1. is **primitive recursion**:

Text

Description automatically generated

In our case, is .

When ,

which is equivalent to

Say there’s a projection function

(which is of course primitive recursive), then

QED.

1. This is a 2-parameter function. Let’s call it We will do induction on .

Base case:

Which is a projection function, which we know is primitive recursive.

Inductive case. Assuming is primitive recursive, prove is also primitive recursive:

Which equals

And we just proved that is primitive recursive. This is a composition schema. QED.

1. We define

Because if is the min, then , and ; similarly, if is the min, then , so it will also return correctly. is thus primitive recursive by composition. QED.

1. can be done via an algorithm similar to mergesort:

As one can see, this is recursive. The base case is projection of one variable onto the variable itself (which is primitive recursive), and the inductive case is primitive recursive by composition. QED.

1. We define

Because if is the max, then , and it returns ; similarly, if is the max, then , and , which is correct. is thus primitive recursive by composition of and .

1. Same as question 4, but just change to .
2. We define

Because only when does it return . All other cases will make it return . This function is thus primitive recursive by composition of of two other primitive recursive functions: a constant function and a projection function .

1. We define

Because , which is the case when ; and , as is the case when . is thus primitive recursive by composition of of two other primitive recursive functions: a constant function and , which we already proved to be primitive recursive. QED.

1. We define

Because at least one of and is . The bigger value will be the absolute difference between and . This function is thus primitive recursive because it is the composition schema by of two primitive recursive functions.

1. We define

Because when goes up by , so does the remainder, unless the remainder is , in which case increasing by makes it divisible by .

We now define again:

where is the indicator function, which we now will prove is also primitive recursive:

claim:

Is primitive recursive.

Proof:

We can define it as

Because if , then , and of that is ; similarly, when they aren’t equal, , which will return .

Thus, is primitive recursive because it is a composition of primitive recursive functions , , , , and . QED.

1. We define

Which can be redefined as

Which is primitive recursive via composition of addition, , , and via induction assumption.

1. I think is missing the base case , because does not exist in the natural numbers. I’m gonna assign . Therefore,

We just said satisfies the base case. In the inductive case:

Thus, is primitive recursive via and and .

1. Once again, I think is missing the base case , so we’re gonna make it . The function can now be defined as

We just did the base case. For the inductive case,

Thus, is primitive recursive via and and .

1. That means we can have ,,, but not . We could thus define

Because returns when the input is and outputs for everything greater than that.

1. That means we need both and to be , so we could define

Because

1. When returns , we need to return ; when returns , we need to return . We define
2. We define

This is primitive recursive because it is composition by , which is primitive recursive.

1. We define

Because . Which makes the function . And we proved is primitive recursive.

1. Proof of Skolem’s theorem

We will use strong induction on the length of the formula of bounded quantifier logic.

Base case: an atom is recursively defined to be a constant (constant function, which is primitive recursive) or a variable from the input variables (projection function, which is also primitive recursive), or an -ary function whose arguments are atoms (primitive recursive functions are closed under composition).

Inductive case: assuming all sub-functions of at length are primitive recursive, then is recursive. This is because is these smaller sub-functions joined by and , which we all proved in exercises 14 to 18 are primitive recursive.

QED.