Combinators

Words, language, theory

Remember that for a language L over an alphabet Σ one has $L \subseteq \Sigma^*$

Σ^* consists of all strings, possibly nonsensical

$L \subseteq \Sigma^*$ chooses in some way *meaningful* strings called *sentences*

often such a language is given by a grammar

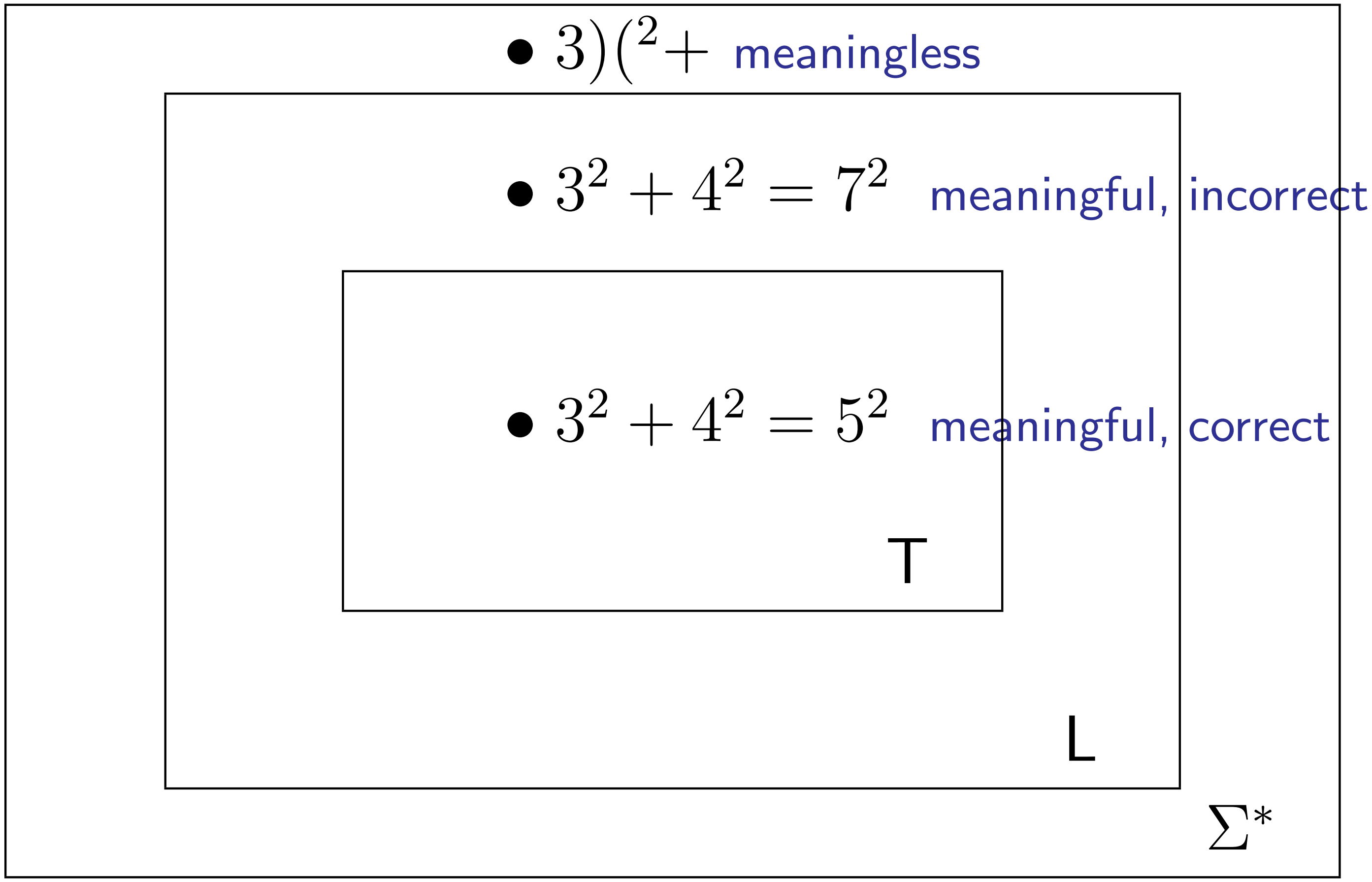
With a theory T we go one step further:

A theory in a language L is just a subset $T \subseteq L$

selecting a set of *correct* sentences

often such a theory is given by an axiomatic system

Words, language, theory



$$\Sigma \longmapsto \Sigma^* \longmapsto L \longmapsto T$$

Combinators

$$\Sigma_{\mathbf{CL}} = \{\mathbf{I}, \mathbf{K}, \mathbf{S}, x, ',), (, =\}$$

We introduce several simple regular grammars over $\Sigma_{\mathbf{CL}}$.

(i) $\boxed{\texttt{constant} \;:=\; \mathbf{I} \mid \mathbf{K} \mid \mathbf{S}}$

(ii) $\boxed{\texttt{variable} \;:=\; x \mid \texttt{variable}'}$

(iii) $\boxed{\texttt{term} \;:=\; \texttt{constant} \mid \texttt{variable} \mid (\texttt{term}\;\texttt{term})}$

(iv) $\boxed{\texttt{formula} \;:=\; \texttt{term} = \texttt{term}}$

Intuition:

in (FA) the term F stands for a *function* and A for an *argument*

Combinatory Logic

Axioms

$$\begin{array}{rcll} \mathbf{I}P &=& P & \textbf{(I)} \\ \mathbf{K}PQ &=& P & \textbf{(K)} \\ \mathbf{S}PQR &=& PR(QR) & \textbf{(S)} \end{array}$$

Deduction rules

$$\begin{array}{rcl} & & P = P \\ P = Q & \Rightarrow & Q = P \\ P = Q, Q = R & \Rightarrow & P = R \\ P = Q & \Rightarrow & PR = QR \\ P = Q & \Rightarrow & RP = RQ \end{array}$$

Here P, Q, R denote arbitrary terms (also called *combinators*

$\mathbf{I}P$ stands for $(\mathbf{I}P)$, $\mathbf{K}PQ$ for $((\mathbf{K}P)Q)$ and $\mathbf{S}PQR$ for $(((\mathbf{S}P)Q)R)$

In general $PQ_1 \ldots Q_n \equiv (..((PQ_1)Q_2) \ldots Q_n)$ (association to the left)

Write $P =_{\mathbf{CL}} Q$ if $P = Q$ is provable from these axioms and rules

Some magic with combinators

PROPOSITION.

(i) Let $\mathbf{D} \equiv \mathbf{SII}$. Then (doubling)

$$\mathbf{D}x =_{\mathbf{CL}} xx.$$

(ii) Let $\mathbf{B} \equiv \mathbf{S}(\mathbf{KS})\mathbf{K}$. Then (composition)

$$\mathbf{B}fgx =_{\mathbf{CL}} f(gx).$$

(iii) Let $\mathbf{L} \equiv \mathbf{D}(\mathbf{BDD})$. Then (self-doubling, life!)

$$\mathbf{L} =_{\mathbf{CL}} \mathbf{LL}.$$

PROOF.

(i)
$$\begin{array}{lcl} \mathbf{D}x & \equiv & \mathbf{SII}x \\ & = & \mathbf{I}x(\mathbf{I}x) \\ & = & xx. \end{array}$$

(ii)
$$\begin{array}{lcl} \mathbf{B}fgx & \equiv & \mathbf{S}(\mathbf{KS})\mathbf{K}fgx \\ & = & \mathbf{KS}f(\mathbf{K}f)gx \\ & = & \mathbf{S}(\mathbf{K}f)gx \\ & = & \mathbf{K}fx(gx) \\ & = & f(gx). \end{array}$$

(iii)
$$\begin{array}{lcl} \mathbf{L} & \equiv & \mathbf{D}(\mathbf{BDD}) \\ & = & \mathbf{BDD}(\mathbf{BDD}) \\ & = & \mathbf{D}(\mathbf{D}(\mathbf{BDD})) \\ & \equiv & \mathbf{DL} \\ & = & \mathbf{LL}. \end{array}$$

We want to understand and preferably also to control this!

First insight

LEMMA. For every term P and every variable x, there is a term Q such that x does not occur in Q and

$$Qx =_{\mathbf{CL}} P.$$

We denote this term Q constructed in the proof as $[x]P$.

PROOF. Induction on the complexity of P.

Case 1. P is a constant or a variable .

Subcase 1.1 $P \equiv \mathbf{C}$ with $\mathbf{C} \in \{\mathbf{I}, \mathbf{K}, \mathbf{S}\}$. Take $[x]\mathbf{C} \equiv \mathbf{KC}$. Then indeed

$$([x]\mathbf{C})x =_{\mathbf{CL}} K\mathbf{C}x =_{\mathbf{CL}} \mathbf{C}.$$

Subcase 1.2 $P \equiv x$. Take $[x]x \equiv \mathbf{I}$. Then

$$([x]x)x \equiv \mathbf{I}x =_{\mathbf{CL}} x.$$

Subcase 1.3 $P \equiv y \not\equiv x$. Take $[x]y \equiv \mathbf{K}x$. Then indeed

$$([x]y)x \equiv \mathbf{K}yx =_{\mathbf{CL}} y.$$

Case 2. $P \equiv UV$. Take $[x](UV) \equiv \mathbf{S}([x]U)([x]V)$. Then indeed

$$([x](UV))x \equiv \mathbf{S}([x]U)([x]V)x =_{\mathbf{CL}} (([x]U)x)(([x]V)x) =_{\mathbf{CL}} UV. \blacksquare$$

Algorithms

The previous proof gave

P	$[x]P$	$([x]P)x = P?$
$\mathbf{C}$	$\mathbf{KC}$	$\mathbf{KC}x = \mathbf{C}$
x	$\mathbf{I}$	$\mathbf{I}x = x$
$y \not\equiv x$	$\mathbf{K}y$	$\mathbf{K}yx = y$
UV	$\mathbf{S}([x]U)([x]V)$	$\mathbf{S}([x]U)([x]V)x =$ $(([x]U)x)(([x]V)x) = UV$

More efficient algorithm

P	$[x]P$
x	$\mathbf{I}$
P with $x \notin P$	$\mathbf{K}P$
UV	$\mathbf{S}([x]U)([x]V)$

Second Insight: Fixed Points

THEOREM For all combinators P there exists an X such that

$$PX =_{\mathbf{CL}} X$$

PROOF. Given P, define

$$\begin{array}{rcl} W & \triangleq & [x]P(xx) \\ X & \triangleq & WW \end{array}$$

Then X is a so called *fixed point* of P.

$$\begin{array}{rcl} X & \equiv & WW \\ & \equiv & ([x]P(xx))W \\ & =_{\mathbf{CL}} & P(WW) \\ & \equiv & PX \end{array}$$

Hence $PX =_{\mathbf{CL}} X$. ■

$\mathbf{L}$ is a fixed point of $\mathbf{D}$: one has $\mathbf{L} = \mathbf{DL} = \mathbf{LL}$.