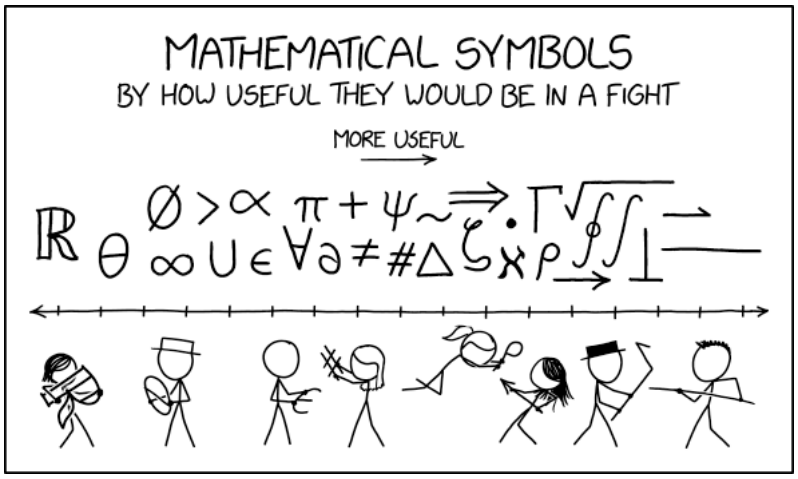
COMS30048 hand-out: notation and terminology



Within *any* document, the notation and terminology used can have a subtle but important impact on clarity and hence ease of understanding. However, selecting and implementing a scheme can be much harder than it might seem: to see why, consider that a) different documents can, legitimately, use different schemes: often this is partly due to subjectivity, but also because the selection depends on the context (e.g., aims or intended audience), b) precedence ofter trumps quality, in the sense it is common to follow an existing scheme rather than improve it, and c) terminology in particular can change over time. With no de facto "best" scheme, the teaching material you are provided with follows standard schemes wherever possible, but *attempts* to be as clear and (self-)consistent as possible. The following offers a complete overview wrt. this unit (or part thereof); if you identify a missing case or a case used incorrectly or inconsistently somewhere, let someone know so it can be corrected!

General

- n usually denotes a fixed size (or length), often some parameter, with n_x the equivalent wrt. a specific x.
- l usually denotes a variable size (or length), often some parameter, with l_x the equivalent wrt. a specific x.
- $x \leftarrow y$ is used to denote an assignment of the value y to the variable x.
- $x \xleftarrow{\$} Y$ is used to denote an assignment of a value y, sampled uniformly at from the set Y st. $y \in Y$, to the variable x.
- $x \stackrel{?}{=} y$ is used to denote a test or comparison, in this case equality, between x and y that yields a Boolean (i.e., **true** or **false**) result.
- $x \ldots y$ denotes the range of values between x and y inclusive; this implies a need to generate intermediate values, st. the notation is only useful iff. doing so is obvious *and* unambiguous.
- $\Pr[x]$ denotes the probability of event x occurring.
- $\hat{x}$ denotes the representation of some value x, which must be interpreted wrt. the on the context.
- $|X|$ denotes the cardinality (or size) of some object X.
- X_i refers to an indexed (or numbered) element within some object X, namely the i-th such element.
- $X[Y]$ refers to a named (or labelled) field Y within some object X.
- $X.Y$ refers to a named (or labelled) field Y within some name-space X; this is useful to disambiguate different Y with the same identifier.
- Specific use of fonts often highlights the purpose of an identifier:
  - X is used to identify a variable,
  - X is used to identify an algorithm,

[1] http://xkcd.com/2343

- $\mathcal{X}$ is used to identify a program,
- $\mathbf{X}$ is used to identify a process,
- $\mathcal{X}$ is used to identify a participant (or party).

Quantities

- For clarity, we use the modern prefixes to distinguish between binary and decimal SI units: for example, 1MB = $1 \cdot 10^6$B whereas 1MiB = $1 \cdot 2^6$B.
- Given a word size w (e.g., the natural size as dictated by a given processor), we assume

$$\begin{aligned} \text{bit} &\equiv 1\text{-bit} \\ \text{nybble} &\equiv 4\text{-bit} \\ \text{byte} &\equiv 8\text{-bit} \\ \\ \text{half-word} &\equiv (w/2)\text{-bit} \\ \text{word} &\equiv w\text{-bit} \\ \text{double-word} &\equiv (w \cdot 2)\text{-bit} \\ \text{quad-word} &\equiv (w \cdot 4)\text{-bit} \end{aligned}$$

  but note that standards in particular often use the term octet as a synonym for byte (st. an octet string is therefore a byte-sequence): although less natural, we follow this terminology where it seems of value to match associated literature.

Collections (i.e., sets, sequences and tuples)

- An n-element tuple X is denoted
$$X = (X_0, X_1, \ldots, X_{n-1})$$
  where X_i denotes the i-th element, and $|X|$ denotes the cardinality (or size).
- An n-element set X is denoted
$$X = \{X_0, X_1, \ldots, X_{n-1}\}$$
  where X_i denotes the i-th element, and $|X|$ denotes the cardinality (or size); so-called set builder notation allows a short-hand st.
$$X = \{x \mid f(x)\}$$
  denotes the set of all x st. the predicate $f(x) =$ **true**.
- An n-element sequence X is denoted
$$X = \langle X_0, X_1, \ldots, X_{n-1} \rangle$$
  where X_i denotes the i-th element, and $|X|$ denotes the cardinality (or size). Note that sequences are read left-to-right, so
$$X = \langle X_0, X_1, \ldots, X_{n-1} \rangle$$
  has the first (resp. last) element on the left (resp. right); using this ordering matches a static array definition in C.
- If X and Y are sequences, $X \parallel Y$ denotes their concatenation: given
$$X = \langle X_0, X_1, \ldots, X_{n-1} \rangle$$
  and
$$Y = \langle Y_0, Y_1, \ldots, Y_{n-1} \rangle,$$
  we have
$$X \parallel Y = \langle X_0, X_1, \ldots, X_{n-1}, Y_0, Y_1, \ldots, Y_{n-1} \rangle.$$
  That is, a natural order of elements in the result is maintained if read left-to-right: X, the LHS, contributes the lower-indexed elements, whereas Y, the RHS, contributes the higher-indexed elements.
- A (sparse) set of indices, including ranges, can be used to construct most collections, with an appropriate means of combination assumed. Consider a sequence X, for example, where
$$X_{0,\ldots,2,5,7} \equiv X_0 \parallel X_1 \parallel X_2 \parallel X_5 \parallel X_7.$$

- There are some special-case sets to keep in mind:
  - $\emptyset$ is the empty set,
  - $\mathcal{U}$ is the universal set,
  - $\mathbb{B}$ is the set of binary digits (i.e., bits),
  - $\mathbb{N}$ is the set of natural numbers,
  - $\mathbb{Z}$ is the set of integers,
  - $\mathbb{Z}_N$ is the set of integers modulo N,
  - $\mathbb{Z}_N^*$ is the set of integers modulo N restricted to those with a multiplicative inverse (i.e, the multiplicative group, of size $\Phi(N)$, formed from integers coprime to N),
  - $\{0,1\}^n$ is the set of bit-sequences of length n,
  - $\{0,1\}^*$ is the set of bit-sequences of arbitrary (but finite) length.

Bits and bit-sequences

- The operators $\neg$, $\wedge$, $\vee$ and $\oplus$ denote Boolean NOT, AND, OR, and XOR respectively, with $\overline{\wedge}$, $\overline{\vee}$, and $\overline{\oplus}$ denoting NAND, NOR, and NXOR; all of these may be overloaded to cater for bit-sequences rather than simply bits.
- $\text{HW}(X)$ denotes the Hamming weight of bit-sequence X, st.

$$\text{HW}(X) = \sum_{i=0}^{i<n} X_i.$$

- $\text{HD}(X)$ denotes the Hamming distance between bit-sequences X and Y, st.

$$\text{HD}(X,Y) = \sum_{i=0}^{i<n} X_i \oplus Y_i.$$

- $\textsc{Par}^+(X)$ (resp. $\textsc{Par}^-(X)$) denotes the even (resp. odd) parity bit for X, st.

$$\begin{array}{rcrcrcl} \textsc{Par}^+(X) & = & \sum_{i=0}^{i<n} X_i & \pmod 2 & = & \bigoplus_{i=0}^{i<n} X_i \\ \textsc{Par}^-(X) & = & \neg\,(\,\sum_{i=0}^{i<n} X_i & \pmod 2\,) & = & \neg\,(\,\bigoplus_{i=0}^{i<n} X_i\,) \end{array}$$

  noting that $\textsc{Par}(X)$ is used if/when the type is irrelevant.
- $\text{LSB}_l(X)$ (resp. $\text{MSB}_l(X)$) denotes the l least- (resp. most-) significant bits of X (where we assume $l = 1$ if omitted).
- The operators $\ll$ and $\gg$ denote left- and right-shift; $\lll$ and $\ggg$ denote left- and right-rotate (beware of context: $\ll$ and $\gg$ are also used to denote "much less than" and "much greater than").

Numeric representation and operations

- $x_{(b)}$ denotes x is expressed in radix- or base-b; where no base is specified, it is safe to assume decimal (i.e., that $b = 10$).
- Writing a literal $x = 123_{(10)}$ is equivalent to writing a sequence

$$x = \langle 3, 2, 1 \rangle_{(10)},$$

  in these sense x_i, the i-th digit of x, is well defined in both cases.
- $|x_{(b)}|$ denotes the number of digits in x, if expressed in base-b.
- Unless otherwise stated, a little-endian digit ordering is assumed; the least-significant (resp. most-significant) digit is thus x_0 (resp. x_n).
- For some x, we let $\text{ext}_0^w(x)$ and $\text{ext}_\pm^w(x)$ respectively denote zero- or sign-extension to w-bits (allowing omission of either specifier where appropriate).
- For some operator $\odot$, we let $\odot_s^w$ and $\odot_u^w$ respectively denote w-bit signed and unsigned variants (allowing omission of either specifier where appropriate).

Digital logic

- **false** and **true** are used to denote Boolean logic values in a decisional context, with 0 and 1 preferred in a computational context.
- **Z** denotes the high impedance value: this is typically used to capture a floating (i.e., disconnected) or null value, which can be overridden by 0 or 1.
- **?** denotes the don't care value: if $x =$ **?** then x can be 0 or 1 without impacting on the validity of associated computation.
- **X** denotes the unknown value: if $x =$ **X** we do not *know* (take care: this differs from do not *care*) whether x is 0 or 1 (e.g., the wire x is driven inconsistently).
- *GND* is used to denote the ground voltage level, whereas V_{dd}, V_{ss}, V_{cc} and V_{ee} are variously used to denote the drive voltage level. Although too imprecise for some purposes, for concreteness it is enough to consider that

$$\begin{array}{lcrcl} GND & \equiv & 0 & \simeq & 0\text{V} \\ V_{dd} & \equiv & 1 & \simeq & 5\text{V} \end{array}$$

Processor design and behaviour

- $\mathsf{GPR}[x]$ denotes the w-bit general-purpose register number x.
- R denotes an n-bit special-purpose register R. Often such registers are divided into fields, in which case $\mathsf{R}[f]$ denotes the field f within register R. For example, $\mathsf{CSPR}[M]$ denotes the processor mode field in the ARM Current Program Status Register (CSPR); the five least-significant bits of CSPR represent this field, so we *could* write

$$\mathsf{CSPR}[M] \equiv \mathsf{LSB}_5(\mathsf{CSPR}) \equiv \mathsf{CSPR}_{0\ldots4}$$

  instead (although they are arguably less descriptive).
- $\mathsf{MEM}[x]$ denotes the 8-bit memory location x.
- $\mathsf{MEM}[x]^y$ denotes the 8-bit memory locations x through to $x + y - 1$,, i.e.,

$$\mathsf{MEM}[x]^y \equiv \mathsf{MEM}[x+0] \parallel \mathsf{MEM}[x+1] \parallel \cdots \parallel \mathsf{MEM}[x+y-1]$$

  assuming

$$\mathsf{MEM}[x] \equiv \mathsf{MEM}[x]^1.$$

Cryptography

- x is a public value x, and x is a private (or secret) value.
- $x^{(k)}$ denotes some quantity x relating to the k-th round (e.g., the k-th round key).
- $x[\![i]\!]_j^b$ denotes the j-th byte in an i-th, b-byte block within some larger, n-block quantity (e.g., plaintext or ciphertext) x; the brackets could be viewed as "splitting" x into blocks and selecting the i-th such block
- Where not obvious, arithmetic operations may be suffixed by their type (or structure they apply to); for example, $\oplus_{\mathbb{F}_{2^8}}$ denotes an addition in the finite field $\mathbb{F}_{2^8}$.
- For a group $\mathbb{G}$, writing $\mathbb{G} = \langle g \rangle$ means that the group is generated by the generator g.
- For an additive group $\mathbb{G}$, let $[y]\,x$ denote (scalar) multiplication of x by $y \in \mathbb{Z}$ (this is more often written $y \cdot x$ where $x \in \mathbb{Z}$, for example). For an multiplicative group $\mathbb{G}$, let x^y denote exponentiation of x by $y \in \mathbb{Z}$ (this is more often written $y \cdot x$ where $x \in \mathbb{Z}$, for example).
- $\mathsf{negl}(\mathsf{n})$ is used to denote a negligible function.
- While their use is context-specific, it is *usually* the case that
  - λ denotes a security parameter,
  - τ denotes a MAC tag,
  - σ denotes a digital signature (or certificate),
  - μ denotes a hash or encoding function (typically used in a digital signature scheme),
  - κ denotes the RSA key-equation variable (i.e., st. $e \cdot d = 1 + \kappa \cdot \Phi(N)$),
  - ς denotes a PRG seed, and
  - ρ denotes the padding applied so some message.