Computability

Here is a clean way to define general recursive functions. Consider the 3x+1 function with natural number inputs.

```
f(x) = if x=0 then 1
       else if even(x) then f(x/2)
            else f(3x+1)
            fi
       fi
```

Using Lambda Notation

```
f = λ(x. if x=0 then 1
         else if even(x) then f(x/2)
              else f(3x+1))
```

Here is a related term with function input f

```
λ(f. λ(x. if x=0 then 1
          else if even(x) then f(x/2)
               else f(3x+1)))
```

The recursive function is computed using this term.

Defining Recursive Functions in CTT

```
fix(λ(f. λ(x. if x=0 then 1
              else if even(x) then f(x/2)
                   else f(3x+1)
                   fi
              fi)))
```

Recursion in General

f(x) = F(f,x) is a recursive definition, also
f = λ(x.F(f,x)) is another expression of it, and the CTT definition is:

fix(λ(f. λ(x. F(f,x)))

which reduces in one step to:

λ(x.F(fix(λ(f. λ(x. F(f,x)))),x))

by substituting the fix term for f in λ(x.F(f,x)) .

Recursion in General

f(x) = F(f,x) is a recursive definition, also
f = λ(x.F(f,x)) is another expression of it, and a simple definition is:

fix(λ(f. λ(x. F(f,x)))

which reduces in one step to:

λ(x.F(fix(λ(f. λ(x. F(f,x)))),x))

by substituting the fix term for f in λ(x.F(f,x)) .

Non-terminating Computations

CTT defines all general recursive functions, hence non-terminating ones such as this

fix(λ(x.x))

which in one reduction step reduces to itself!

This system of computation in the object language is a simple functional programming language.

Partial Functions

The concept of a partial function is an example of how challenging it is to include all computation in the object theory. It is also key to including unsolvability results with a minimum effort; the halting problem and related concepts are fundamentally about whether computations converge, and in type theory this is the essence of partiality. For example, we do not know that the 3x+1 function belongs to the type N -> N.

Partial Functions

We do however know that the 3x+1 function, call it f in this slide, is a partial function from numbers to numbers, thus for any n, f(n) is a number if it converges (halts).

In CTT we say that a value a belongs to the bar type Ā provided that it belongs to A if it converges. So f belongs to A → Ā for Ā = N.

Unsolvable Problems

It is remarkable that we can prove that there is no function in CTT that can solve the convergence problem for elements of basic bar types.

We will show this for non empty type Ā with element ā that converges in A for basic types such as Z, N, list(A), etc. We rely on the typing that if F maps Ā to Ā, then fix(F) is in Ā.

Unsolvable Problems

Suppose there is a function h that decides halting. Define the following element of Ā:

d = fix(λ(x. if h(x) then ↑ else ā fi))

where ↑ is a diverging element, say fix(λ(x.x)).

Now we ask for the value of h(d) and find a contradiction as follows:

Generalized Halting Problem

Suppose that h(d) = *t,* then d converges, but according to its definition, the result is the diverging computation ↑ because by computing the fix term for one step, we reduce

d = fix(λ(x. if h(x) then ↑ else ā fi))

to d = if h(d) then ↑ else ā fi .

If hd(d) = *f,* then we see that d converges to ā.