

Macros

Now we want to use macros of the form:

$W \leftarrow f(V_1, \dots, V_n)$

in our programs, where W ,

V_1, \dots, V_n can be any variables; W could be among V_1, \dots, V_n .

We expand the macro as follows:

$$\begin{aligned} Z_m &\leftarrow 0 \\ Z_{m+1} &\leftarrow V_1 \\ Z_{m+2} &\leftarrow V_2 \\ &\vdots \\ Z_{m+n} &\leftarrow V_n \\ Z_{m+n+1} &\leftarrow 0 \\ Z_{m+n+2} &\leftarrow 0 \\ &\vdots \\ Z_{m+n+k} &\leftarrow 0 \\ Q_m & \\ [E_m] \quad W &\leftarrow Z_m \end{aligned}$$

September 17, 2009

Theory of Computation
Lecture 4: Programs and Computable Functions III

1

Macros

Whenever we expand a macro, the number m has to be chosen **large enough** so that none of the variables or labels in Q_m occur in the main program.

Note that in the expansion the output and local variables are set to **zero**, although at the start of the main program they are set to zero anyway.

This is necessary because the macro expansion may be part of a **loop** in the main program.

September 17, 2009

Theory of Computation
Lecture 4: Programs and Computable Functions III

2

Macros

Obviously, if $f(V_1, \dots, V_n)$ is undefined (\uparrow), the program Q_m will never terminate.

So if f is not total, and the macro $W \leftarrow f(V_1, \dots, V_n)$ is encountered when V_1, \dots, V_n have values for which f is not defined, the main program will never terminate.

Example:

$Z \leftarrow X_1 - X_2$

$Y \leftarrow Z + X_3$

This program computes $f(x_1, x_2, x_3)$, where

$$\begin{aligned} f(x_1, x_2, x_3) &= (x_1 - x_2) + x_3, \text{ if } x_1 \geq x_2 \\ &= \uparrow, \text{ if } x_1 < x_2 \end{aligned}$$

September 17, 2009

Theory of Computation
Lecture 4: Programs and Computable Functions III

3

Macros

Now let us introduce macros of the form

IF $P(V_1, \dots, V_n)$ GOTO L ,

where $P(x_1, \dots, x_n)$ is a computable predicate.

This will be based on the convention that TRUE = 1, FALSE = 0.

According to this convention, predicates are just total functions whose values are always either 0 or 1.

September 17, 2009

Theory of Computation
Lecture 4: Programs and Computable Functions III

4

Macros

Let $P(x_1, \dots, x_n)$ be a computable predicate.

Then we expand the macro

IF $P(V_1, \dots, V_n)$ GOTO L

to

$Z \leftarrow P(V_1, \dots, V_n)$

IF $Z \neq 0$ GOTO L

As usual, the variable Z has to be chosen to create no conflicts with the main program.

September 17, 2009

Theory of Computation
Lecture 4: Programs and Computable Functions III

5

Macros

Example:

How can we expand the macro

IF $V=0$ GOTO L ?

$V = 0$ corresponds to the following predicate $P(x)$:

$P(x) = \text{TRUE}$, if $x = 0$

= FALSE, otherwise

This can be computed by the following program:

IF $X \neq 0$ GOTO E

$Y \leftarrow Y+1$

September 17, 2009

Theory of Computation
Lecture 4: Programs and Computable Functions III

6

(Partially) Computable Functions

By introducing macros, we have seen that it is possible to compute complex functions with our very simple programming language \mathcal{L} .

Notice that macros do not change the specification of the language, but they just simplify writing down programs.

As you know, we can always replace macros with actual code.

So what are the limitations of the language \mathcal{L} ?

In order to find out, we need to do some mathematics.

September 17, 2009

Theory of Computation
Lecture 4: Programs and Computable Functions III

7

Composition

Let us **combine** computable functions in such a way that the output of one becomes an input to another.

For example, we could combine the functions f and g to obtain a new function h :

$$h(x) = f(g(x))$$

Let us now take a more general view:

Definition: Let f be a function of k variables and let g_1, \dots, g_k be functions of n variables. Let

$$h(x_1, \dots, x_n) = f(g_1(x_1, \dots, x_n), \dots, g_k(x_1, \dots, x_n)).$$

Then h is said to be obtained from f and g_1, \dots, g_k by **composition**.

September 17, 2009

Theory of Computation
Lecture 4: Programs and Computable Functions III

8

Composition

Theorem 1.1: If h is obtained from the (partially) computable functions f, g_1, \dots, g_k by composition, then h is (partially) computable.

Proof: The following program obviously computes h :

$$Z_1 \leftarrow g_1(X_1, \dots, X_n)$$

:

$$Z_k \leftarrow g_k(X_1, \dots, X_n)$$

$$Y \leftarrow f(Z_1, \dots, Z_k)$$

If f, g_1, \dots, g_k are not only partially computable but are also total, then so is h . ■

September 17, 2009

Theory of Computation
Lecture 4: Programs and Computable Functions III

9

Composition

Example:

We know that $f(x_1, x_2) = x_1 + x_2$ is a computable function.

We also know that $g_1(x) = x^2$ and $g_2(x) = 3x$ are computable functions.

According to **Theorem 1.1**, the following function $h(x)$ must then also be computable:

$$h(x) = f(g_1(x), g_2(x)) = f(x^2, 3x) = x^2 + 3x$$

September 17, 2009

Theory of Computation
Lecture 4: Programs and
Computable Functions III

10

Recursion

Let k be some fixed number and

$$h(0) = k$$

$$h(t + 1) = g(t, h(t)),$$

where g is some given total function of two variables.

Then we say that h is obtained from g by **primitive recursion**, or simply **recursion**.

Theorem 2.1: Let h be obtained as shown above, and let g be computable. Then h is also computable.

September 17, 2009

Theory of Computation
Lecture 4: Programs and
Computable Functions III

11

Recursion

Proof:

Obviously, the function $f(x) = k$ is computable.

The program computing $f(x)$ simply consists of k times the instruction $Y \leftarrow Y + 1$.

This gives us the macro $Y \leftarrow k$.

Now we can write a program that computes $h(x)$:

This is your homework question 1!

September 17, 2009

Theory of Computation
Lecture 4: Programs and
Computable Functions III

12

More Homework Questions

Question 2: Let \mathcal{P} be the following program:

```
IF X ≠ 0 GOTO A
Z ← Z + 1
IF Z ≠ 0 GOTO B
[A] X ← X - 1
    Y ← Y + 1
    IF X ≠ 0 GOTO A
[B] Y ← Y + 1
    Y ← Y + 1
```

What is the function $f(x)$ computed by \mathcal{P} ?

Question 3: Write a program in \mathcal{L} that computes $f(x) = 2x$ without using any macros. After the program terminates, the variable X must contain its initial value (the input).