Computer programs and natural number arithmetic

Wong Tin Lok

27 July, 2011

Plan

- What is a computer program?
- What can(not) be achieved by a computer program?
- Can a computer solve Diophantine equations?

Characteristics of a program

A (computer) program...

- is a *finite* sequence of instructions that can be run by a machine;
- can take a *finite* piece of data as input;
- can return a *finite* piece of information as output after a *finite* amount of time;
- may perform an infinite loop without giving an output (depending on the input).

Deciding a set of natural numbers

Convention

$\mathbb{N} = \{0, 1, 2, \ldots\}$ and $k \in \mathbb{N}$.

Definition

A set $A \subseteq \mathbb{N}^k$ is *decidable* if there is a program $\mathcal{P}$ which when given any input $\bar{x} \in \mathbb{N}^k$,

- outputs TRUE if $\bar{x} \in A$;
- outputs FALSE if $\bar{x} \notin A$.

Global assumptions

- All our programs take a fixed *finite* number of natural numbers as inputs.
- Outputs of our programs can only be TRUE or FALSE, if any.
- There is *no* time or memory restriction on our computers.

We are only interested in the relationship between the inputs and the outputs of our programs.

What is a program?

Programs can use

- variables $v_0, v_1, \ldots$;
- assignments $\leftarrow$;
- conditionals and Boolean operations **if-then**, **&**, **∨**, **not**, $=$;
- arithmetical operations and relations 0, 1, $+$, $\times$, $\leqslant$;
- **for** loops; and
- **repeat-until** loops.

} restricted versions

Church–Turing Thesis (Definition of a program)

Every program is equivalent to one that uses *only* the above.

Example: prime numbers

Consider $A = \{x \in \mathbb{N} : x \text{ is a prime number}\}$

$$= \left\{ x \in \mathbb{N} \;\middle|\; \begin{array}{l} \forall u,v \leqslant x \; \big(x = uv \to u = 1 \vee v = 1\big) \\ \;\&\; x \neq 1 \end{array} \right\}.$$

input $x \in \mathbb{N}$.
answer $\leftarrow$ TRUE.
for $u \leftarrow 0, \ldots, x$ **do**:
  for $v \leftarrow 0, \ldots, x$ **do**:
    if $x = uv$ & $u \neq 1$ & $v \neq 1$, **then** *answer* $\leftarrow$ FALSE.
if $x = 1$, **then** *answer* $\leftarrow$ FALSE.
output *answer*.

No **repeat-until** is involved.

Observation

This program stops on *all* inputs.

Exercise: coprime numbers

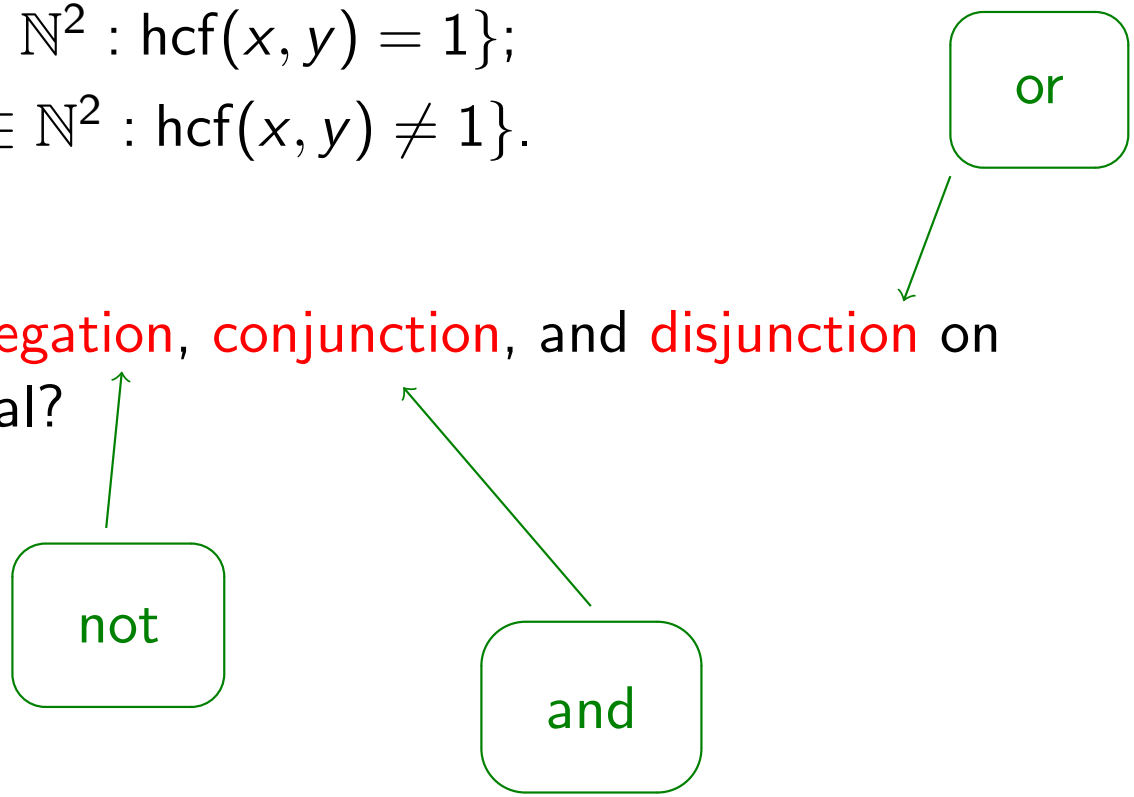
Write programs that decide

(a) $B = \{(x, y) \in \mathbb{N}^2 : \mathsf{hcf}(x, y) = 1\}$;

(b) $B' = \{(x, y) \in \mathbb{N}^2 : \mathsf{hcf}(x, y) \neq 1\}$.

Question

How to perform negation, conjunction, and disjunction on programs in general?



Example: Fermat Last Theorem

Consider $C_{17} = \left\{x \in \mathbb{N} : \exists z\ \exists y \leqslant z\ (y \neq 0\ \&\ x^{17} + y^{17} = z^{17})\right\}$.

input $x \in \mathbb{N}$.
repeat, **starting with** $z \leftarrow 0$:
  for $y \leftarrow 0, \ldots, z$ **do**:
    if $y \neq 0$ **&** $x^{17} + y^{17} = z^{17}$, **then** *answer* $\leftarrow$ TRUE.
next z **until** *answer* $=$ TRUE.
output *answer*.

Observation

Wiles and Taylor (1995) proved that $C_{17} = \{0\}$. However, the program above does not stop on input 1, for example!

Semidecidability

(*Semi-* means *half-*.)

Definition

A set $A \subseteq \mathbb{N}^k$ is *semidecidable* if there is a program $\mathcal{P}$ such that

$$A = \{\bar{x} \in \mathbb{N}^k : \mathcal{P} \text{ outputs } \textsc{true} \text{ on input } \bar{x}\}.$$

Example

C_{17} is semidecidable.

Remark

All decidable sets are semidecidable.

Programs and arithmetic

Programs	Formulas in arithmetic
if-then, **&**, **∨**, **not**	$\rightarrow$, &, $\vee$, $\neg$
$0, 1, +, \times, \leqslant$	$0, 1, +, \times, \leqslant$
for loops	$\forall v \leqslant t$ and $\exists v \leqslant t$
repeat-until loops	$\exists v$

Correspondence Principle

The above table gives a correspondence between programs $\mathcal{P}$ and formulas $\varphi(\bar{x})$ in arithmetic such that

$$\{\bar{x} \in \mathbb{N} : \mathcal{P} \text{ outputs } \textsc{true} \text{ on input } \bar{x}\} = \{\bar{x} \in \mathbb{N} : \varphi(\bar{x})\}.$$

Decidability is arithmetical

Definition

Formulas that correspond to a program are called Σ_1. They are of the form

$$\exists \bar{v}\ \chi(\bar{v}, \bar{x}),$$

where $\chi(\bar{v}, \bar{x})$ is made up of the Boolean operations, the arithmetical operations and relations, and *bounded quantifiers* (i.e., $\forall v \leqslant t$ and $\exists v \leqslant t$).

Correspondence Principle (restated)

The semidecidable sets are exactly those of the form

$$\{\bar{x} \in \mathbb{N}^k : \varphi(\bar{x})\},$$

where $\varphi(\bar{x})$ is Σ_1.

Example: an infinite loop

input $x \in \mathbb{N}$.
answer $\leftarrow$ FALSE.
repeat, **starting with** $z \leftarrow 0$:
  if $z \neq z$, **then** *answer* $\leftarrow$ TRUE.
next z **until** *answer* $=$ TRUE.
output *answer*.

This corresponds to the formula

$$\exists z\ (z \neq z).$$

Undecidable sets

Question

Are there undecidable sets?

Answer

Yes, by counting.

Question

Are all semidecidable sets decidable?

Remark

Our previous example C_{17} of a semidecidable set is decidable.

The Halting Set

(*Halt* means *stop*.)

Definition

$$H = \{(\ulcorner \mathcal{P} \urcorner, x) \in \mathbb{N}^2 : \text{the program } \mathcal{P} \text{ stops on input } x\}.$$

Observation

Every program $\mathcal{P}$ is a natural number, which we denote by $\ulcorner \mathcal{P} \urcorner$.

Proposition

H is semidecidable.

Proof.

Given an input $(\ulcorner \mathcal{P} \urcorner, x) \in \mathbb{N}^2$, run the program $\mathcal{P}$ on input x. $\square$

Theorem

H is not decidable.

The Halting Set H is not decidable

Proof

Suppose there is a program $\mathcal{K}$ that decides H. Let $\mathcal{L}$ be the following program.

> **input** $\ulcorner \mathcal{P} \urcorner \in \mathbb{N}$.
> **if** $\mathcal{P}$ `stops on input` $\ulcorner \mathcal{P} \urcorner$,
> **then** `perform an infinite loop`,
> **else** *answer* $\leftarrow$ TRUE.
> **output** *answer*.

Then $\mathcal{L}$ stops on input $\ulcorner \mathcal{P} \urcorner$

$\Leftrightarrow$ $\mathcal{P}$ does *not* stop on input $\ulcorner \mathcal{P} \urcorner$.

Question

What happens when we run $\mathcal{L}$ on input $\ulcorner \mathcal{L} \urcorner$?

Russell's Paradox!

Running $\mathcal{L}$ on input $\ulcorner \mathcal{L} \urcorner$

Proof (continued).

Recall that $\mathcal{L}$ stops on input $\ulcorner \mathcal{P} \urcorner$

$\Leftrightarrow$ $\mathcal{P}$ does *not* stop on input $\ulcorner \mathcal{P} \urcorner$.

- If $\mathcal{L}$ stops on input $\ulcorner \mathcal{L} \urcorner$, then $\mathcal{L}$ does not stop on input $\ulcorner \mathcal{L} \urcorner$.
- If $\mathcal{L}$ does not stop on input $\ulcorner \mathcal{L} \urcorner$, then $\mathcal{L}$ stops on input $\ulcorner \mathcal{L} \urcorner$.

This is a contradiction. □

No program can check whether
a program stops on an input.

How mathematical is H?

The halting set is not quite a 'mathematically interesting' object.

Open problem

Find a 'mathematically interesting' semidecidable set that is not decidable.

Gödel's Incompleteness Theorem

Gödel Incompleteness Theorem (1931, rudimentary form)

No program can tell us the truth of any given statement about $\mathbb{N}$.

Proof.

- By the Correspondence Principle, there is a formula $\varphi(x, y)$ in arithmetic that corresponds to the Halting Set H.
- Since H is not decidable, there is no program that tells us the truth of $\varphi(x, y)$ correctly for all $x, y \in \mathbb{N}$. $\square$

Diophantine equations

Diophantine equations are polynomial equations with *integer* coefficients to be solved in the *integers* (or in the natural numbers).

Examples

- $\{(x, y, z) \in \mathbb{N}^3 : x^2 + y^2 = z^2\}$.
- $\{(x, y, z, w) \in \mathbb{N}^4 : x^4 + y^4 + z^4 + w^4 = (x + y + z + w)^4\}$.
- $\{(a, b, c) \in \mathbb{N}^3 : \exists x \ (ax^2 + bx + c = 0)\}$.
- $\{(x, y, z) \in \mathbb{N}^3 : \exists w \ (x^3 + y^3 + z^3 = w^3)\}$.

Diophantine sets

Definition

A set $A \subseteq \mathbb{N}^k$ is *Diophantine* if

$$A = \{\bar{x} \in \mathbb{N}^k : \exists \bar{v}\ p(\bar{x}, \bar{v}) = q(\bar{x}, \bar{v})\}$$

for some polynomials p, q with coefficients in $\mathbb{N}$.

Exercise

Convince yourself that every Diophantine set is semidecidable.

Hilbert's Tenth Problem

Hilbert's Tenth Problem (1900)

Find a general algorithm to solve Diophantine equations.

Hilbert's Tenth Problem (simplified form)

Show that every Diophantine set is decidable.

Solution (MRDP 1970)

This is *not* possible!

The solution

MRDP Theorem (Matiyasevich–Robinson–Davis–Putnam)

Every semidecidable set is Diophantine.

Corollary

There is an undecidable Diophantine set.

Proof.

By the MRDP Theorem, the Halting Set H is a Diophantine set that is not decidable. □

How mathematical is the Halting Set?

Theorem (Jones 1980)

$H = \{(\rho, x) \in \mathbb{N}^2 : \psi(\rho, x)\}$, where $\psi(\rho, x)$ is

$$\begin{aligned}
&\exists a \exists b \exists c \exists d \exists e \exists f \exists g \exists h \exists i \exists j \exists k \exists l \exists m \exists n \exists o \exists p \exists q \exists r \exists s \exists t \exists u \exists w \exists y \exists z \exists \alpha \exists \gamma \exists \eta \exists \theta \exists \lambda \exists \tau \exists \varphi \\
&\quad \big(elg^2 + \alpha - (b - xy)q^2\big)\big(q - b^{5^{60}}\big)\big(\lambda + q^4 - 1 - \lambda b^5\big)\big(k - r - 1 - hp - h\big) \\
&\quad \big(l - u - t\theta\big)\big(e - y - m\theta\big)\big(n - q^{16}\big)\big(p - 2ws^2r^2n^2\big)\big(p^2k^2 - k^2 + 1 - \tau^2\big) \\
&\quad \big(c - 2r - 1 - \varphi\big)\big(\theta + 2z - b^5\big)\big(\big((zuy)^2 + u\big)^2 + y - \rho\big)\big(a - (wn^2 + 1)rsn^2\big) \\
&\quad \big(4(c - ksn^2)^2 + \eta - k^2\big)\big(d - bw - ca + 2c - 4\alpha\gamma + 5\gamma\big)\big(d^2 - (a^2 - 1)c^2 - 1\big) \\
&\quad \big(f^2 - (a^2 - 1)i^2c^4 - 1\big)\big((d + of)^2 - \big((a + f^2(d^2 - a)\big)^2 - 1\big)(2r + 1 + jc)^2 - 1\big) \\
&\quad \left(\begin{array}{l} \big(g + eq^3 + lq^5 + \big(2(e - z\lambda)(1 + xb^5 + g)^4 + \lambda b^5 + \lambda b^5 q^4\big)q^4\big)(n^2 - n) \\ \quad + \big(q^3 - bl + l + \theta\lambda q^3 + (b^5 - 2)q^5\big)(n^2 - 1) - r \end{array}\right) = 0.
\end{aligned}$$

Connections with complexity theory

MRDP Theorem

Every semidecidable set is Diophantine.

Definition

A formula is Σ_0 if it is made up of the Boolean operations, the arithmetical operations and relations, and bounded quantifiers.

Open question (Paris)

Is the MRDP Theorem provable using only Σ_0-induction?

Observation (Wilkie)

If the answer is yes, then NP = co-NP.

Conclusion

- There is a correspondence between programs and arithmetic.
- Decidability is an arithmetical property.
- The Halting Set is an important undecidable set.
- An analysis of undecidable sets results in number-theoretic problems that cannot be solved by a program.

Webpage: http://mat140.bham.ac.uk/∼wongtl/
E-mail: wongtl03c62@gmail.com