Intensional Type Logic (Montague)

Goal: Semantics for natural language (sentences) — "NL as a formal language"

- Starting point: "Disambiguated" language of propositional logic $\mathcal{A}^0$ — fully bracketed version of $\mathcal{L}$ with one-place predicates $'P(a)'$, where a is an individual constant; no variables, no quantifiers.
- Standard semantic theory for $\mathcal{A}^0$ truth functions: truth values$^n \longrightarrow$ truth values
- $\langle E, f \rangle$ is a model of $\mathcal{A}^0$
  $E \neq \emptyset, f$: assignment
  $f('a') \in E \longrightarrow\ 'a'$ (basic expressions of $\mathcal{A}^0$)
  $f('P') \subset E \longrightarrow\ 'P'$
  $f('\neg') \subset E \longrightarrow\ '\neg'$ etc. $('\wedge','\vee')$

The *model structure* E provides *possible denotations* that could be given to expressions of any language.

Model assignment assigns certain of these possible denotations to the expressions of some language.

- Altering the denotations of one-place predicates:
  Instead of assigning to $'P'$ subsets of E we assign them functions:
  $E \to \{\top, \bot\}$
  i.e. we identify a subset $X \subset E$ with its characteristic function.
  Reason: easier to generalize.

- Possible denotations relative to a model set E:
  Introduction of **types**.

  Basic types: e "entities" (possible individuals)
  t truth values
  The set T^0 of classical types is the smallest set s.t. $e \in T^0, t \in T^0$, and whenever $\sigma, \tau \in T^0$ then the pair $\langle \sigma, \tau \rangle \in T^0$.

$\langle \sigma, \tau \rangle$: the type of functions from things of type σ to things of type τ

Examples for types: $e, t, \langle e, t \rangle$, $\langle t, t \rangle$ truth function of $'\neg'$,
$\langle t \langle t, t \rangle \rangle$ truth functions of $'\wedge','\vee'$

- The set of possible denotations of type τ relative to a model structure E is defined recursively:
  E is the set of possible denotations of type e.
  $\{\top, \bot\}$ is the set of possible denotations of type t.
  Whenever X is a set of possible denotations of type σ,
  and Y is a set of possible denotations of type τ,
  Y^X (the set of functions of X to Y) is the set of possible denotations of type σ, τ.

- Strictly speaking: A model of a language assigns values only to basic expressions of the the language.
  Assignment of values to composite expressions by means of semantic rules (one for each syntactic rule)
  Extend f to f' which is defined for all expressions of $\mathcal{A}^0$:

– Basic expressions ζ: $f'(\zeta) = f(\zeta)$
– One-place predicate P, individual constant a:
$f'(F_0^0(P, a)) = (f'(P))(f'(a))$
– Formula ϕ, one-place connective ζ: $f'(F_0^1(\zeta, \phi)) = (f'(\zeta))(f'(\phi))$
– Formulas ϕ, ψ, two-place connective ζ:
$f'(F_0^2(\zeta, \phi, \psi)) = (f'(\zeta))(f'(\phi), f'(\psi))$