17·4·2025

Rule of substitution that we shall follow: [(b->b)/a] is ok but [(a->a)/a] not allowed
That is for a substitution [σ/*a*] there should **not be any occurrence** of *a* in σ

Definition: Type substitution: a type substitution ***s*** is any expression

$$[\sigma_1 / a_1 \; .... \; \sigma_n / a_n]$$

Where σ_i are types and a_i 's are distinct type variables.
For any τ define ***s***(τ) to be the type obtained by simultaneously substituting σ_1 for a_1 ... σ_n for a_n throughout τ. We call ***s***(τ) a substitution instance of τ.
When n=0, called empty substitution e(τ) = τ, when n=1, s will be called a single substitution.

The set $\{a_1 .. a_n\}$ will be called Dom(s)—variable domain of s
Vars(σ_1 .. σ_n) will be called the Range(s)—the variable range of s

Definition: composition of two substitutions
If s and t are any substitutions, say,

$$s \equiv [\sigma_1 / a_1 \; .... \; \sigma_n / a_n] \qquad t \equiv [\tau_1 / b_1 \; .... \; \tau_p / b_p] \qquad \text{define}$$

$$s \bullet t \equiv [\sigma_{\{i_1\}} / a_{\{i_1\}} \; .... \; \sigma_{\{i_h\}} / a_{\{i_h\}} \; , \; s(\tau_1) / b_1 \; .... \; s(\tau_p) / b_p]$$

where $\{a_{\{i_1\}} \; .. \; a_{\{i_h\}} \}$ = Dom(s) – Dom(t) and h = 0 ..n

Lemma : (i) Dom(s • t) = Dom(s) ∪ Dom(t)
(ii) (s • t)(τ) ≡ s(t(τ))
(iii) r • (s•t) = (r•s)• t associative

Example: let t = [e/c, e/b] s = [a/e] Dom(s) = {e}
Then s•t ≡ [a / e, s(e) / c, s(e) / b] ≡ [a / e, a / c, a / b]

Definition:

s(<τ_1 .. τ_n>) = <s(τ_1) .. s(τ_n)>

s(Γ) = {x_1 : s(τ_1) ... x_m : s(τ_m)}

s(Γ |-> M : τ) = s(Γ) |-> M : s(τ)

Definition: most general unifier (mgu) of <ρ,τ> is a unifier **u** such that for every other unifier s of <ρ,τ>, we have s(ρ) ≡ s' (**u**(ρ)) for some s'. if v ≡ **u**(ρ) for mgu **u** <ρ,τ> we shall call v a most general unification of <ρ,τ>.

Example: to prove that u is mgu.
Let ρ = c -> e τ = b -> c let s = [a/c, a/e, a/b] suppose u = [e/c, e/b] let s' = [a/e]
Verify that s ≡ s' • u

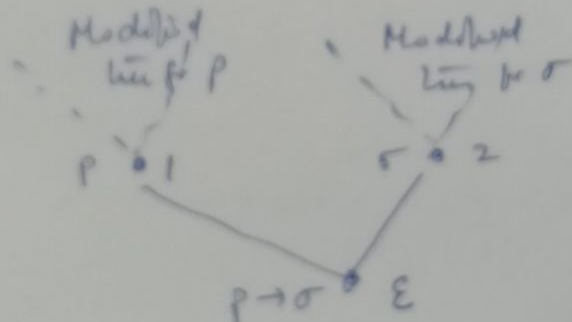
Construction-tree of a type

(i) τ is an atom: $\tau \equiv e$

tree has one node $e \bullet \varepsilon$

(ii) $\tau \equiv \rho \to \sigma$

its tree is built from the trees for ρ and σ by first putting "1" on the left-end of each position label in the tree for ρ, and next putting "2" on the left end of each position-label in the tree for σ, and then placing an extra node beneath the two modified trees:



eg. $\tau \equiv \sigma \to \sigma' \to \sigma''$

$\equiv \sigma \to (\sigma' \to \sigma'')$ let $\beta \equiv \sigma' \to \sigma''$