Concepts, Terminology, and Notations for IOI Competition Tasks

Tom Verhoeff*

12 September 2004 (To be presented at IOI 2004)

Abstract

In this document, we classify concepts, terminology and notations with respect to their usability in IOI competition tasks. We distinguish three usability classes: *basic knowledge* (can be used without further ado), *to-be-defined* (can be used, but must be defined explicitly), and *to-be-avoided* (must not be used). The classification primarily concerns the use in task descriptions, which are presented to contestants at the beginning of the competition. But it also concerns solutions handed out after the competition. This classification can benefit both organizers and participants in their preperations for an IOI competition.

Status: Content to be discussed; not approved for official use.

1 Introduction

There are several reasons for imposing restrictions on the use of concepts, terminology, and notations in IOI competition tasks:

1. The IOI competition is *not* intended to test knowledge, but rather to test algorithmic problem-solving skills. Many IOI contestants have not yet completed secondary education. Therefore, the IOI organizers cannot assume that contestants have much prior knowledge of concepts, terminology, and notation in the area of algorithmic problems (or any other area, for that matter).

2. IOI contestants come from all over the world and have diverse educational backgrounds. The organizers should aim at formulations that are understandable by all contestants.

3. Concepts, terminology, and notations used in computing science are fraught with complications. CS professionals have learned to cope with these after many years of training. For example, the concept of a graph is rather vague as such. It requires many additional details to make precise what is meant. A CS professional can often infer these details from subtle clues in the context. This cannot be expected from an average IOI contestant.

During the preparations for past IOI competitions, and even during those IOIs, the appropriateness of concepts, terminology, and notations used in competition tasks has been discussed and decided. This has often led to changes in task descriptions.

*Faculty of Mathematics and Computing Science, Eindhoven University of Technology. E-mail: `T.Verhoeff@TUE.NL`

In this document, we consider an extensive list of concepts, terminology and notations, and classify them with respect to their usability in IOI competition tasks. We focus on the use in (official English) task descriptions presented (possibly after translation) to contestants at the beginning of the competition. Somewhat less severe restrictions apply to solutions handed out after the competition.

We hope that by standardizing the classification, we can avoid redoing "old" discussions at every IOI. Standardization will help task authors to write better initial vesions. It will also improve the effectiveness of ISC review meetings, leaving more time for other matters. Furthermore, standardization will enable coaches to prepare their contestants better, which we hope will reduce the number of (unnecessary) low scores. Finally, it may be useful for CS education in secondary education in general.

The restrictions mentioned in this document apply to the organizers only. There are no restrictions on what concepts, terminology, and notations the contestants may use in their own work, except that the programs they submit must be written in one of the approved programming languages.

Matters like style, logical structure, layout, and typography of task descriptions and solutions also need attention, but they fall outside the scope of this document.

1.1 Approach

Let us first clarify what we mean by concept, terminology, and notation.

'Concept' is a rather abstract notion in itself. It refers to a general idea with a specific focus. For instance, the concept of a *number*, a *set*, *incidence* (in geometry). Often, a concept is not atomic, but a composite of (simpler) concepts.

'Terminology' refers to the typical words used in connection with a concept. For instance, the *elements* of a *set*, a point *lies on* a line.

'Notation' refers to the typical (mathematical) symbols used in connection with a concept. For instance, { 0, 1, 2 } denotes the set consisting of the three numbers 0, 1, and 2.

We distinguish three classes in the usability classification. In order of increasing restrictiveness, these are:

- Basic Knowledge (BK): can be used without further definition.
- To-Be-Defined (TBD): can be used, but needs to be defined explicitly.
- To-Be-Avoided (TBA): cannot be used.

Note that, for a given concept, the related terminology and notations may end up in different classes.

The classification presented here was obtained as a combination of

- Personal experiences gathered during meetings at past IOIs.
- Personal experiences from teaching and organizing programming contests.
- Analysis of task descriptions of past IOI competitions.
- Discussions with various members of the IOI community.

When classifying '...' (fill in the concept, terminology, or notation to be classified), one can consider the following question.

> How appropriate would you find a sentence of the form 'Let X be ...', when it would appear in a task description without an explicit definition of '...'?

Here are some examples:

1. 'Let x be a number'
2. 'Let n be a positive integer'
3. 'Let p be a prime less than n'
4. 'Let S be a set of n elements'
5. 'Let S be $A \cap (B \cup C)$'
6. 'Let G be a graph'
7. 'Let R be a transitive relation on S'
8. 'Let M be an $n \times n$ matrix of integers'
9. 'Let T be a triangle with vertices A, B, and C'
10. 'Let P be a point on line ℓ'

Typical responses are likely to be:

1. Too vague, could x be a fraction, a real number, a complex number?
2. OK, but at some point an upper bound on n may need to given as well.
3. OK, but what about $p = 1$, and does everyone know the definition of prime? Also see problem The Primes on Day 1 of IOI'94.
4. OK, but with a slight hesitation.
5. OK, if you write it out carefully in words, but the notation may not be common high school knowledge, or is it?
6. Too vague, there are so many kinds of graphs.
7. No, relations and their properties in an abstract sense are, well, too abstract. IOI contestants should be able to handle concrete relations, but not like this.
8. No, matrices are not commonly treated in high school.
9. OK, but should it be obvious to them that A, B, and C are not collinear?
10. OK, because the word *incidence* is not used.

The classification may evolve over time. Certain concepts, terminology, and notation are begging to be promoted to BK.

1.2 Organization

The classification is organized in three sections, one for each class. Within a class, the entries are grouped by 'relatedness'. Each entry is labeled by one or more words in boldface. When the words are in italics, this means that they refer to a concept, and that the words are not themselves the terminology being classified (see e.g. *incidence* under Basic Knowledge, Geometry). Otherwise, these words are also the actual terminology being classified. Where relevant, the related notation is given as well. In some cases, a motivation is also included.

Note that translators may (have to) substitute equivalent terminology (or more elaborate descriptions) and notations, when this is more appropriate for their contestants. Translators may also request to include a basic definition, if they feel that their contestants are not sufficiently familiar with the matter. However, translators may not put in extra information that goes beyond the definition.

In various descriptions, we have used a sequence of subscripted variables. For instance, see *decimal notation*, *set*, *tuple*, and *sequence*. As a convention in this document, we have chosen to start with the subscript 0; also see [3]. Note, however, that the classification would not be affected when starting with subscript 1. This is a matter of meta-notation.

1.3 Abbreviations

adj. Adjective

BK Basic Knowledge

iff If and only if

n. Noun

rel. Relation

TBA To Be Avoided

TBD To Be Defined

v. Verb

2 Basic Concepts, Terminology, and Notations

Although we mention some definitions, this document is not intended to offer a complete formalization. Also see [6].

2.1 Numbers

Integer (n.) A whole number, signed: $\ldots, -2, -1, 0, 1, 2, \ldots$

***Decimal notation*, digit** (n.) Integer n, $n > 0$, is written as $d_k d_{k-1} \ldots d_1 d_0$ (a sequence of digits), where $n = \sum_{i=0}^{k} d_i * 10^i$ with $0 \le d_i < 10$ for $0 \le i < k$ and $0 < d_k < 10$.

The International System of Units (SI, [2]) allows the digits of numerical values having more than four digits to be separated by a thin, fixed space into groups of three, counting from the right (least significant) end, to improve readability. Neither dots nor commas are ever inserted in the spaces between groups.

Addition, subtraction, multiplication, division, exponentiation (n.) Add, subtract, multiply, divide, raise to a power (v.); sum, difference, product, quotient, power (n.). Binary operations on numbers: $a + b$, $a - b$, ab or $a * b$, a/b (for $b \neq 0$), a^b ($a \neq 0$ or $b > 0$).

N.B. Do not use the notation $a \times b$ for multiplication. 0^0 is TBD. Division and exponentiation (negative exponent) are not closed operations on integers. Integer division is TBD.

Negation (n.) Negate (v.). Unary operator on numbers: $-a$.

Equals, differs from, is less than, at most, greater than, at least Binary relations on numbers: $a = b, a \neq b, a < b, a \leq b, a > b, a \geq b$.

Positive, negative (adj. on numbers) Strictly greater than zero, strictly less than zero. N.B. In some languages (notably French), 'positive' and 'negative' include zero. When translating the English 'positive' and 'negative', it is then necessary to add 'strictly'.

Square (n.) A number that can be written as a^2 for some integer a. (Also see under Geometry.)

Divisible; divisor, multiple (relation; n.) Integer a is divisible by integer d, d is a divisor of a, a is a multiple of d, iff there exists an integer q such that $a = d * q$.

Even, odd (adj. on integers) Divisible, not divisible by two.

Modulo Binary operator on integers: $a \bmod b$ for $b > 0$; satisfies $0 \leq a \bmod b < b$ and $a - a \bmod b$ is a multiple of b.

Prime, composite number (n.) A prime is an integer greater than 1, having only itself and 1 as divisors. A composite number is an integer that can be written as the product of two integers, each greater than 1. (N.B. 1 is neither a prime nor a composite number. In a task description, this needs to be stated explicitly.)

Minimum, maximum Of a finite nonempty set of numbers: the smallest, largest number in the set. Notation min, max is TBD.

Fraction, numerator, denominator (n.) Number of the form $\frac{a}{b}$ with integer numerator a and positive integer denominator b.

Percentage (n.) Fraction expressed in units of $1\% = 1/100$.

Absolute value Function on numbers: $|a| = \max \{ -a, a \}$

2.2 Structures

Set, element of, (in)equality, subset, intersection, union, size $\{ a_0, \ldots, a_{k-1} \}$ is (denotes) the set consisting of the elements a_i for $0 \leq i < k$ ($k \geq 0$; order and multiplicity are irrelevant; $k = 0$ is allowed: empty set). Notations: $A = B$, $A \neq B$; other notations are TBD.

Interval Set of elements between given lower and upper bound. The task description must clarify which bounds are included. Notation TBD.

Tuple (n.) $(a_0, a_1, \ldots, a_{k-1})$ is the tuple of k elements a_i for $0 \le i < k$ ($k \ge 2$; order and multiplicity are relevant; only finite tuples). N.B. Tuples of zero elements and one element are TBA.

Sequence, length Notation must be explained, e.g. $a_0\, a_1\, \cdots\, a_{k-1}$ or $a_0, a_1, \ldots, a_{k-1}$ or $\langle a_0, a_1, \ldots, a_{k-1} \rangle$ is the sequence of length k, consisting of the elements a_i for $0 \le i < k$ ($k \ge 0$; order and multiplicity are relevant; only finite sequences).

N.B. Sequences of zero elements and one element are allowed, but the task description must be explicit about their treatment and must use appropriate notation (e.g. with open/close markers) to avoid confusion. Infinite sequences must be presented as a function. Do not use the notations $\{ a_0, \ldots, a_{k-1} \}$ and $(a_0, a_1, \ldots, a_{k-1})$ for sequences.

Function, domain, range, function application (n.) A function f from domain D to range R associates to each value in D a single value in R. The value that f associates to x is denoted by $f(x)$, and sometimes also by f_x; read as 'function f applied to argument x'. Applied to (a tuple of) multiple arguments: $f(x, \ldots, z)$. N.B. Notation $f : D \to R$ is TBA.

Lexicographic order; precede, succeed Relation on tuples of equal length and on sequences. Note that the task description must clarify whether the reflexive or irrelexive version is meant. Here are the irreflexive definitions.

On tuples of equal length: $(a_0, a_1, \ldots, a_{k-1})$ precedes $(b_0, b_1, \ldots, b_{k-1})$ iff there exists an index j ($0 \le j < k$) such that $a_i = b_i$ for all i with $0 \le i < j$, and $a_j < b_j$.

On sequences: $s = \langle a_0, a_1, \ldots, a_{k-1} \rangle$ precedes $t = \langle b_0, b_1, \ldots, b_{\ell-1} \rangle$ iff either $k < \ell$ and $a_i = b_i$ for all i with $0 \le i < k$ (i.e., $s \ne t$ is a prefix of t), or there exists an index j with $0 \le j < \min\{ k, \ell \}$ such that $a_i = b_i$ for all i with $0 \le i < j$, and $a_j < b_j$.

Permutation (n.) The task description must clarify whether a specific order of elements in a set is meant ('324 and 234 are two permutations of $\{ 2, 3, 4 \}$'), or a bijective function from a set to itself.

Sorted (adj. on sequences) The task description must explicitly state the direction and wheter the ordering is strict or not. Terminology like ascending, increasing, descending, decreasing is TBD.

Table, row, column, entry Rectangular layout of $M * N$ elements, called the entries, $a_{i,j}$ with $0 \le i < M$ and $0 \le j < N$, in M rows and N columns:

$$\begin{array}{cccc} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{array}$$

Rows are numbered from the top, columns from the left. Be careful with M or N equal zero or one.

Adjacent, consecutive, neighbor, ... Task description must disambiguate.

2.3 Geometry

Only in bounded regions.

Cartesian coordinates A tuple of numbers, one for each dimension; x, y (and z) coordinates; use orthonormal basis

Point 0-dimensional object located in K-dimensional space, given by tuple of K coordinates.

Line, line segment Straight 1-dimensional figure. Task description must explicitly state that a line segment includes its end points.

Plane Flat 2-dimensional figure.

Intersection (of geometric figures) Set of common points. Could be empty or contain more than one point.

Incidence Point on line; line through point.

Between A point w.r.t. two other points, all on the same line. The task description must clarify whether 'A lies between B and C' allows or excludes $A = B$ and $A = C$.

Angle Between two line segments with a common end point; between two intersecting lines; smallest amount of rotation needed to align the objects, measured in degrees (unsigned, $\leq 180°$). A right angle is $90°$. Beware of complications; also see [4]. Signed angles and the notation $\angle ABC$ are TBD. Radians and gradians are TBA.

Parallel, perpendicular (rel. on lines) non-intersecting or equal lines in the same plane, lines intersecting at $90°$. Notation $\ell \parallel m$, $\ell \perp m$ is TBD.

Polygon, vertex, side, area Closed planar figure with n sides and n vertices ($n \geq 3$). N.B. This is not just a sequence of points.

Triangle Polygon with three sides. Notation $\triangle ABC$ is TBD.

Rectangle, square Polygon with four sides and four right angles, opposite/all sides have equal length.

Rectangular block, cube, volume

Inside, outside Point w.r.t. 2D- or 3D-figure. The task description must be explicit about the treatment of the boundary.

Translation, reflection, rotation Operations on geometric figures. Translation in a given direction over a given distance. Reflection in a given line. Rotation around a given point by a given angle.

2.4 Miscellaneous

Boolean (n.) The type consisting of the values *false* and *true*.

Informal logic not, and, or, implies, for all, exists; if-then-else.

Subscript notation, indexing a_i

Algebraic expression Built from constants, variables, operators, (round) parentheses for grouping, function applications. N.B. Square brackets $[\cdots]$ and braces $\{\cdots\}$ are TBA for grouping; quantified expressions (involving dummies) are TBA.

Definition by recurrence relation, (mutually) recursive definition Definition of an object in terms of itself, or of multiple objects in terms of each other (and themselves). To qualify as basic, it must be easy to verify that it is well defined; that is, the base case(s) must be explicitly given and the recursive occurences must be obviously 'simpler'. For instance, the Fibonacci numbers F_i ($i \geq 0$) are defined by $F_i = i$ for $i \leq 1$ and $F_i = F_{i-2} + F_{i-1}$ for $i \geq 2$.

Character, string Task description must make explicit which characters play a role; usually it is best to restrict these to a subset of the (upper or lower case Latin alphabet); avoid non-printing characters, control characters (newline, carriage return, formfeed, tab) and blanks. May have to use multiple notations next to each other, in accordance with all approved programming languages. Compare to 'sequence': be careful with the empty string, and a string consisting of a single character (versus that single character by itself).

3 To-Be-Defined Concepts, Terminology, and Notations

The list in this section is not intended to be complete. It mentions some typical cases. Note that definitions must be short and easy to understand.

Natural number Nonnegative integer (includes zero).

Integer notation in bases other than ten No standardized notation.

Greatest common divisor, least common multiple

Floor, ceiling $\lfloor x \rfloor$, $\lceil x \rceil$

Set-related notations $\emptyset$ (empty set), $a \in S$ (membership), $A \subseteq B$ (subset), $A \cap B$ (intersection), $A \cup B$ (union)

(A)symmetric set difference, Cartesian set product Notation: $S - T$, $S \div T$, $S \times T$ N.B. $S \times T \times U$, $(S \times T) \times U$, and $S \times (T \times U)$ are TBA, because of the subtle differences.

Size of set, length of sequence Notation: e.g. $\#S$, or $|S|$

Mean, median, modus, variance, standard deviation Of sequence (or bag) of numbers.

Relation (n.) A relation R on set S is a subset of $S \times S$. Instead of $(s, t) \in R$, one often writes $s\ R\ t$, to express that s is related to t under R.

Partial order, total order, equivalence relation Special kinds of relations.

Reflexive, irreflexive, symmetric, antisymmetric, transitive (adj. on relations)

Injective, surjective, bijective (adj. on functions) N.B. The adjective 'one-to-one' (synonym for 'injective') is best avoided, because it can easily be confused with 'one-to-one correspondence' (synonym for 'bijective').

Polynomial, degree, coefficient Function of the form $f(x) = c_n x^n + \cdots + c_1 x + c_0$ with $c_n \neq 0$; degree is $n \geq 0$, coefficients are c_i. N.B. Do not confuse the formal expression and the function.

Bag (Also known as multiset) Collection of elements; order irrelevant, multiplicity relevant. No standardized notation. N.B.: $[a_0, a_1, \ldots, a_{n-1}]$ is a set in Pascal.

Often it may be easier to present it as a frequency function (but that also depends on the desired operations).

Operations and predicates on sequences Catenation, insertion, deletion, occurence in, indexing, subsequence, prefix, suffix, reverse, palindrome

Formal language, grammar Terminals, non-terminals, production rules; regular expressions

Automaton (Non)determinmistic finite state automaton. N.B. Definition of pushdown automaton or Turing machine could turn out to be too long.

Matrix, matrix multiplication Often easier to present as table.

Vector, inner product Often easier to present as tuple or sequence.

***Graph*, vertex, edge, path, connected, cycle, acyclic, tree** TBD, because these are unclear/ambiguous without further details: (un)directed, self-loops, multigraph, hypergraph, labels on vertices and/or edges, root?

Typically, graphs are described indirectly, using some kind of metaphor. However, care must be taken that the metaphor is not misleading. For example, a network of canals between cities suggests planarity, though this may not be the intention.

Factorial, binomial coefficient $n!$ defined by $0! = 1$ and $(n+1)! = n! * (n+1)$ for $n \geq 0$; $\binom{n}{k} = \frac{n!}{k!(n-k)!}$

Symmetry Transformation that leaves an object invariant.

Scalene, acute, obtuse, isosceles, equilateral, right Adjectives on triangles.

Collinear Three points are collinear if there exists a line containing them all

Convex (adj. on geometric figures) For every pair of points A, B inside the figure, the line segment with end points A, B lies entirely inside the figure.

4 To-Be-Avoided Concepts, Terminology, and Notations

Superscripts, nested subscripts $A^{(k)}$, A_{n_i}

Non-Latin letters Arabic, Cyrillic, Greek (α), Hebrew ($\aleph$), ... (Sorry :-) And also Gothic fonts, ...

N.B. Be reminded that this applies to the official English version only.

Formal quantified expression $\sum_{i=0}^{n} f(i) \quad \forall_{i \in S} P(i)$

Infinity As concept, also notations ∞, $\aleph$, ω

Algorithm But BK in solutions

Order notation $\mathcal{O}(f(n))$

BK in solutions, e.g. $(N \log N)$, incl. the terminology logarithmic, linear, quadratic, cubic, exponential

Algorithmic complexity, (un)decidable, NP-complete

Heuristic, approximation, randomization

Concurrent computation, process, thread

Boolean algebra, ... False, true, conjunction $\wedge$, disjunction $\vee$, negation $\neg$, implication $\Rightarrow$, equivalence $\equiv$, ... (consider promotion to BK in the future)

Formal logic, proposition, predicate, proof

Substitution

Information, entropy As measured in terms of bits.

Real numbers including square roots, logarithms, trigonometric function values (sin, cos, ...), $\pi \approx 3.14\cdots$, $\mathrm{e} \approx 2.71\cdots$

Floating-point number, fixed-point number [5]

Complex number Could be TBD with integer Re and Im; operations are then also TBD.

Generating function

Algebra, group, ring, field, ...

Linear algebra, vector space, determinant, ...

Polar coordinates

Circle, radius, diameter

Ellipse, parabola, hyperbola, focal points, cone

Asymptote, asymptotic

Numerical analysis, error [5]

Probability, random variable

Calculus, analysis Limit, real function, continuity, derivative, integral, differential equation, ...

5 Loose Ends

Physics concepts, units, ... cf. International System of Units (SI). The concepts of distance, length, time, speed, and mass may play a role, but no knowledge of physical laws may be assumed. Other physical concepts (such as accelleration, force, density, charge, ...) are best avoided.

Chemistry, biology, ... Avoid ...

Another loose end: It is also good to keep in mind that certain restrictions apply to organizers when they present (fragments of) program texts, such as e.g. interfaces of libraries.

Questions:

- Missing concepts, terminology, notations?
- Superfluous items?
- Misclassifications?
- Further references?

References

[1] ACM. *The ACM Computing Classification System*. http://www.acm.org/class/ 1998.

[2] CGPM. *Resolution 7 of the 9th General Conference on Weights and Measures*. 1948. Also see item #16 of http://physics.nist.gov/cuu/Units/rules.html.

[3] Edsger W. Dijkstra. "Why numbering should start at zero." EWD 831, August 1982. http://www.cs.utexas.edu/users/EWD/ewd08xx/EWD831.PDF Transcription: http://www.cs.utexas.edu/users/EWD/transcriptions/EWD08xx/EWD831.html

[4] Edsger W. Dijkstra. "On arcs and angles." EWD 1193, December 1994. http://www.cs.utexas.edu/users/EWD/ewd11xx/EWD1193.PDF

[5] Gyula Horváth, Tom Verhoeff. "Numerical Difficulties in Pre-University Informatics Education and Competitions", *Informatics in Education*, **2**(1):21-38 (2002).

[6] Eric W. Weisstein. *MathWorld* – A Wolfram Web Resource. http://mathworld.wolfram.com/