Discrete mathematics, specification languages, and programming languages

Dr. Constantinos Constantinides, P.Eng.

Department of Computer Science and Software Engineering
Concordia University

Full compatibility of specification/programming languages with discrete mathematics

- Not every concept in discrete mathematics is supported by all specification languages and all programming languages.

- For example, the cons (list construction) operation in discrete mathematics:
  - Not supported by the Z Specification or the Object-Z specification.
  - Supported by the Lisp programming language.

Consistency in notation

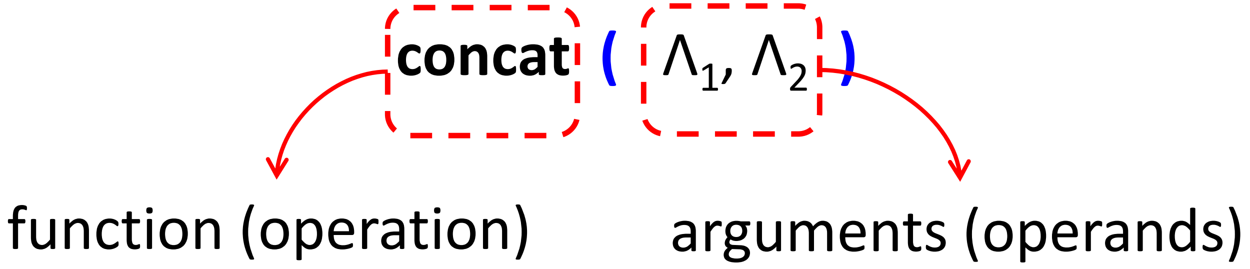
- Notation in discrete mathematics is not always consistent to that in specification languages (e.g. Z, Object-Z), or in programming languages (e.g. Lisp, Prolog, Java).

- For example: The concatenation operation.

– Math:	concat
– Z, Object-Z:	$\frown$
– Lisp:	append
– Java:	<string> + <string> (for string concatenation)

Consistency in notation /cont.

DISCRETE MATHEMATICS

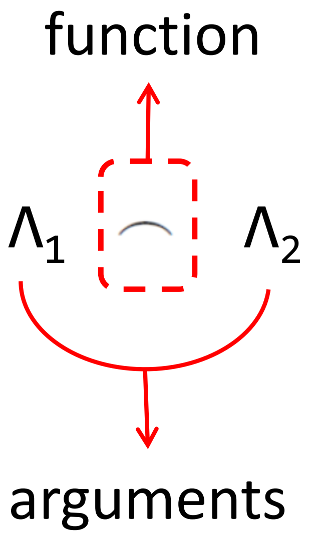


concat (Λ_1, Λ_2)

function (operation)

arguments (operands)

SPECIFICATION LANGUAGE (Z, OBJECT-Z)

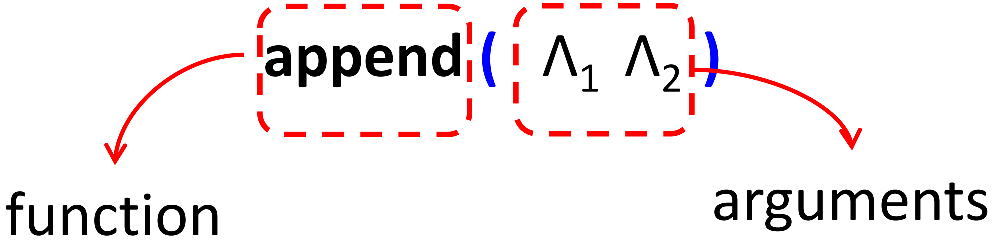


function

$\Lambda_1 \frown \Lambda_2$

arguments

PROGRAMMING LANGUAGE (LISP)



append $(\Lambda_1\ \Lambda_2)$

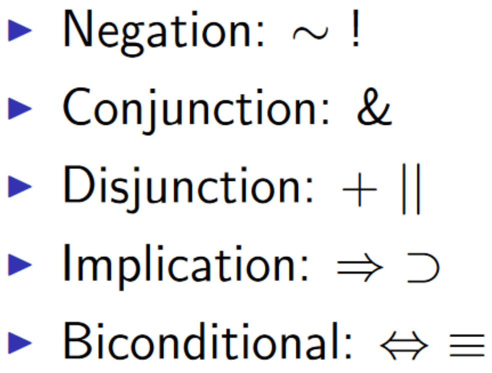
function

arguments

Alternative notations

- Sometimes, alternative notations exist in discrete mathematics.

- For example, compare the notation we use in *Propositional Logic* with the following alternatives:



  - Negation: $\sim$!
  - Conjunction: &
  - Disjunction: $+$ $||$
  - Implication: $\Rightarrow$ $\supset$
  - Biconditional: $\Leftrightarrow$ $\equiv$

- Though not wrong to mix notations, it is generally considered a bad practice.

Overloaded notations

- The $\oplus$ symbol indicates *relational override* (see ‘Relations’).
- It is also used to indicate *exclusive disjunction* (ex-or).
- In examples (such as in Z/Object-Z) where we might have both concepts, we will use $\underline{V}$ as an alternative notation to ex-or.