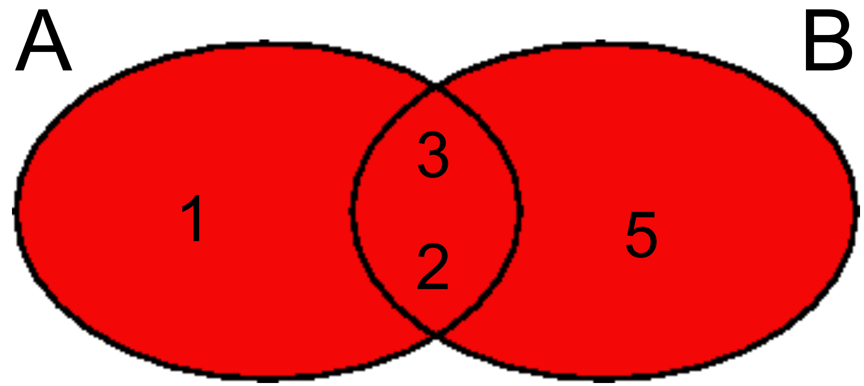
Maths for Computing

Lecture 5

Thomas Nowotny
Chichester 1, Room CI 105
t.nowotny@sussex.ac.uk

SETS AND INTERVALS

Set operations



Union:

$$A \cup B = \{x : x \in A \text{ or } x \in B\}$$

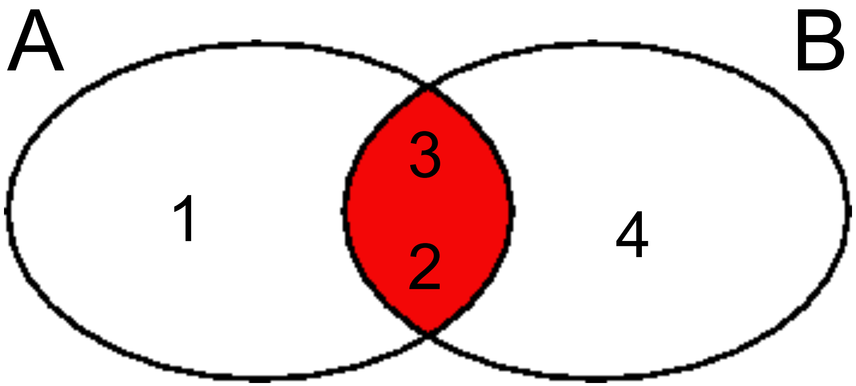
Examples:

No duplication of elements!

$$\{1,2,3\} \cup \{2,3,5\} = \{1,2,3,5\}$$

$$\{1,2,3,\ldots\} \cup \{-1,-2,-3,\ldots\} \cup \{0\} = \mathbb{Z}$$

Set Operations



Intersection:

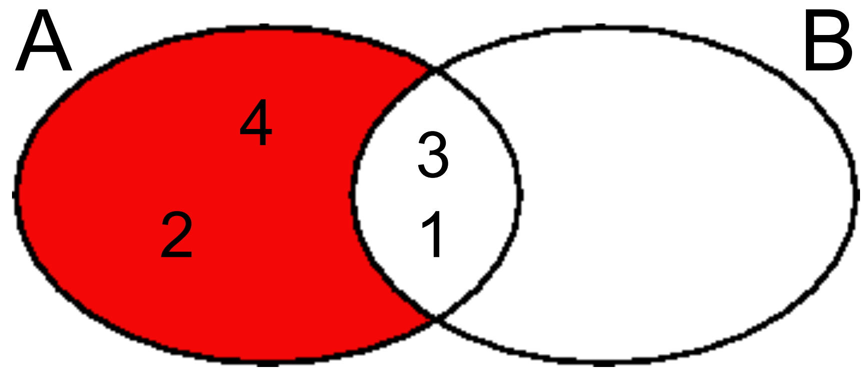
$$A \cap B = \{x : x \in A \text{ and } x \in B\}$$

Examples:

$$\{1, 2, 3\} \cap \{2, 3, 4\} = \{2, 3\}$$

$$\mathbb{N} \cap \mathbb{Z} = \mathbb{N}$$

Set Operations



Subtraction:

$$A \setminus B = \{x \in A \,:\, x \notin B\}$$

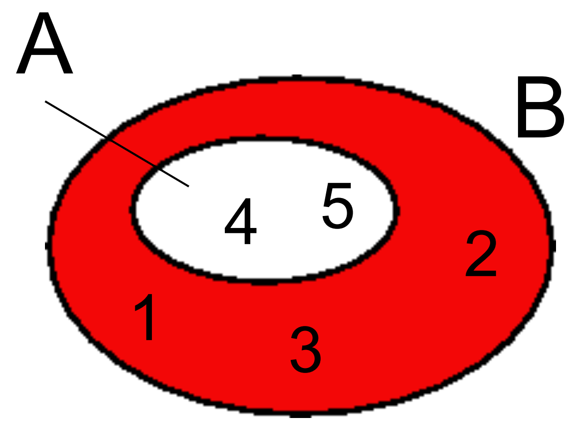
alternative notation: $A - B$

Example:

$$\{1,2,3,4\} \setminus \{1,3\} = \{2,4\}$$

I like to read “without”

Set operations



Complement:

If $A \subset B$ then $A^C = B \setminus A$

" A^C is the complement of A in B "

Example:

$B = \{1, 2, 3, 4, 5\}$ $A = \{4, 5\}$

$A^C = \{1, 2, 3\}$

Dictionary of set theory

Symbol	Meaning	Symbol	Meaning
$\in$	Element of	$\{\ldots\}$	Set of elements
$\subset$, $\subseteq$	Subset	$\setminus$	Subtract, "without"
$\supset$, $\supseteq$	Superset	^{c}	Complement
$\cap$	Intersection	card	Cardinality
$\cup$	Union	$\emptyset$	**Empty set**

Intervals

- Special type of sets, parts of $\mathbb{R}$

In the following: $x, y \in \mathbb{R}$ and $x < y$

$$[x, y] = \{z \in \mathbb{R} : x \leq z \leq y\}$$

All numbers between x and y (**boundaries included**)

Other intervals

alternative
notation

$$]x, y[= \{z \in \mathbb{R} : x < z < y\} = (x, y)$$

… as before but **boundaries not included**

$$]x, y] = (x, y]$$

$$[x, y[= [x, y)$$

… and so on.

Important

- Different brackets mean completely different things!

$$[x, y]$$

All numbers of $\mathbb{R}$ between x and y, including x and y. **Infinitely many numbers!**

$$\{x, y\}$$

Literally, just the numbers x and y. **2 numbers**

REGULAR EXPRESSIONS

Alphabet, words

- An **alphabet** $\mathcal{S}$ is a set of symbols (or letters), e.g.
  $\mathcal{S} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
  $\mathcal{S} = \{\text{a, b, c, d, e, \ldots, x, y, z}\}$
  (mathematically it's just a set)
- A **word** is a combination or string of symbols from $\mathcal{S}$, e.g.
  123, 1000, 0102, a, hello, world, xza
- The symbol ϵ stands for the empty string
  (greek letter "epsilon")

Language

- A **language** $\mathcal{L}$ is a set of words (over an alphabet $\mathcal{S}$), e.g.

$$\mathcal{L} = \{00, 01, 10, 11\}$$ (finite language)

$$\mathcal{L} = \{01, 001, 0001, 00001, \ldots\}$$ (infinite language)

$$\mathcal{L} = \{\text{hello}, \text{world}\}$$ (finite language)

Regular languages

- A regular language can be defined like this (over an alphabet $\mathcal{S}$):
  - The empty language is regular
  - The singleton language $\{a\}$ is regular ($a \in \mathcal{S}$)
  - If $\mathcal{A}$ and $\mathcal{B}$ are regular languages, then $\mathcal{A} \cup \mathcal{B}$ (union) and $\mathcal{A} \circ \mathcal{B}$ (concatenation) and $\mathcal{A}*$ (Kleene star) are regular.
  - No other languages are regular

Some examples for the operations

Alphabet $\mathcal{S} = \{\mathrm{a}, \mathrm{b}\}$

Language $\mathcal{A} = \{\mathrm{a}, \mathrm{aa}\}$

Language $\mathcal{B} = \{\mathrm{b}, \mathrm{bb}\}$

- Union:

  $\mathcal{A} \cup \mathcal{B} = \{\mathrm{a}, \mathrm{aa}, \mathrm{b}, \mathrm{bb}\}$

- Concatenation:

  $\mathcal{A} \circ \mathcal{B} = \{\mathrm{ab}, \mathrm{abb}, \mathrm{aab}, \mathrm{aabb}\}$

- Kleene star:

  $\mathcal{A}* = \{\mathrm{a}, \mathrm{aa}, \mathrm{aaa}, \mathrm{aaaa}, \ldots\}$

Regular expressions

- Regular expressions are used to define regular languages
- A regular expression describes the legal word in a language by a matching operation:
  - '*a*' matches the symbol 'a' in the alphabet
  - The '|' denotes alternatives (Boolean (x)or)
  - Brackets '(' and ')' are used for grouping

Regular expressions

- ‘*’ matches zero or more of the preceding symbol
- ‘+’ matches one or more of the preceding symbol
- ‘?’ matches 0 or 1 of the preceding symbol

- How these work is best explained by examples:

RegExp: Examples (or)

Regular expression — Match

- *ab* matches — ab
- *a|b* matches — a, and matches b
- *aa|bb* matches — aa and matches bb
- *(aa)|(bb)* matches — aa and matches bb
- a(a|b)b matches — aab and matches abb

Identical by definition

RegExp: Examples (multipliers)

- *a*$*$ matches ϵ, a, aa, aaa, …
- *ab*$*$ matches a, ab, abb, …
- *(ab)*$*$ matches ϵ, ab, abab, …
- *(ab)*+ matches ab, abab, …
- *(ab)?* matches ϵ, ab

More about RegExp

- The symbol '+' isn't really necessary, why?
  '*a*+' is the same as '*aa*$*$'
- The Kleene star in RegExp is not the '$*$' you would use in a filename as a wildcard
- The '?' equally is not what you would use in a file name, e.g.

  (Unix) wildcard: 'h$*$' matches 'hello', 'hrt', …

  RegExp '*h*$*$' matches ϵ, 'h', 'hh', 'hhh', …
  (Unix) wildcard: 'h?llo' matches 'hello' and 'hallo'
  RegExp '*h?llo*' matches 'llo', 'hllo'

Additional notation

- To get the 'wildcard' type expressions, these notations can be used:
  - '*a{a,b,c}*$*$' which denotes 'a' followed by any number of 'a', 'b', or 'c' in any mix.
  - '*a* $\mathcal{S}*$' would be the equivalent of the wildcard type star
  - '*a* $\mathcal{S}$' would be the equivalent of the wildcard questionmark – Note: **Not** '*a* $\mathcal{S}$ *?*' !!!
  - These are just notations for something we already knew – what is it?

Core vs additional notation

- $a\mathcal{S}$ is the same as *a(a|b|c|d…|z)*
- $a\mathcal{S}$ *?* is the same as *a(a|b|c|…|z)?*

Precedence of operators

Precedence	Operator
Highest	(,)
Middle	?, *, +
Lowest	|

BB Examples

- a?b? matches ϵ , a , b, ab
- a $*$ b $*$ matches a, b, aa, bb, ab, aab, abb, aaab, aabb, abbb, …
- $\mathcal{S}$ ={a,b}, then $\mathcal{S}*$ matches everything made of ‘a’ and ‘b’
- $\mathcal{S}$ ={a, b, …, z}, then a $\mathcal{S}*$ matches all words starting with ‘a’
- ab | ba matches ab, ba
- a(b|a)a matches aba, aaa