Math Logic: Model Theory & Computability
Lecture 05

Example. In $\sigma_{mon} := (1, \cdot)$, let $\underline{A} := (\mathbb{Z}, 1, \cdot)$ and $\underline{B} := (\mathbb{Z}, \overset{0}{\underset{\shortparallel}{1}}, \overset{+}{\underset{\shortparallel}{\cdot}})$.
Then $t := (v_1 \cdot 1) \cdot v_3$ is a σ_{mon}-term, and $t(v_1, v_3)$ and $t(v_1, v_3, v_2)$ are both extended terms. $t^{\underline{A}}(v_1, v_3) : \mathbb{Z}^2 \to \mathbb{Z}$ is given by $(n_1, n_2) \mapsto (n_1 \cdot 1) \cdot n_2$ while $t^{\underline{B}}(v_1, v_3) : \mathbb{Z}^2 \to \mathbb{Z}$ taking $(n_1, n_2) \mapsto (n_1 + 0) + n_2$.
On the other hand, $t^{\underline{A}}(v_1, v_3, v_2) : \mathbb{Z}^3 \to \mathbb{Z}$ given by $(n_1, n_2, n_3) \mapsto (n_1 \cdot 1) \cdot n_2$ and $t^{\underline{B}}(v_1, v_3, v_2) : \mathbb{Z}^3 \to \mathbb{Z}$ is given by $(n_1, n_2, n_3) \mapsto (n_1 + 0) + n_2$.

We call the variables present in $\vec{v}$ but not used in t *dummy variables*. Note that to denote the value of the function $t^{\underline{A}}(v_1, v_3, v_2)$ at (n_1, n_2, n_3), we would have to write $t^{\underline{A}}(v_1, v_3, v_2)(n_1, n_2, n_3)$, which is too cumbersome, so when there is no chance of confusion, we'd simply write $t^{\underline{A}}(n_1, n_2, n_3)$.

Just like homomorphisms preserve the constants and operations, they also also preserve interpretations of terms:

Prop. Let $\underline{A} := (A, \sigma)$ and $\underline{B} := (B, \sigma)$ be σ-structures and $h : \underline{A} \to \underline{B}$ be a homomorphism. Then for each extended σ-term $t(\vec{v})$ with $n := |\vec{v}|$, we have that
$$h \circ t^{\underline{A}}(\vec{v}) = t^{\underline{B}}(\vec{v}) \circ h,$$
more precisely, $h(t^{\underline{A}}(\vec{a})) = t^{\underline{B}}(h(\vec{a}))$ for all $\vec{a} \in A^n$.

Proof. We prove this by induction on the length/construction of t.

__Case 1:__ $t = c$, where $c \in \text{Const}(\sigma)$. Then $h(t^{A}(\vec{a})) = h(c^{\underline{A}}) = c^{\underline{B}} = t^{\underline{B}}(h(\vec{a}))$.

__Case 2:__ $t = v_i$, where v_i is a variable. Then because $t(\vec{v})$ is an ext. term,

this v_i must appear in $\vec{v}$, say as the j^{th} variable, so
$h(t^{\mathcal{A}}(\vec{a})) = h(a_j) = (h(\vec{a}))_j = t^{\mathcal{B}}(h(\vec{a}))$.

<u>Case 3:</u> $t = f(t_1, \dots, t_k)$, where $f \in \mathrm{Funct}_k(\sigma)$ and $t_1, \dots, t_k$ are σ-terms. Then because $t(\vec{v})$ is an ext term, so are $t_1(\vec{v}), \dots, t_k(\vec{v})$. Hence by induction, $h(t_i^{\mathcal{A}}(\vec{a})) = t_i^{\mathcal{B}}(h(\vec{a}))$ for each $i \in \{1, \dots, k\}$. Moreover, because h is a hom, h commutes with the interpretation of f, so $h(t^{\mathcal{A}}(\vec{a})) = h(f^{\mathcal{A}}(t_1^{\mathcal{A}}(\vec{a}), \dots, t_k^{\mathcal{A}}(\vec{a}))) = f^{\mathcal{B}}(h(t_1^{\mathcal{A}}(\vec{a})), \dots, h(t_k^{\mathcal{A}}(\vec{a})))$
$= f^{\mathcal{B}}(t_1^{\mathcal{B}}(h(\vec{a})), \dots, t_k^{\mathcal{B}}(h(\vec{a}))) = t^{\mathcal{B}}(h(\vec{a}))$. $\square$

Terms are names for new functions obtained by composition from constants and variables. We also need names for relations (= true/false valued functions) obtained from the equality and relation symbols in σ by logical connectives and quantifiers.

<u>Def.</u> A σ-formula is a word φ in the alphabet A_σ defined inductively as follows:

(i) $\varphi := t_1 = t_2$, where t_1, t_2 are σ-terms.

(ii) $\varphi := R(t_1, t_2, \dots, t_k)$, where $R \in \mathrm{Rel}_k(\sigma)$ and $t_1, \dots, t_k$ are σ-terms.

(iii) $(\psi \vee \eta)$, where ψ, η are σ-formulas.

(iv) $(\neg \psi)$, where ψ is a σ-formula.

(v) $(\exists v_i \psi)$, where ψ is a σ-formula and v_i is a variable.

<u>Examples.</u> (a) $\sigma_{\text{arithm}} := (0, S, +, \cdot)$, where S is a unary function symbol and the other symbols are as expected. Then the following are formulas: $S(S(0)) = v_0$, $\exists v_4 \big(((v_1 \cdot v_0) + S(0) = v_3) \vee (S(0) = 0) \big)$.

(b) In the same signature σ_{arithm}, let's abbreviate by $\dot{k} := \underbrace{S(S(\dots S}_{k}(0))))$.

Also $v_i^k := (\dots(\underbrace{v_i \cdot v_i) \cdot \dots \cdot v_i}_{k})$. Also, let $\psi \wedge \eta := \neg((\neg\psi) \vee (\neg\eta))$

$\psi \to \eta := (\neg\psi) \vee \eta$

Then the following are σ_{arith}-formulas: $\forall v_i \psi := \neg(\exists v_i (\neg\psi))$

- $\leq := (\exists z\, (x + z = y))$
- $div := (\exists z\, (x \cdot z = y))$
- $prime := (\forall x\, (div \to ((x = 1) \vee (x = y))))$

[we use other letter like x, y, z, u, v, w for variable with the understanding that these should all be replaced by $v_0, v_1, v_2, \dots$]

We now need to define extended σ-formulas, but unlike with terms, it's not as simple as just requiring the vector $\vec{v}$ to include all variables present in a given σ-formula hence some variable may appear under quantifiers. For example $\forall x\, (x = x)$ is a σ-formula and intuitively, it is always true and to compute this we don't need to give x a value from "outside". Even worse, in the formula

$$(\forall x (x = y)) \vee (\neg (x = z)).$$

We need to define a notion of a free variable...