

## Comp 411 Notes - Wed, Mar 30 2005

Recursive Types

Scribe: Seth James Nielson

**NOTE:** We have ammended the schedule to reflect our somewhat slower than anticipated pace. The homework that was due Monday has been pushed back. However, the following problems have been in their place 20.1.1, 20.1.2, 20.1.3, 20.1.4, 20.1.5. These are due Monday April 4th, 2005.

In today's lecture, we investigated subtyping a recursive function type.

### 1 Recursive Function Types

Suppose we want to compose some "black boxes". Before we can compose them, we have to be able to "pass" them around. This means that we need to type them.

$$\begin{array}{l} \square \longrightarrow \mathbf{O} \quad \mu\alpha.() \rightarrow \alpha \times 0 \\ | \longrightarrow \square \longrightarrow \mathbf{O} \quad \mu\alpha.I \rightarrow \alpha \times 0 \\ | \longrightarrow \square \quad \mu\alpha.I \rightarrow \alpha \end{array}$$

We can convert  $\mu\alpha.I \rightarrow \alpha \times 0$  into something that terminates. The new version looks like  $\mu\alpha.I \rightarrow (\alpha \times 0 + 1)$ . We have converted a type that is strictly infinite into something that is arbitrary in length.

It should be noted that in this particular case, the term that is arbitrary in length is also *possibly infinite*. This is because the type is a  $\rightarrow$  type

(function) and delays the evaluation of the right side. So, the type can be either infinite or arbitrary.

## 2 Subtyping

How do we subtype recursive types?

We keep the default subtyping rules:

$$\frac{}{t <: t} \quad \frac{t_1 <: t_2 \quad t_2 <: t_3}{t_1 <: t_3}$$

Suggested judgement:

$$\frac{t_1[\alpha := \mu\alpha.t_1] <: t_2[\beta := \mu\beta.t_2]}{\mu\alpha.t_1 <: \mu\beta.t_2}$$

Do these rules work? Well, one problem is that given the current rule, the antecedent *grows* rather than reduces. This means that it is not clear that we can terminate. Rather than reducing the problem to something easier to prove, it has become something harder to prove.

We'll modify our judgements to include a  $\Gamma$  to represent the recursive context.

$$\frac{}{\Gamma \vdash t <: t} \quad \frac{\Gamma \vdash t_1 <: t_2 \quad \Gamma \vdash t_2 <: t_3}{\Gamma \vdash t_1 <: t_3}$$

Why is the  $\Gamma$  the same in the antecedant of the second judgement? (we come back to this in a minute)

What do we add to the third judgement? When do we add to  $\Gamma$ .

$$\frac{\Gamma, \alpha <: \beta \vdash t_1 <: t_2}{\Gamma \vdash \mu\alpha.t_1 <: \mu\beta.t_2}$$

We also have to add:

$$\overline{\Gamma, \alpha <: \beta \vdash \alpha <: \beta}$$

Once again, in the judgement

$$\frac{\Gamma \vdash t_1 <: t_2 \quad \Gamma \vdash t_2 <: t_3}{\Gamma \vdash t_1 <: t_3}$$

Why are the  $\Gamma$ 's in the antecedent the same? BECAUSE, there is not any information in the judgement to alter the  $\Gamma$ . If we were traversing up the tree, we could not deterministically alter the two  $\Gamma$ 's in the antecedent.

Assuming that  $\text{Even} <: \text{Nat}$ , can we relate the following two terms:

$$\mu\alpha.\text{Nat} \rightarrow (\text{Even} \times \alpha) \quad \mu\beta.\text{Even} \rightarrow (\text{Nat} \times \beta)$$

Which is a subtype of the other?

Let's look at the subtyping rule for functions.

$$\frac{t_3 <: t_1 \quad t_2 <: t_4}{t_1 \rightarrow t_2 <: t_3 \rightarrow t_4}$$

We see that  $\text{Even} <: \text{Nat}$ , but what do we do about  $(\text{Even} \times \alpha)$  and  $(\text{Nat} \times \beta)$ ? Well, what our fourth subtyping rule tells us is that when trying to determine if  $\alpha <: \beta$ , assume that  $\alpha <: \beta$  inside of the recursive term. So, we can determine that  $M_a <: M_b$ .

We can also see that not  $M_a <: M_b$ .

To show this more clearly, we label the different subtypable components.

$$\mu\alpha. \quad \underset{1}{\text{Nat}} \quad \rightarrow \quad \overset{M_a}{(\text{Even} \times \alpha)} \quad \left| \quad \mu\beta. \quad \underset{4}{\text{Even}} \quad \rightarrow \quad \overset{M_b}{(\text{Nat} \times \beta)} \quad \underset{6}{\beta}$$

According to the rule for subtyping functions, all of the following must be true.

4 <: 1	Even <: Nat	TRUE
2 <: 5	Even <: Nat	TRUE
3 <: 6	$\alpha <: \beta$	TRUE

So, indeed  $M_a <: M_b$ . We can also show  $M_b \not<: M_a$  through a simple contradiction:

4  $\not<:$  1 as Nat  $\not<:$  Even

Now, what about

$M_a$					$M_b$				
$\mu\alpha$	Nat	$\times \alpha \rightarrow$	(Even		$\mu\beta$	Even	$\times \beta \rightarrow$	(Nat	$\times \beta)$
1	2	3	4		5	6	7	8	

Again looking at each subcomponent we see:

5 <: 1	Even <: Nat	TRUE
6 <: 2	$\beta \not<: \alpha$	FALSE!
3 <: 7	Even <: Nat	TRUE
4 <: 8	$\alpha <: \beta$	TRUE

So, in this case,  $M_a \not<: M_b$  (and, as a side note,  $M_b \not<: M_a$ ).