

NACLO Style Guide

Version 3

Contents

Introduction

1. Submission guidelines
2. Publication criteria
 - 2.1. Self-contained
 - 2.2. Exploratory
 - 2.3. Multi-level
3. Prior knowledge
 - 3.1. Language
 - 3.2. Math and computers
 - 3.3. Other knowledge
 - 3.4. Acceptable vocabulary
4. Acceptable languages
 - 4.1. Common languages
 - 4.2. Artificial languages, codes and ciphers
5. Other conventions
 - 5.1. Instructions and “flavor text”
 - 5.2. Layout and formatting
6. Sample recommendations

Introduction

“Linguistic Olympiad” problems, although similar to problem sets in courses or textbooks, differ in some important ways. Contests are very different from courses, and beyond that secondary-school students are a very specific audience. It is not necessarily difficult to set problems for them, but it requires some particular care; it is easy to overestimate what they know, and easy to underestimate what they can do.

The biggest difference between an “LO” problem and most course or textbook problems is that NACLO is not trying to impart *specific* lessons about language, languages, or language technologies. Among the goals of course or textbook is to have (ideally) all the students achieve a certain basic level of knowledge, so that they can move on to more advanced levels that build on that knowledge. NACLO cannot do this; we only have these students in our thrall for a few hours a year, and they *will* forget most of the details by the time the next year rolls around.

Instead we try to provide a certain kind of learning experience, one built around discovery rather than instruction. Our take-home lessons are the ones that follow from that experience: that languages are systematic and the rules are there for you to discover – they’re not just decrees from teachers and books – and there are many ways to discover language, study language, and think about language.

Few contestants will achieve every specific insight, and many will achieve only basic insights, and that’s entirely okay. That’s part of the fun – no one is handing them anything, so everything they achieve is something to be proud of.

This is the fundamental way in which NACLO problems differ from textbook problems. In a textbook, explaining “case” ahead of the case problem might be necessary, if it is important to later lessons. You don’t want anyone to miss a foundational lesson. In NACLO, explaining “case” might actually *undermine* our foundational lesson: it would take away the opportunity to discover something, and reinforce the idea that language facts need to be learned from on high. Not all the contestants will understand case at the end, and many that do will later forget, but the experience of discovery will stay with them.

This document is intended to serve as a “style guide” for problem writers, detailing questions that NACLO problem writers may have:

- What criteria do the program committee use when choosing problems for publication? (Section 2)
- What prior knowledge can I assume that contestants have? (Section 3)
- Are there any languages that I should avoid writing a problem about? (Section 4)
- What other conventions should I keep in mind? (Section 5)

1. Submission guidelines:

- Problems and solutions should be submitted together, but as separate documents. (We cannot verify the correctness, solvability, or appropriateness of problems when we do not know what answers, or what range of answers, the author accepts as correct.)
- Problems should be submitted as .DOC/.DOCX, .PDF, .TXT, or .RTF. If your problem depends on special fonts or an intricate layout, include a .PDF so that reviewers and editors can be sure they see it as you intended.
- If your problem uses a special font, include it in your submission.

2. Publication criteria

These are the criteria that the NACLO editors use to choose problems for publication. While not all published problems meet all these criteria ideally – sometimes strengths in one area can make up for weaknesses in another – these are the three criteria that inform almost all discussion of problem choice.

2.1. Self-contained

- The problem should not require outside knowledge to reach a solution. Outside knowledge might include knowledge of a theoretical concept, prior experience with a certain way of thinking, or real-world knowledge of culture or technology.
- Vocabulary beyond the terms listed in Section 3 should be defined in the problem statement.
- There should be enough data in the problem to rule out incorrect (“parasitic”) solutions.

2.2. Exploratory

Problems should allow the contestant to feel like they’ve discovered something through their own effort. Avoid leading the contestant by the hand to the solution, or explaining what’s supposed to be interesting. Consider whether facts given in the instructions could instead be insights that the contestant discovers in the data.

2.3. Multi-Level

Problems should allow a range of insights, with basic insights leading to more advanced ones. Problems shouldn’t just be simple analogy, nor should they be nearly impenetrable. Ideally, some problem tasks should be solvable with only basic insight into the problem, while others should require a much more sophisticated understanding. This way, contestants of various skill levels can come away with some satisfaction, and contest organizers get a gradient of scores to help choose winners.

3. Prior knowledge

In general, problems should avoid relying on prior knowledge, and the less a problem assumes about what a contestant knows, the stronger the problem will be. Nonetheless, all writing requires making some assumptions about who the audience is and what they know.

NACLO problems are intended to be appropriate for a 9th through 12th grade audience, although it is not uncommon for advanced 7th or 8th grade students to compete as well. Regarding what a U.S. student may or may not have learned before 9th grade, the *Common Core State Standards* are a useful reference.¹

3.1. Language

- You can generally assume that the contestant knows what an average adult knows about grammar, but shares their misconceptions. Linguistic vocabulary that is in common usage (“noun”, “vowel”, “sentence”, etc.) can be assumed, but also note that contestants will often have fuzzy and technically inaccurate versions of these. (See Section 3.4 for more detail.)
 - We can make few assumptions about what grammatical concepts contestants have explicitly been taught, or what they have taught about them; standards for grammar instruction in English education are based largely on *results* (clear and correct writing) rather than specific *concepts* the student must know.
 - Contestants will often understand that spelling differs from pronunciation, but may not have given this much thought. If a problem represents English phonetically, it may be helpful to point this out, although the contestant may be able to discover this for themselves from the data.
 - The idea that one variety of English is “correct”, and that others are sloppy or not rule-governed, is as common as it is in the wider culture. It can be helpful, when presenting data on non-standard English varieties, or English-derived pidgins and creoles, to emphasize that these are rule-based systems.
- You can assume fluency in English, but contestants will not necessarily be able to make native-quality judgments about grammaticality or contextual felicity.
 - The “intuitive method” could, potentially, play a role in a problem, but avoiding making it such that contestants must intuit the same judgments as the author to get the problem right. Even if a contestant does speak English natively, the ability to draw out

¹ Not all states have adopted these standards, not all schools are up to standard, and Canadian guidelines are somewhat different, but the standards can nonetheless serve as a rough guide to when different language concepts tend to be introduced.

We do not expect individual problem authors to have read these standards, but these are the education standards that most contestants will be under, and the standards that inform this guide.

one's own intuitions – including knowing how to separate genuine intuition from pedagogical preconceptions and social attitudes – is a kind of prior experience.

- Contestants will have a variety of levels of experience with other languages. They may or may not realize that other languages differ grammatically from English.
 - It is usually not necessary to *explicitly* disabuse them of these misconceptions. Many misconceptions can be dispelled through exploration and play with real language data.
- Cursive writing is not necessarily taught anymore, or may be taught in elementary school but not maintained into high school. Contestants may not recognize the more divergent letterforms or be able to distinguish similar letters like *m* and *n*.

3.2. Math and computers

- Math up to (roughly) basic algebra can be assumed. The average 9th grader² has likely been introduced to the concept of the function, but not extensively, and probably not applied to anything but numbers. Familiarity with formal logic cannot be assumed.
- Contestants have, by 9th grade, probably seen very basic probability and statistics (e.g., recognizing what a scatter plot represents, and drawing informal conclusions about it, but may not have done anything more formal than this).
- Many contestants will not have had any computer science or programming background, and may not know much about how computers operate.
 - Problems in which contestants “program”, “debug”, or otherwise reason about abstract computers usually need to be very explicit about what these computers can do – what operations they can perform, how instructions are interpreted, etc.

3.3. Other knowledge

- You can assume a basic level of knowledge about the world. It is safe to assume they know that grapes are a kind of fruit and London is in England. It is probably not safe to assume they know cucumbers are a fruit, or can name world capitals in general.
 - Occasionally problems involve a trivia component – say, knowing the answers will be the names of world capitals, but not knowing which ones. This is potentially ok, depending on the structure of the problem. If it is an incidental question at the end, with no other discoveries depending on it, it is potentially alright. If the answer can be narrowed down without having heard of the item in question, it is probably alright. If the item is

² Many of our actual contestants are somewhat ahead-of-grade in math, sometimes dramatically so, but many others are not.

somewhat obscure and knowing it is a prerequisite to solving the central mystery, the problem should probably be restructured.

- Problems should avoid assuming familiarity with U.S. or Canadian culture specifically, if this knowledge is necessary for a solution. Contestants come from a variety of backgrounds and vary in how familiar they are with pop and traditional culture.

3.4. Vocabulary

Many grammatical terms are in common English usage, and you can assume that contestants have some familiarity with them, but also note that they may have only vague understandings of these, or share popular misconceptions. They will also not necessarily recognize when a common English word like “object”, “argument”, “function”, or “character” is being used in a specialized sense.

Below, saying a term is “acceptable” means that it can be used without further explanation, but if a specific technical definition of this term is intended, it should be introduced. (However, “acceptable” does not mean that using it is better than not using it. Not using technical vocabulary at all is usually the best choice.) Saying a term should be “introduced” means that it should be accompanied by a paraphrase, an explanation, or relevant examples.

- “Noun”, “verb”, and “adjective” are acceptable, but contestants will often think of these in loose semantic terms like “A verb is an action word”. If you require a more technical notion of word class, it should be introduced.
 - Further categories, like “preposition” or “conjunction”, will be familiar but should be introduced by example.
 - “Part of speech” is acceptable, but if you need a particular interpretation of what these are it should be introduced.
- “Singular” and “plural” are probably acceptable. “Gender” and “number” in technical senses should be introduced unless it is clear enough from context.
- “Word” and “sentence” are acceptable, although note that they will understand this in an orthographic sense. “Question” and “command” are acceptable. “Phrase” and “clause” are likely concepts they have encountered, but we nonetheless recommend introducing them; contestants may not have in mind the exact definition the author intends.
- The terms “subject”, “direct object”, “indirect object”, “predicate”, “active voice” and “passive voice” are likely to be familiar, but not necessarily understood. If necessary to the problem they should be introduced in greater detail.
 - Take care with the use of “object” on its own, since this can be understood non-technically.
- “Prefix” and “suffix” are likely to be familiar but if it is not in an environment where context makes it clear, introduce these.
- “Tense”, and “past tense” and “future tense”, will be familiar. They are probably familiar with the term “perfect” but this should be introduced if needed.
- “1st person”, “2nd person”, etc. are likely to be familiar terms, but these terms can often be avoided in any case.

- “Consonant” and “vowel” are acceptable; again note that they will often understand this orthographically, as in “The vowels are A, E, I, O, U, and sometimes Y.”
 - It often happens that the vowels in the problem language are orthographic vowels in English or obvious variants thereof, in which case it may be acceptable to refer to them as “vowels” without further explanation, and likewise for consonants. If not, “vowel” and “consonant” should be specified more exactly.
- Contestants are probably familiar with “syllable”, but note that they will not necessarily syllabify words or count syllables exactly the way a linguist would.
- “Grammar” in its technical sense should be introduced; “grammatical” and “ungrammatical” may or may not be acceptable depending on the context.
- Speaking of words as being “borrowed” or “loaned” from another language is acceptable.
- It is generally acceptable to speak of languages being “related” to or “descended” from other languages, or occurring in “families”, but if the problem depends on this notion make clear what this means.
- Speaking of English letter forms as the “Roman alphabet” is usually acceptable; even if contestants are unfamiliar with this usage it is often clear in context.
- “Transcribe” and “transcription” may be familiar, and “transliteration” and “orthography” are often unfamiliar, but in any case these ideas can often be rendered using more everyday language.
- “Parse”, “parser”, etc. should be introduced.
- “Ambiguous”, “ambiguity”, etc. should be introduced or avoided.
- “String” to mean a sequence of characters should be introduced, or simply avoided. “Character” should likewise be introduced or avoided.
- “Argument” in any of its uses should be introduced, since this can mean many different things. “Function” in reference to a function from numbers to numbers is probably acceptable if the context makes it clear, but other usages should be introduced. The construction “function from ... to ...” is probably unfamiliar.

Vocabulary beyond these terms should be introduced.

4. Acceptable languages

4.1. Common languages

NACLO is unlikely to publish problems on commonly-spoken and commonly-learned languages, although it depends on the type of problem. In general, NACLO avoids publishing problems on languages that are commonly offered as foreign language electives in secondary school, languages that many contestants speak, and languages commonly learned in religious instruction. (This list includes but is not limited to Latin, French, Spanish, German, Russian, Arabic, Hebrew, Japanese, Mandarin, and Korean.) A problem on a language closely related to these, or a language that uses the same basic writing system, might be rejected if it can be solved by knowing the related language.

The goal is not to “punish” contestants who know these languages,³ just to ensure that no large demographic of contestants has an immediate advantage. Part of the goal for each round is to identify the elite solvers who should continue to the next round, and this identification is made difficult if a significant number of contestants can solve a problem automatically.⁴

It should be noted, however, that this requirement mostly applies to problems where knowing the language in question gives solvers a significant advantage. For example, if the goal of a puzzle on French is to figure out what phrases like “*le plume de ma tante*” mean, knowing French makes this problem trivial. On the other hand, IOL2003 posed a problem in which contestants tried to deduce the conditions under which the French prefix “re-” gets an acute accent. This is not something that a speaker of French would know automatically, the way they know what “*la plume de ma tante*” means, and such a problem would not necessarily be rejected just because it concerns French.

Note also that other English-language linguistic olympiads have different demographics of contestants, and may be happy to use problems that NACLO itself cannot.

4.2. Artificial languages, codes, and ciphers

Although they are not forbidden, NACLO editors usually do not choose problems on constructed languages. In general, if a problem is meant to illustrate that a particular interesting phenomenon occurs in human language, it is best to use a natural human language to illustrate it.

There are a few exceptions where constructed language problems may be appropriate:

- If the constructed language is historically noteworthy, editors are more likely to accept the problem, especially if the puzzle emphasizes the feature or phenomenon that makes it historically noteworthy.
- Sometimes, in problems meant to illustrate computational or formal approaches, real language data may be too messy or too complicated. It *might* be appropriate, in such problems, to illustrate the algorithm or formalism using a constructed language.
- Problems in which contestants must deduce the properties of unknown words usually use made-up words, for obvious reasons.

³ Based on our conversations with contestants, they do not *want* unfair advantages, and even express disappointment when they feel their achievement was not entirely earned.

⁴ We cannot erase every advantage – sometimes contestants do speak languages like Basque – but one contestant speaking Basque does not throw off the score distribution for an entire contest.

Codes and ciphers are potentially acceptable topics for problems, although they are less likely to be published than problems on natural languages. Like other artificial communication systems, a code or cipher is more likely to be published if the system is historically significant and the problem emphasizes why it is noteworthy, or if it illustrates a scientifically or historically noteworthy idea.

5. Other conventions

5.1. Instructions and “flavor text”

- In problems that present languages, one of the first paragraphs (usually the first) should be a short description of where and by whom it is or was spoken.
- It is good to indicate, somewhere in the problem, how particular characters are pronounced, even if this is not necessary to solve the problem. Use non-technical language (e.g., “is pronounced like the ‘th’ in ‘there’” rather than “represents the voiced interdental fricative”). If there is no satisfactory way to do this, it is also acceptable to simply say “X represents a sound that does not occur in English.”
- If the problem presents a computational method or formalism in a clever disguise, reveal this somewhere in the problem, even in a footnote.
- Try to avoid having more than one page of background material before getting to problem data or a task.
- For language names, use whatever name the speech community would consider respectful, if they have a preference. When there is no preference, or mixed preferences, use your best judgment. If you do not know, err towards using the speech community’s own name for the language and introducing the English name (if applicable) parenthetically. Avoid making puns or jokes based on the names of cultures, nations, or languages, if the group in question would find it disrespectful.
- Avoid representing languages with violent scenarios. Although it was once common for linguists to illustrate the prototypical transitive sentence with, say, “The man hit the boy”, many speech communities no longer consider this acceptable.
- Avoid referring to contestants as “kids” or “children”, or referring to people older than them as “adults” or “grown-ups”.

5.2. Problem structure and layout

- When doing “matching” tasks where contestants have to match up translations, fill-in-the-blanks tasks with word banks, or other similar tasks, our usual convention is to have the contestant enter capital letters (rather than numbers, lowercase letters, or whole words/phrases).
- Meanwhile, tasks and subtasks are usually numbered. Label tasks Q1, Q2, etc. and label their subtasks 1, 2, etc. (The “Q” will be changed to the letter of the problem in the final booklet.)

- In general, lean towards using letters rather than numbers to represent abstract forms. For example, if you need to express a sequence of symbols, lean towards “ABAACBB” rather than “1211322”.
- Try to keep data together, when it is of the same type, rather than breaking it up between questions. (That is, avoid having each question accompanied by the exact data that solves it.)
- If possible, avoid using linguistic labels (1st person, plural, dative, etc.) to describe pieces of data; instead, try to accompany each piece of data with a rough translation (“I walk”, “dogs”, “to the horse”, etc.).
- Rather than asking for specific pieces of knowledge (e.g., “How does this language turn nouns into verbs?”), it is usually better to ask contestants to do something concrete that requires that knowledge (e.g., asking them how you would say “to make a cake”) and then simply ask, at the end, “Explain your answers.” The effect is roughly the same, but the latter technique often assumes less about the contestant’s prior knowledge state.
 - Also, it makes it easier to adapt the problem to contests (like the NACLO Open Round) that rely on autograding and cannot easily handle prose answers.

6. Example Recommendations:

In the instructions: “‘Stemming’ refers to the process of removing prefixes and suffixes from a word until only its root remains.” Do I need to define “prefix”, “suffix”, and “root”?

Recommendation: Defining by example may be appropriate here, to remind contestants what these words refer to, but with examples these words will probably be clear from context.

In the instructions: “Note that this language represents subjects as suffixes rather than as separate words.” Do I need to define “subject” and “suffix”?

Recommendation: It may be better to avoid pointing this out in the first place, and as such avoid the problem of defining “subject” and “suffix”. If this insight can be discovered in the data, it is better to let the contestants discover it for themselves.

In the instructions: “The Dalmatian language, now extinct, was descended from Latin and was closely related to Italian.” Do I need to define “descended”, and “related”?

Recommendation: It is probably not necessary to go into further detail here; even if contestants do not know exactly what these words mean, it is probably not required for them to reach a solution.

In the instructions: “Below are sample texts from three languages. Which two are more closely related and why?” Do I need to define “related”?

Recommendation: It will probably be necessary to define “related” further here, as the contestant’s interpretation of this term will affect how they go about solving this.

In the instructions: “‘sh’, ‘ch’, ‘ts’, and ‘ng’ represent single sounds.

Recommendation: This kind of hint is usually appropriate, but might also be rephrased as a solution task if adequately discoverable from the data. (That is, asking the contestant which sequences of letters represent single sounds, or asserting that “sh” represents a single sound and asking what other sequences represent single sounds.) It depends on the problem and what the problem is about.

In the instructions: “How would a computer go about doing this task?”

Recommendation: This question is probably not appropriate. It assumes a body of shared knowledge between author and contestant: what computers are capable of, what operations are basic vs. derived, what level of abstraction the author expects in the answer, etc. Even if the contestant is familiar with computers, their assumptions may differ significantly from the author’s. Questions like this should set up a relatively explicit model of computation that the contestant can experiment with.