Goedel's Theorem 6 (abridged)

Lecture Contents

- L_A can express all recursive (and p.r.) functions
- The role of the β-function trick in proving that result
- In fact, Σ_1 wffs suffice to express all recursive functions
- **Q** can capture all recursive functions
- Expressing and capturing recursive properties and relations.

Sufficiently strong

- Goedel's theorem was about 'sufficiently strong' formal theories.
- Theories that can capture all decidable relations are sufficiently strong.
- The language L_A in fact can express all recursive functions; and even the weak theory $\mathbf{Q}$ can capture them all.
- This is enough for Goedel's result. In fact, for $\mathbf{Q}$ and PA it is enough that the p.r. functions can be captured.

Expressing recursive functions

- If k-ary function f is p.r. then there is an L_A formula ϕ such that:

$$\phi[\overline{n}_1, \ldots, \overline{n}_k, \overline{m}] \text{ is true iff } f(n_1 \ldots n_k) = m$$

where at most $x_1 \ldots x_n$ are free in ϕ.

Expressing recursive functions

- Given the definition of recursive functions it is enough to show that:
  1. L_A can express the initial functions.
  2. If L_A can express g and h then it can express any function f defined by their composition.
  3. If L_A can express g and h then it can express any function f defined by primitive recursion on g and h.
  4. If L_A can express g then it can express any function f defined by minimisation on g.
- Then we will have shown that L_A expresses all recursive functions.

Capturing recursive functions (the tedious way)

- It is enough to show that:
  1. **Q** can capture the initial functions.
  2. If **Q** can capture g and h then it can capture any function f defined by their composition.
  3. If **Q** can capture g and h then it can capture any function f defined by primitive recursion on g and h.
  4. If **Q** can capture g then it can capture any function f defined by minimisation on g.
- Then we will have shown that **Q** captures all recursive functions.