_T(k,\ell)$ として書くことができる．
- $\exists k proof_T(k,\ell)$ を $prov_T(\ell)$ と書くことにする．$prov_T(\ell)$ は，ℓ のコードする論理式が T で証明できることを主張する L_{PA}-論理式となっている．
- $\neg prov_T(\ulcorner x \neq x \urcorner)$ を考えると，これは T が矛盾しない (consistent) ことを主張する L_{PA}-文になっていることがわかる．この文のことを $consis_T$ と書くことにする．

定理 (第２不完全性定理)． PA を含む (あるいは PA がそこで解釈できる) 具体的に与えられた理論 T が無矛盾なら，$consis_T$ は T で証明できない．

第２不完全性定理

- 上で導入した $proof_{\mathrm{PA}}(k,\ell)$ と同様の論理式を，具体的に書くことのできる理論 T に対しても $proof_T(k,\ell)$ として書くことができる．
- $\exists k proof_T(k,\ell)$ を $prov_T(\ell)$ と書くことにする． $prov_T(\ell)$ は，ℓ のコードする論理式が T で証明できることを主張する L_{PA}-論理式となっている．
- $\neg prov_T(\ulcorner x \neq x \urcorner)$ を考えると，これは T が矛盾しない (consistent) ことを主張する L_{PA}-文になっていることがわかる．この文のことを $consis_T$ と書くことにする．

定理 (第２不完全性定理). PA を含む (あるいは PA がそこで解釈できる) 具体的に与えられた理論 T が無矛盾なら，$consis_T$ は T で証明できない．

第2不完全性定理

- 上で導入した $proof_{PA}(k,\ell)$ と同様の論理式を，具体的に書くことのできる理論 T に対しても $proof_T(k,\ell)$ として書くことができる．
- $\exists k proof_T(k,\ell)$ を $prov_T(\ell)$ と書くことにする． $prov_T(\ell)$ は，ℓ のコードする論理式が T で証明できることを主張する L_{PA}-論理式となっている．
- $\neg prov_T(\ulcorner x \not\equiv x \urcorner)$ を考えると，これは T が矛盾しない (consistent) ことを主張する L_{PA}-文になっていることがわかる． この文のことを $consis_T$ と書くことにする．

定理 (第2不完全性定理). PA を含む (あるいは PA がそこで解釈できる) 具体的に与えられた理論 T が無矛盾なら，$consis_T$ は T で証明できない．

第２不完全性定理

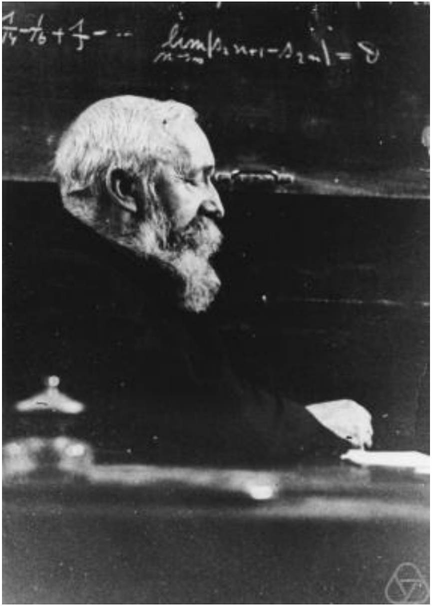
- 上で導入した $proof_{PA}(k, \ell)$ と同様の論理式を，具体的に書くことのできる理論 T に対しても $proof_T(k, \ell)$ として書くことができる．
- $\exists k proof_T(k, \ell)$ を $prov_T(\ell)$ と書くことにする． $prov_T(\ell)$ は， ℓ のコードする論理式が T で証明できることを主張する L_{PA}-論理式となっている．
- $\neg prov_T(\ulcorner x \not\equiv x \urcorner)$ を考えると， これは T が矛盾しない (consistent) ことを主張する L_{PA}-文になっていることがわかる． この文のことを $consis_T$ と書くことにする．

定理 (第２不完全性定理)． PA を含む (あるいは PA がそこで解釈できる) 具体的に与えられた理論 T が無矛盾なら， $consis_T$ は T で証明できない．

第２不完全性定理

- 上で導入した $proof_{\mathrm{PA}}(k,\ell)$ と同様の論理式を，具体的に書くことのできる理論 T に対しても $proof_T(k,\ell)$ として書くことができる．
- $\exists k proof_T(k,\ell)$ を $prov_T(\ell)$ と書くことにする． $prov_T(\ell)$ は，ℓ のコードする論理式が T で証明できることを主張する L_{PA}-論理式となっている．
- $\neg prov_T(\ulcorner x \not\equiv x \urcorner)$ を考えると，これは T が矛盾しない (consistent) ことを主張する L_{PA}-文になっていることがわかる．この文のことを $consis_T$ と書くことにする．

定理 (第２不完全性定理)． PA を含む (あるいは PA がそこで解釈できる) 具体的に与えられた理論 T が無矛盾なら，$consis_T$ は T で証明できない．



指数関数が PA で論理式で表せることの説明で「中国の剰余定理」について触れた．ゲーデルはウィーン大学の学部生のころ，P. フルトヴェングラー (1869(明治 2) – 1940(昭和 15), 指揮者フルトヴェングラーのいとこ) の講義を聞いて感銘を受け物理から数学に転科している．

中国の剰余定理はゲーデルの不完全性定理の証明のキーになるが，ゲーデルが中国の剰余定理を習ったのはこのフルトヴェングラーの講義だったことが知られている