Formal Systems / Formalisms

Algebra, Calculi, Logic

Golovach Ivan

RChain Coop

July, 2018

Formalisms

Alphabet, Word, Lang

Alphabet

An alphabet Σ is a finite fixed set of symbols: numbers, letters ('a', 'ω', 'Δ', '$\mathbb{N}$'), punctuation ('(', ':', ']'), special('$\perp$', '$\Rightarrow$', '$\forall$', '$\neg$').

Word

A word over an alphabet is any finite sequence of alphabet symbols. For any alphabet, there is only one word of length 0, the empty word (""), which is denoted by ϵ. Set of all words over alphabet Σ denoted Σ^*.

Example

Strings of brackets

$\Sigma = \{(,)\}$

$\Sigma^* = \{\epsilon, (,), ((, (),)(,)), (((, ((), ()(, ...\}$

Formal language

A formal language L over an alphabet Σ is a fixed subset (finite or infinite) of Σ^*. A formal language is a purely syntactic construction without any semantics. Such a language exist before any interpretation is assigned to its words that is, before it has any meaning.

Example

Dyck lang
$\Sigma = (,)$
Words - correctly nested finite sequences of brackets.
()(()(())) - a WORD of the Dyck lang.
))((- NOT a word of the Dyck lang (not correctly nested).
()()()()()... - NOT a word of the Dyck lang (infinite).

BNF / Context-Free Language

A BNF definition (or context-free grammar) of formal language is a set of production rules that describe all possible words in a given formal language.

Example

Dyck Lang
Terminals/alphabet: $\Sigma = \{(,)\}$
Non-terminals: { D }
Production rules: $D ::= \epsilon | (D)D$
Start symbol: { D }
Term generation:
$D \to (D)D \to (D)\epsilon \to (D) \to ((D)D) \to ((D)\epsilon) \to ((D)) \to ((\epsilon)) \to (())$

Formalisms

Algebra, Calculi, Logic

Formalism

???

Example

Algebra - formalism with focus on EQUALITY.

Example

Calculi - formalism with focus on REDUCTION.

Example

Logic - formalism with focus on TRUTH values and QUANTIFIERS.

Formalisms

Algebra

???

???

Example

???

Formalisms

Calculi

???

???

Example

???

Formalisms

Logic

???

???

Example

???

Process combinators

$+$ - alternative composition
; - sequential composition
$|-$parallel.composition

Baeten, "A Brief History of Process Algebra":

$x + y = y + x$
$x + (y + z) = (x + y) + z$
$x + x = x$
$(x + y); z = x; z + y; z$
$(x; y); z = x; (y; z)$
$x \mid y = y \mid x$
$(x \mid y) \mid z = x \mid (y \mid z)$

Formalisms

Process Calculi

???

???

Example

???

Process logic

Hennessy-Milner Logic (HML)
Caires/Cardelli Logic
Namespace Logic

Example

HML formulae for $\exists(\xrightarrow{a} \cdot \xrightarrow{b} \cdot \xrightarrow{c})$:
$\phi = (\langle a \rangle tt) \wedge ([a]\langle b \rangle tt) \wedge ([a][b]\langle c \rangle tt)$

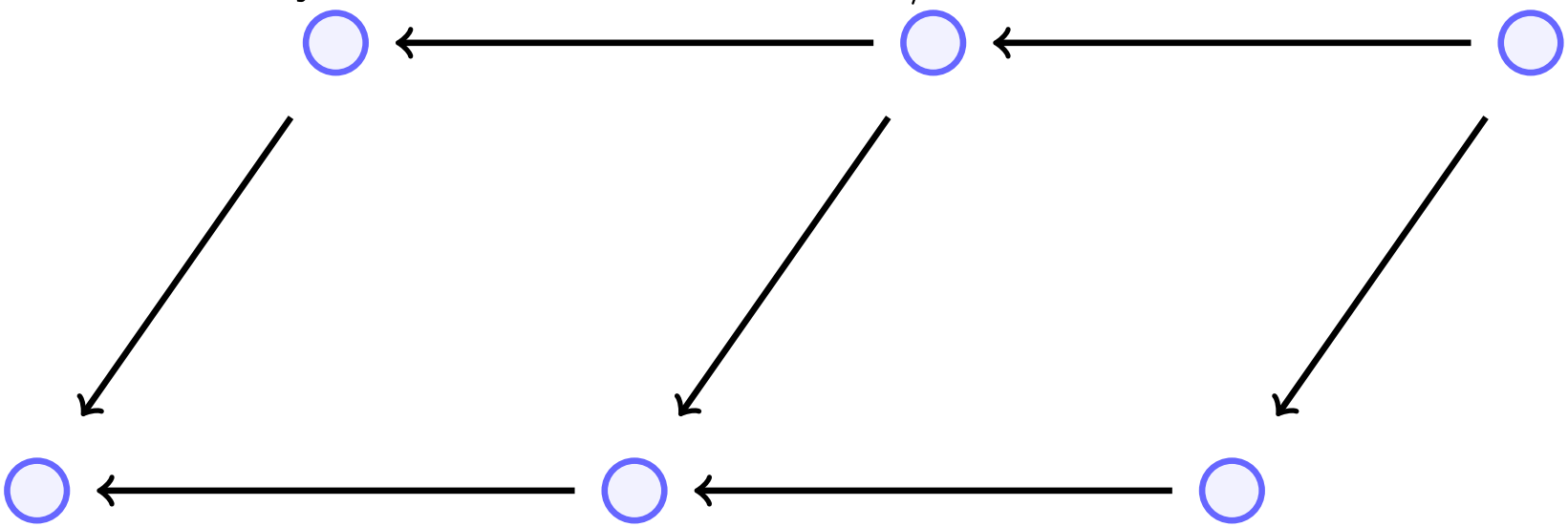
Formalisms

Hierarchy

LOGIC

.

.....Hennessy-Milner.....Caires/Cardelli.....Namespace



ccs.....pi-calculi.....rho-calculi

.

PROCESS CALCULI