The above proof is not very satisfactory because it does not give a concrete example of an undecidable set. We will now define the halting problem and prove it undecidable. The halting problem plays an important role in computability theory. In order to do that, we make some preliminary observations first: since there are only countably many operator terms that define partial functions of arity 1, there is a bijection from some subset C of $\mathbb{N}$, the set of "codes", to the set of such operator terms. For $e \in C$ we will write φ_e for the partial recursive function defined by the operator term with code e. For the time being, it is irrelevant which set C and which mapping $e \mapsto \varphi_e$ we pick. In Section 1.4 it will become relevant and we will give a concrete definition of C and the mapping $e \mapsto \varphi_e$.

Definition 1.9. The *halting problem* is $H = \{(e, x) \in C \times \mathbb{N} \mid \varphi_e(x) \downarrow\}$. Moreover, we define $K = \{e \in C \mid \varphi_e(e) \downarrow\}$.

Theorem 1.2. *K is undecidable.*

Proof. Define $f : \mathbb{N} \hookrightarrow \mathbb{N}$ by

$$f(n) = \begin{cases} 0 & \text{if } n \notin K \\ \text{undefined} & \text{if } n \in K \end{cases}$$

Suppose that K is decidable, i.e., χ_K is recursive, then, by Lemma 1.2, f is partial recursive. Let e be s.t. $f = \varphi_e$. Then we have

$$e \in K \overset{\text{Def. } f}{\iff} f(e) = \varphi_e(e) \text{ is undefined} \overset{\text{Def. } K}{\iff} e \notin K$$

which is a contradiction. □

Corollary 1.1. *H is undecidable.*

Proof. Suppose $\chi_H : \mathbb{N}^2 \to \mathbb{N}$ would be recursive, then so would be $\chi_K : \mathbb{N} \to \mathbb{N}$ because $\chi_K(x) = \chi_H(x, x)$. □

The main aim of the rest of this chapter is to prove that K is recursively enumerable, i.e., that there is a total recursive function $f : \mathbb{N} \to \mathbb{N}$ s.t. $f(\mathbb{N}) = K$. The existence of a recursively enumerable and undecidable set will then allow to obtain a first, weak, version of the first incompleteness theorem. In order to establish recursive enumerability of K we will have to code operator terms and computations as natural numbers. This technique, arithmetisation or "Gödelisation", in particular when used in conjunction with diagonalisation, is central, not only for the proof of the incompleteness theorems but for many results in mathematical logic.

1.3 Coding pairs, tuples, and trees

We will develop our coding machinery on a sufficiently general level to allow its reuse later when we code formulas and proofs. We start in this section with coding pairs, tuples, and trees. Before we do so, we need some more closure properties of the primitive recursive functions.

Lemma 1.3. *If $f : \mathbb{N}^{k+1} \to \mathbb{N}$ is primitive recursive, then so are:*

1. $(\overline{x}, z) \mapsto \sum_{y=0}^{z} f(\overline{x}, y)$,
2. $(\overline{x}, z) \mapsto \prod_{y=0}^{z} f(\overline{x}, y)$,

3. $(\overline{x}, z) \mapsto \min\{f(\overline{x}, y) \mid 0 \leq y \leq z\}$, *and*

4. $(\overline{x}, z) \mapsto \max\{f(\overline{x}, y) \mid 0 \leq y \leq z\}$.

Assuming in addition that $f : \mathbb{N}^{k+1} \to \{0, 1\}$, *so are:*

5. $(\overline{x}, z) \mapsto \forall y \leq z\, f(\overline{x}, y) = \begin{cases} 1 & \text{if for all } y \in \{0, \ldots, z\}\text{: } f(\overline{x}, y) = 1 \\ 0 & \text{if there is } y \in \{0, \ldots, z\} \text{ s.t. } f(\overline{x}, y) = 0 \end{cases}$,

6. $(\overline{x}, z) \mapsto \exists y \leq z\, f(\overline{x}, y) = \begin{cases} 1 & \text{if there is } y \in \{0, \ldots, z\} \text{ s.t. } f(\overline{x}, y) = 1 \\ 0 & \text{if for all } y \in \{0, \ldots, z\}\text{: } f(\overline{x}, y) = 0 \end{cases}$, *and*

7. $(\overline{x}, z) \mapsto (\mu y \leq z) f(\overline{x}, y) = \begin{cases} \text{the least } y \leq z \text{ s.t. } f(\overline{x}, y) = 1 & \text{if such a } y \text{ exists} \\ 0 & \text{otherwise} \end{cases}$.

Proof. For 1., note that the finite sum can be defined with primitive recursion as

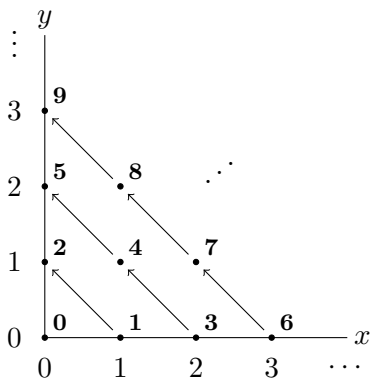
$$\sum_{y=0}^{0} f(\overline{x}, y) = f(\overline{x}, 0)$$
$$\sum_{y=0}^{z+1} f(\overline{x}, y) = \big(\sum_{y=0}^{z} f(\overline{x}, y)\big) + f(\overline{x}, z+1).$$

For 2., 3., and 4. proceed analogously. If $f : \mathbb{N}^{k+1} \to \{0, 1\}$, then $\forall y \leq z\, f(\overline{x}, y) = \min\{f(\overline{x}, y) \mid 0 \leq y \leq z\}$ and $\exists y \leq z\, f(\overline{x}, y) = \max\{f(\overline{x}, y) \mid 0 \leq y \leq z\}$ which shows 5. and 6. For 7. define $f' : \mathbb{N}^{k+1} \to \mathbb{N}$ as

$$f'(\overline{x}, z) = \begin{cases} 1 & \text{if } f(\overline{x}, z) = 1 \text{ and } \forall z' < z\, f(\overline{x}, z') = 0 \\ 0 & \text{otherwise} \end{cases}$$

and observe that $(\mu y \leq z) f(\overline{x}, y) = \sum_{y=0}^{z} y \cdot f'(\overline{x}, y)$. □

We want to encode a pair of natural numbers as a single natural number. One option for doing that would be, e.g., to code (x, y) as $2^x 3^y$. However, we would like to i) avoid exponentation and ii) obtain a bijection. Therefore we wuse the mapping illustrated in the the following diagram:



This mapping from $\mathbb{N}^2$ to $\mathbb{N}$ is obviously bijective. Now we want to define it symbolically. To that aim, observe that pairs with the same sum are put on the same chain of arrows. More precisely, we have $(x_1, y_1) < (x_2, y_2)$ iff $x_1 + y_1 < x_2 + y_2$ or $(x_1 + y_1 = x_2 + y_2$ and $y_1 < y_2)$. There are $\sum_{i=1}^{x+y} i$ pairs with a sum less than $x + y$. Therefore we can define this bijection by

$$\langle x, y\rangle = (\sum_{i=1}^{x+y} i) + y = \frac{(x+y)(x+y+1)}{2} + y.$$

Observe that $x, y \leq \langle x, y\rangle$ and that, if $(x, y) \notin \{(0, 0), (1, 0)\}$, then $x < \langle x, y\rangle$ and $y < \langle x, y\rangle$. Another noteworthy feature of this pairing function is that it permits a definition in the usual language of arithmetical theories (which contains addition and multiplication but not exponentiation) as $z = \langle x, y\rangle$ iff $2z = (x + y)(x + y + 1) + 2y$. We define the inverses of the pairing function $\mathrm{l} : \mathbb{N} \to \mathbb{N}, \langle x, y\rangle \mapsto x$ and $\mathrm{r} : \mathbb{N} \to \mathbb{N}, \langle x, y\rangle \mapsto y$. Based on this pairing function we can now proceed to code tuples.

Definition 1.10. For $k \geq 3$ define $\langle \cdot, \ldots, \cdot\rangle : \mathbb{N}^k \to \mathbb{N}$ as $\langle x_1, \ldots, x_k\rangle = \langle x_1, \langle x_2, \ldots, x_k\rangle\rangle$. For $k = 1$ define $\langle \cdot\rangle : \mathbb{N} \to \mathbb{N}$ as the identity function.

For fixed $k \geq 1$, the function $\langle \cdot, \ldots, \cdot\rangle$ is bijective. The union over these functions is *not* bijective, consider, e.g., $0 = \langle 0, 0\rangle = \langle 0, 0, 0\rangle = \cdots$

Lemma 1.4. *The following functions are primitive recursive:*

1. *for $k \geq 1$:* $\langle \cdot, \ldots, \cdot\rangle : \mathbb{N}^k \to \mathbb{N}, (x_1, \ldots, x_k) \mapsto \langle x_1, \ldots, x_k\rangle$
2. $\pi : \mathbb{N}^3 \to \mathbb{N}, (k, i, x) \mapsto \begin{cases} x_i & \text{if } k \geq 1,\ 1 \leq i \leq k, \text{ and } x = \langle x_1, \ldots, x_k\rangle \\ 0 & \text{if } k = 0,\ i = 0, \text{ or } i > k \end{cases}$

Proof. We first show 1. If $k = 1$, then $\langle \cdot\rangle = \mathrm{P}_1^1$ is primitive recursive. If $k \geq 2$, observe that, for an even number z, $\frac{z}{2} = (\mu z_0 \leq z)\, 2 \cdot z_0 = z$. Therefore the pairing function $\langle \cdot, \cdot\rangle : \mathbb{N}^2 \to \mathbb{N}$ is primitive recursive. For $k \geq 3$, we obtain $\langle \cdot, \ldots, \cdot\rangle : \mathbb{N}^k \to \mathbb{N}$ by composing the pairing function with itself a suitable number of times.

For 2., note that

$$\begin{aligned} l(z) &= (\mu x \leq z)(\exists y \leq z)\, \langle x, y\rangle = z \text{ and} \\ r(z) &= (\mu y \leq z)(\exists x \leq z)\, \langle x, y\rangle = z. \end{aligned}$$

So both l and r are primitive recursive. Therefore, also $(j, z) \mapsto r^j(z)$ is primitive recursive. We have

$$\pi(k, i, x) = \begin{cases} r^{i-1}(x) & \text{if } k \geq 1 \text{ and } i = k \\ l(r^{i-1}(x)) & \text{if } k \geq 1 \text{ and } 1 \leq i < k \\ 0 & \text{otherwise} \end{cases}$$

and therefore also π is primitive recursive. □

Now that we have primitive recursive tuples we can show another useful closure property: the primitive recursive functions are closed under course-of-value recursion. To that aim define first:

Definition 1.11. Let $h : \mathbb{N}^{k+1} \to \mathbb{N}$. The *history function* $\hat{h} : \mathbb{N}^{k+1} \to \mathbb{N}$ of h is defined as $\hat{h}(\overline{x}, y) = \langle h(\overline{x}, y), \ldots, h(\overline{x}, 0)\rangle$.

Lemma 1.5. *If $f : \mathbb{N}^k \to \mathbb{N}$ and $g : \mathbb{N}^{k+2} \to \mathbb{N}$ are primitive recursive, then so is the function $h : \mathbb{N}^{k+1} \to \mathbb{N}$ defined by:*

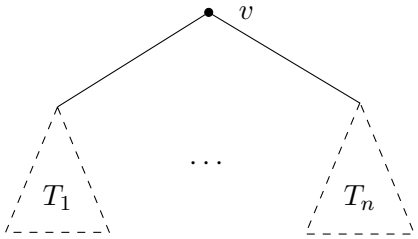
$$h(\overline{x}, 0) = f(\overline{x}) \text{ and}$$
$$h(\overline{x}, y+1) = g(\overline{x}, y, \hat{h}(\overline{x}, y)).$$

Proof. It suffices to show that $\hat{h}$ is primitive recursive because $h(\overline{x}, y) = \pi(y+1, 1, \hat{h}(\overline{x}, y))$. To this aim, note that

$$\hat{h}(\overline{x}, 0) = h(\overline{x}, 0) = f(\overline{x}) \text{ and}$$
$$\hat{h}(\overline{x}, y+1) = \langle h(\overline{x}, y+1), \hat{h}(\overline{x}, y)\rangle = \langle g(\overline{x}, y, \hat{h}(\overline{x}, y)), \hat{h}(\overline{x}, y)\rangle.$$

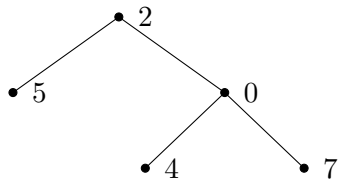
□

As a next step, we want to encode finite ordered trees, i.e., the order of subtrees is significant, whose vertices are labelled by natural numbers. Each such tree T will be encoded as a natural number $\#T$. We use our tuple encoding and define the code of a tree by induction on the structure of the tree: a tree of the form $T =$



is encoded as $\#T = \langle v, n, \#T_1, \ldots, \#T_n\rangle$ where $\#T_i$ is the code of the subtree T_i. Note that this definition includes the case $\langle v, 0\rangle$ for a leaf. Also note that # is just a function from trees to natural numbers with any a priori connection to primitive recursion or computability theory.

Example 1.4. The code of the ordered labelled tree



is $\langle 2, 2, \langle 5, 0\rangle, \langle 0, 2, \langle 4, 0\rangle, \langle 7, 0\rangle\rangle\rangle$ which is the 84-digit natural number

120443650830443822950654392810134061331537938301945868395455743743602923276498173998.

If the natural number $m = \langle v, n, m_1, \ldots, m_n\rangle$ is given, then $v = l(m)$ and $n = l(r(m))$. Therefore, v, n, m_1, …, m_n are determined uniquely by m. Furthermore, $n > 0$ implies that $m_i < m$ for $i = 1, \ldots, n$. So, by induction we can conclude that the mapping # from ordered labelled trees to $\mathbb{N}$ is bijective. Moreover, the functions $m \mapsto n$ and $(m, i) \mapsto m_i$ are primitive recursive.

Lemma 1.6 (Tree recursion)**.** *If $f : \mathbb{N} \to \mathbb{N}$ and $g : \mathbb{N}^4 \to \mathbb{N}$ are primitive recursive, then so is $h : \mathbb{N} \to \mathbb{N}$, defined by*

$$h(\langle v, n, x_1, \ldots, x_n\rangle) = \begin{cases} f(v) & \text{if } n = 0 \\ g(v, n, \langle x_1, \ldots, x_n\rangle, \langle h(x_1), \ldots, h(x_n)\rangle) & \text{if } n > 0 \end{cases}$$

Proof. First note that $v, n, x_1, \ldots, x_n$ are all well-defined since $\#$ from ordered trees to $\mathbb{N}$ is injective. Moreover, they can be computed by primitive recursive functions from $\langle v, n, x_1, \ldots, x_n \rangle$. Since $\#$ is also surjective, h is a total function. Furthermore, note that, for $n > 0$, $x_i < \langle v, n, x_1, \ldots, x_n \rangle$ for all $i \in \{1, \ldots, n\}$. Therefore $\langle h(x_1), \ldots, h(x_n) \rangle$ can be computed by a primitive recursive function, based on the projection π, from the value $\hat{h}(\langle v, n, x_1, \ldots, x_n \rangle)$ of the history function of h. So, by course-of-values recursion, h is primitive recursive. $\square$