

## 1.8 Recursion with Measure

**1.8.1 Introduction.** For efficient computation computer programming requires definitions of functions with almost arbitrary recursion. Since we do not wish such extensions to be inconsistent we restrict ourselves to *regular* recursive definitions. The condition of regularity for the recursive definition of a form  $f(\vec{x}) = \tau[f; \vec{x}]$  means that there must be a *measure*  $\mu[\vec{x}]$  in which the recursion goes down; i.e. we have  $\mu[\vec{\rho}] < \mu[\vec{x}]$  for every recursive application  $f(\vec{\rho})$  in  $\tau$ . Regular recursive definitions are discussed in the next section. In this section we consider a slightly restrictive form of regular recursion.

Let  $\tau[f; \vec{x}]$  be a term with all free variables indicated, and let  $\mu[\vec{x}]$  be a measure. We call

$$f(\vec{x}) = \tau[\dot{\lambda}\vec{y}.D(\mu[\vec{y}] <_* \mu[\vec{x}], f(\vec{y}), 0); \vec{x}] \quad (1)$$

the definition by (*course of values*) *recursion with measure*  $\mu$ . Note that every recursive application in (1) is surrounded by a *guard* guaranteeing the decrease of recursive arguments in the measure  $\mu$ . This means that every recursive application  $f(\vec{\rho})$  in  $\tau$  is replaced by the term

$$D(\mu[\vec{\rho}] <_* \mu[\vec{x}], f(\vec{\rho}), 0).$$

In the sequel we will use the notation  $\tau[[f]_{\vec{x}}^{\mu}; \vec{x}]$  (or even  $\tau[[f]; \vec{x}]$ ) as an abbreviation for the term on the right-hand side of the identity (1).

**1.8.2 The principle of measure induction.** For every formula  $\varphi[\vec{x}]$  and term  $\mu[\vec{x}]$ , the formula of *induction on  $\vec{x}$  with measure  $\mu[\vec{x}]$  for  $\varphi$*  is the following one:

$$\forall \vec{x} (\forall \vec{y} (\mu[\vec{y}] < \mu[\vec{x}] \rightarrow \varphi[\vec{y}]) \rightarrow \varphi[\vec{x}]) \rightarrow \forall \vec{x} \varphi[\vec{x}]. \quad (1)$$

We assume here that the variables  $\vec{y}$  are different from  $\vec{x}$  and that they do not occur in  $\varphi$ . The formula  $\varphi$  and the term  $\mu$  may contain additional variables as parameters.

Note that for  $\vec{x} \equiv x$  and  $\mu[x] \equiv x$ , the schema of measure induction is just the schema of complete induction.

**1.8.3 Theorem** *The principle of measure induction holds for each formula.*

*Proof.* The principle of measure induction 1.8.2(1) is reduced to mathematical induction as follows. Under the assumption that  $\varphi$  is  $\mu$ -*progressive*:

$$\forall \vec{x} (\forall \vec{y} (\mu[\vec{y}] < \mu[\vec{x}] \rightarrow \varphi[\vec{y}]) \rightarrow \varphi[\vec{x}]), \quad (\dagger_1)$$

we first prove, by induction on  $n$ , the following auxiliary property

$$\forall \bar{z}(\mu[\bar{z}] < n \rightarrow \varphi[\bar{z}]). \quad (\dagger_2)$$

In the base case there is nothing to prove. In the induction step take any  $\bar{z}$  such that  $\mu[\bar{z}] < n+1$  and consider two cases. If  $\mu[\bar{z}] < n$  then we obtain  $\varphi[\bar{z}]$  by IH. If  $\mu[\bar{z}] = n$  then by instantiating of  $(\dagger_1)$  with  $\bar{x} := \bar{z}$  we obtain

$$\forall \bar{y}(\mu[\bar{y}] < n \rightarrow \varphi[\bar{y}]) \rightarrow \varphi[\bar{z}].$$

Now we apply IH to get  $\varphi[\bar{z}]$ .

With the auxiliary property proved we obtain that  $\varphi[\bar{x}]$  holds for every  $\bar{x}$  by instantiating of  $\forall n(\dagger_2)$  with  $n := \mu[\bar{x}] + 1$  and  $\bar{z} := \bar{x}$ .  $\square$

**1.8.4 Recursive definitions with measure.** Let  $\mu[\bar{x}]$  and  $\tau[f; \bar{x}]$  be terms in which no other variables than the  $n$  indicated ones are free. Consider the following definition of an  $n$ -ary function  $f$ :

$$f(\bar{x}) = \tau[[f]_{\bar{x}}^{\mu}; \bar{x}], \quad (1)$$

We say that  $f$  is defined *by recursion with measure*. Any such definition can be viewed as a function operator taking all functions in the terms  $\tau$  and  $\mu$  and yielding the function  $f$  as a result.

We keep the notation introduced in this paragraph fixed until Thm. 1.8.7, where we prove that the class of primitive recursive functions is closed under the operator of recursion with measure.

**1.8.5 Approximation function.** We wish to introduce the function  $f$  as a p.r. function with the help of its *approximation* function  $f^+(z, \bar{x})$ . The additional argument  $z$  plays the role of the depth of recursion counter. It estimates the depth of recursion needed to compute the value  $f(\bar{x})$ . If  $z$  is sufficiently large then we have  $f(\bar{x}) = f^+(z, \bar{x})$ . As we will see below every number  $z > \mu[\bar{x}]$  gives us sufficient estimation of the depth of recursion. This will allow us to defined  $f$  explicitly by  $f(\bar{x}) = f^+(\mu[\bar{x}] + 1, \bar{x})$ .

The approximation function is introduced with the help of *approximation* terms  $\rho^+[f^+; z, \bar{x}]$  which are defined for all subterms  $\rho$  of  $\tau$  to satisfy:

$$\begin{aligned} x_i^+ &\equiv x_i && \text{(variable)} \\ g(\rho_1, \dots, \rho_k)^+ &\equiv g(\rho_1^+, \dots, \rho_k^+) && \text{(auxiliary function)} \\ f(\rho_1, \dots, \rho_n)^+ &\equiv f^+(z, \rho_1^+, \dots, \rho_n^+). && \text{(recursive application)} \end{aligned}$$

We define  $f^+$  by the following nested simple recursion:

$$f^+(0, \bar{x}) = 0 \quad (1)$$

$$f^+(z+1, \bar{x}) = \tau^+[\lambda z_1 \bar{y}. D(\mu[\bar{y}] <_* \mu[\bar{x}], f^+(z, \bar{y}), 0); z, \bar{x}]. \quad (2)$$

The approximation function is primitive recursive by Thm. 1.7.15.

Below we will use the notation  $\tau^+[[f^+]_{z,\vec{x}}^\mu; z, \vec{x}]$  (or even  $\tau^+[[f^+]; z, \vec{x}]$ ) as an abbreviation for the term on the right-hand side of the equation (2). We will also use the notation  $(\rho_1, \dots, \rho_m)^+$  as an abbreviation for  $(\rho_1^+, \dots, \rho_m^+)$ .

**1.8.6 Monotonicity of the approximation function.** We have

$$\mu[\vec{x}] < z_1 \leq z_2 \rightarrow f^+(z_1, \vec{x}) = f^+(z_2, \vec{x}). \quad (1)$$

The property asserts that the application  $f^+(z, \vec{x})$  yields the same result for all numbers  $z > \mu[\vec{x}]$ .

*Proof.* The property is proved by induction on  $z_2$  as  $\forall \vec{x} \forall z_1 (1)$ .  $\square$

**1.8.7 Theorem** *Primitive recursive functions are closed under recursion with measure.*

*Proof.* Let  $f$  be defined by recursion with measure as in Par. 1.8.4 from p.r. functions. Let further  $f^+$  be its approximation function as in Par. 1.8.5. We claim that we have

$$f(\vec{x}) = f^+(\mu[\vec{x}] + 1, \vec{x}). \quad (\dagger_1)$$

The function  $f^+$  is primitive recursive and so is  $f$ .

The property is proved by measure induction on  $\vec{x}$  with measure  $\mu[\vec{x}]$ . So take any  $\vec{x}$  and prove by (the inner) induction on the structure of subterms  $\rho[f; \vec{x}]$  of  $\tau$  the property

$$\rho[[f]; \vec{x}] = \rho^+[[f^+]; \mu[\vec{x}], \vec{x}]. \quad (\dagger_2)$$

We continue by case analysis of  $\rho$ . The case when  $\rho \equiv f(\vec{\theta})$  follows from

$$\begin{aligned} & D\left(\mu[\vec{\theta}[[f]; \vec{x}]] <_* \mu[\vec{x}], f(\vec{\theta}[[f]; \vec{x}]), 0\right) \stackrel{\text{outer IH}}{=} \\ & = D\left(\mu[\vec{\theta}[[f]; \vec{x}]] <_* \mu[\vec{x}], f^+(\mu[\vec{\theta}[[f]; \vec{x}]] + 1, \vec{\theta}[[f]; \vec{x}]), 0\right) \stackrel{1.8.6(1)}{=} \\ & = D\left(\mu[\vec{\theta}[[f]; \vec{x}]] <_* \mu[\vec{x}], f^+(\mu[\vec{x}], \vec{\theta}[[f]; \vec{x}]), 0\right) \stackrel{\text{inner IH's}}{=} \\ & = D\left(\mu[\vec{\theta}^+[[f^+]; \mu[\vec{x}], \vec{x}]] <_* \mu[\vec{x}], f^+(\mu[\vec{x}], \vec{\theta}^+[[f^+]; \mu[\vec{x}], \vec{x}]), 0\right). \end{aligned}$$

The remaining cases when  $\rho \equiv x_i$  or  $\rho \equiv g(\vec{\theta})$  are straightforward. With the auxiliary property proved the equality  $(\dagger_1)$  is obtained from

$$f(\vec{x}) \stackrel{1.8.4(1)}{=} \tau[[f]; \vec{x}] \stackrel{(\dagger_2)}{=} \tau^+[[f^+]; \mu[\vec{x}], \vec{x}] \stackrel{1.8.5(2)}{=} f^+(\mu[\vec{x}] + 1, \vec{x}). \quad \square$$