SE 2AA4, CS 2ME3 (Introduction to Software Development)

Winter 2018

24 Specification Quality

Dr. Spencer Smith

Faculty of Engineering, McMaster University

March 8, 2018

24 Specification Quality

- Administrative details
- Line Formatter Example
- Converting English to Mathematics

Administrative Details

- A3
  - Part 1 - Specification: due 11:59 pm Mar 12
  - Part 2 - Code: due 11:59 pm Mar 26
- A4
  - Your own design and specification
  - Due April 9 at 11:59 pm

Line Formatter Overview

- The line formatter specification was written to improve on existing specifications
- How well does the specification do with respect to the following qualities: abstract, correct, unambiguous, complete, consistent and verifiable?
- For a requirement specification like that given, what are the advantages and disadvantages of maintaining both a formal specification and a natural language specification?
- A relevant critique in: Meyer (1985) "On Formalism in Specification"

Lorem BL ipsum BL NL dolor BLBLBL sit BL amet, BL
consectetur.ET

Lorem ipsum dolor sit
amet, consectetur.ET

Lorem ipsum
dolor sit
amet,
consectetur.ET

Lorem
ipsum
dolor
sit
amet,
consectetur.ET

Line Formatter

- Input stream signalled with ET
- Exactly one ET character in each input stream
- Character classifications:
  - Break character - BL (blank) and NL (new line)
  - Non Break Character - all others except ET
  - End of text indicator ET
- Word is a non-empty sequence of non break characters
- Break is a sequence of one or more break characters
- Output same sequence of words, except if there is an oversize word
  - Oversize means more than MAXPOS characters, where MAXPOS is a positive integer
  - If there is an oversize word
    - Set Alarm to TRUE
    - Exit the program

Line Formatter

- Up to the point of an error, the program's output should have the following properties
  - A new line should start only between words and at the beginning of the output text, if any
  - A break in the input is reduced to a single break character in the output
  - As many words as possible should be placed on each line (i.e. between successive NL characters)
  - No line may contain more than MAXPOS characters (words and BLs)

Abstract?

- Input stream signalled with ET
- Exactly one ET character in each input stream
- Character classifications:
  - Break character - BL (blank) and NL (new line)
  - Non Break Character - all others except ET
  - End of text indicator ET
- Word is a non-empty sequence of non break characters
- Break is a sequence of one or more break characters
- Output same sequence of words, except if there is an oversize word
  - Oversize means more than MAXPOS characters, where MAXPOS is a positive integer
  - If there is an oversize word
    - Set Alarm to TRUE
    - Exit the program

Not Abstract

- Specifies an implementation for error handling (variable named Alarm)
- Do not have to name the variable Alarm
- Could use exception handling (or another approach) instead
- ET is a machine dependent (program domain) concept - abstractly, the input is simply a finite sequence of characters

Correct?

- Up to the point of an error, the program's output should have the following properties
  - A new line should start only between words and at the beginning of the output text, if any
  - A break in the input is reduced to a single break character in the output
  - As many words as possible should be placed on each line (i.e. between successive NL characters)
  - No line may contain more than MAXPOS characters (words and BLs)

Incorrect!

- A line is defined as being between NLs, which ignores text before the first NL and after the last NL
- The output file does not contain ET, which is either a bug in the spec or a significant non-uniformity

Unambiguous?

- Up to the point of an error, the program's output should have the following properties
  - A new line should start only between words and at the beginning of the output text, if any
  - A break in the input is reduced to a single break character in the output
  - As many words as possible should be placed on each line (i.e. between successive NL characters)
  - No line may contain more than MAXPOS characters (words and BLs)

Ambiguous!

- "output text, if any"
- "point of error" is not defined
- Output matches input to the last acceptable word, or the last acceptable character?
- "trailing blanks ending with ET" is ambiguous (trailing means at the end!)
- The program's output should be the same sequence of words as in the input
  - But the input is not a sequence of words
  - If the input were a sequence of words, what about leading or trailing breaks?
  - "As many words as possible should be placed on each line"
    - WHO WHAT "NL" WHEN
    - WHO "NL" WHAT WHEN

Complete?

- Input stream signalled with ET
- Exactly one ET character in each input stream
- Character classifications:
  - Break character - BL (blank) and NL (new line)
  - Non Break Character - all others except ET
  - End of text indicator ET
- Word is a non-empty sequence of non break characters
- Break is a sequence of one or more break characters
- Output same sequence of words, except if there is an oversize word
  - Oversize means more than MAXPOS characters, where MAXPOS is a positive integer
  - If there is an oversize word
    - Set Alarm to TRUE
    - Exit the program

Not Complete

- Meaning of NL and its relation to the concept of line is left implicit - seem to expect that NL means carriage return on output, but they do not say this
- Alarm is not specified if MAXPOS is never exceeded

Consistent?

- Input stream signalled with ET
- Exactly one ET character in each input stream
- Character classifications:
  - Break character - BL (blank) and NL (new line)
  - Non Break Character - all others except ET
  - End of text indicator ET
- Word is a non-empty sequence of non break characters
- Break is a sequence of one or more break characters
- Output same sequence of words, except if there is an oversize word
  - Oversize means more than MAXPOS characters, where MAXPOS is a positive integer
  - If there is an oversize word
    - Set Alarm to TRUE
    - Exit the program

Not Consistent

- Not consistent!
- "non-empty" and "one or more" (synonyms)
- "stream" and "sequence" (synonyms)
- Is the input a "stream of characters" or a "sequence of words separated by breaks"? – sequence of T is not the same as sequence of sequence of T

Verifiable?

Verifiable?

- The specification cannot be verified, since it is ambiguous and incorrect

Advantages and Disadvantages?

- Advantages and disadvantages of maintaining both formal and a natural language spec?

Advantages and Disadvantages?

- Advantages and disadvantages of maintaining both formal and a natural language spec?

- Advantage of natural language - understandability
- Advantage of formal spec
  - Unambiguous
  - Highlights difficult to informally detect cases
  - Checking for completeness and consistency
  - Amenable to tool support
- Advantage of both - all of the above advantages
- Disadvantages - have to maintain two specs
- Automatic translation
  - Formal spec to natural language has been researched
  - Natural language to formal spec has received more attention

English and Mathematics as Languages

See Baber2002

- English is a language
- So is Mathematics
- Both have
  - Rules of grammar (syntax)
  - Semantics
- When writing in any language, pay attention to grammar and semantics. Get both right.

English and Mathematics: A Difference

- In English and other natural languages
  - Ambiguity desired, intentionally possible
  - Unambiguous statements almost impossible
- In Mathematics
  - Ambiguity not desired, intentionally prevented
  - Ambiguous statements almost impossible (even in probability theory, fuzzy logic)

Mathematics and Engineering

- Therefore, mathematics is the language of engineering