Lessons touched by this meeting according to schedule:

* 10. 18/11/2024
  + Universal function: definition and computability [§5.1, Appendix of §5]
  + Computability of the inverse function, undecidability of the halting problem and of totality [§5.1]
* 11. 19/11/2024
  + Effective operations on computable functions. Exercises. [§5.3, §6.1.1, §6.1.3, §6.1.4, §6.1.6 with slightly different approach]

Immagine che contiene testo, calligrafia, Carattere, bianco

Descrizione generata automaticamenteConsider the universal function:

We want to prove we can create a "universal interpreter" that can:

1. Take any program (by its code number e)
2. Take its inputs (x̄)
3. Run that program on those inputs
4. Return whatever the original program would return

The proof works by showing we can:

1. Store program state (register contents)
2. Simulate program execution step by step
3. Track when the program finishes
4. Extract the final result

When you see (...)₁:

* This means "extract the contents of register 1"
* Register 1 is where programs store their output by convention
* Think of it as "get the return value"

Immagine che contiene calligrafia, Carattere, testo, tipografia

Descrizione generata automaticamenteExamples on how to use it in exercises:

Immagine che contiene testo, schermata, Carattere, documento

Descrizione generata automaticamenteLet’s focus already on the important part of this proof:

Immagine che contiene testo, schermata, Carattere, documento

Descrizione generata automaticamenteThis works for k values; then, we parametrize such search on bounded terms to look for tuples inside of functions.

Immagine che contiene testo, Carattere, schermata, ricevuta

Descrizione generata automaticamenteThe letter Chi (strange X) means “Characteristic function”, and it’s used to characterize predicates:

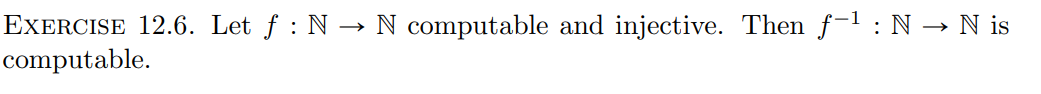


Immagine che contiene calligrafia, testo, Carattere, bianco

Descrizione generata automaticamenteFocus on this proof – if f is not total, computability is not guaranteed, so we need a way to minimize couples of numbers, so to encode them as an integer number:

Immagine che contiene testo, calligrafia, Carattere, inchiostro

Descrizione generata automaticamente

Now, let's talk about the projection functions w\_1 and w\_2. These functions are used to extract the first and second components of a pair, respectively. Formally:

w\_1(⟨x, y⟩) = x w\_2(⟨x, y⟩) = y

Immagine che contiene testo, Carattere, schermata, informazione

Descrizione generata automaticamenteIn other words, given the encoding of a pair ⟨x, y⟩, w\_1 returns the first element x, and w\_2 returns the second element y.

Immagine che contiene testo, calligrafia, Carattere, bianco

Descrizione generata automaticamenteAnother example to comment upon:

Immagine che contiene testo, calligrafia, Carattere, schermata

Descrizione generata automaticamente

Immagine che contiene testo, Carattere, schermata, linea

Descrizione generata automaticamenteLet’s jump to exercises:

We proceed by contradiction. Assume f is computable. Then ∃e. f = φe.

Let P(x) = "∀y≤x. φy(y)↓" be our condition. We can express P(x) formally using the halting predicate:

P(x) = ∏y≤x χH(y,y) where χH(y,y) = sg(μt. H(y,y,t))

Now consider f(e):

*Case 1*: If P(e) holds, then:

f(e) = φe(e) + 1 (by definition of f)

= f(e) + 1 (since we assumed f = φe)

This implies f(e) = f(e) + 1, which is a contradiction.

*Case 2*: If ¬P(e) holds, then:

f(e) = 0 (by definition of f)

φe(e) = f(e) = 0 (since we assumed f = φe)

Immagine che contiene testo, schermata, Carattere, documento

Descrizione generata automaticamenteBut this means φe(e)↓, contradicting ¬P(e) which requires some φy(y)↑ for y≤e.

Present to make everyone understand meaning and notations:

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

Link from some primitive recursive exercises: <https://proofwiki.org/wiki/Category:Primitive_Recursive_Functions>

Exercise: Define the class PR of primitive recursive functions and, using only the definition, prove that the function pmax : N^2 → N, defined by pmax(x,y) = max(2^x, 3^y), is primitive recursive.

Solution: The class PR of primitive recursive functions is the smallest class of functions that contains the basic functions:

1. Zero function: z(x) = 0 for each x ∈ N;

2. Successor function: s(x) = x + 1 for each x ∈ N;

3. Projection functions: U^k\_j(x\_1, ..., x\_k) = x\_j for each (x\_1, ..., x\_k) ∈ N^k and 1 ≤ j ≤ k.

and is closed under the following operations:

1. Composition: If f\_1, ..., f\_n : N^k → N and g : N^n → N are in PR, then the function h : N^k → N defined by h(x̄) = g(f\_1(x̄), ..., f\_n(x̄)) is also in PR.

2. Primitive Recursion: If f : N^k → N and g : N^(k+2) → N are in PR, then the function h : N^(k+1) → N defined by:

h(x̄, 0) = f(x̄)

h(x̄, y+1) = g(x̄, y, h(x̄, y))

To show that pmax(x,y) is in PR, we can build it up from simpler functions in PR:

1. The exponentiation functions exp\_2(x) = 2^x and exp\_3(y) = 3^y can be defined by primitive recursion:

exp\_2(0) = 1

exp\_2(x+1) = 2 · exp\_2(x)

exp\_3(0) = 1

exp\_3(y+1) = 3 · exp\_3(y)

2. The maximum function max(x,y) can also be defined by primitive recursion:

max(x,0) = x

max(x,y+1) = max(s(x), y)

3. Finally, pmax(x,y) can be defined by composition:

pmax(x,y) = max(exp\_2(x), exp\_3(y))

Since exp\_2, exp\_3, and max are all in PR, and PR is closed under composition, we conclude that pmax is also in PR.

