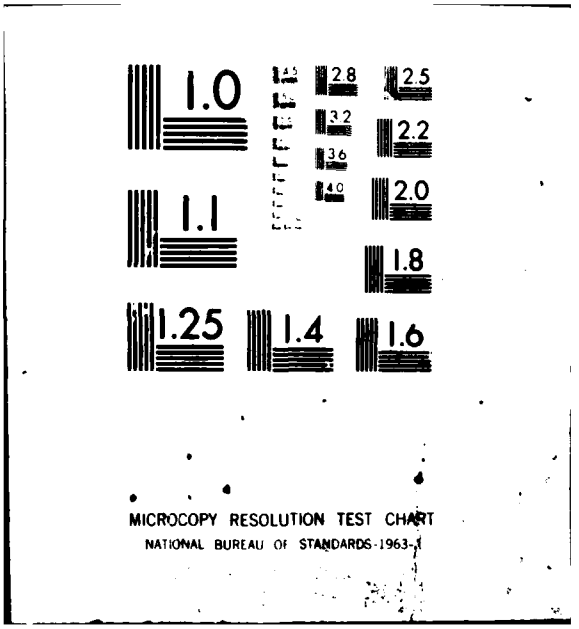
AD-A084 018 ARMY SATELLITE COMMUNICATIONS AGENCY FORT MONMOUTH NJ F/G 9/2
PROJECT ARIES. SDL -- A STRING DESCRIPTION LANGUAGE.(U)
APR 80 R L CONN

UNCLASSIFIED NL


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO. AD-A084 018

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
PROJECT ARIES
SDL -- A String Description Language

5. TYPE OF REPORT & PERIOD COVERED
Final
5/79 to 5/80

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Richard L. Conn

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
US Army Satellite Communications Agency
USA CORADCOM, Attn: DRCPM-SC-4G
Ft Monmouth, NJ 07703

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
Elt: 6.11.01.A
Proj: 1L1 61101 A91A
Task: 33 Work Unit: 131

11. CONTROLLING OFFICE NAME AND ADDRESS
US Army Satellite Communications Agency
USA CORADCOM, Attn: DRCPM-SC-4G
Ft Monmouth, NJ 07703

12. REPORT DATE
18 Apr 80

13. NUMBER OF PAGES
9

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

LEVEL II

ADA084018

16. DISTRIBUTION STATEMENT (of this Report)

Distribution Unlimited

1L161101A91A

This document has been approved for public release and sale; its distribution is unlimited.

DTIC ELECTE MAY 9 1980 S C D

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Distribution Unlimited

18. SUPPLEMENTARY NOTES

Final Rept May 79 - May 80

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

BNF, Extended BNF, Backus-Naur Form, SDL, String Description Language, ASCII, Context-Free Languages

80 5 7 008

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

SDL (String Description Language) is a recursive language which is used to define the structures of strings of ASCII characters. It permits implicit and explicit definition of strings, and it can be generally used to describe almost any language or set of strings. Derived from BNF and Extended BNF, it contains many features from these languages as well as several extensions made by the author. SDL may be used to describe context free languages.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified 041000

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DTIC FILE COPY

SDL -- A String Description Language

by

Richard L Conn

USA Satellite Communications Agency
Automated Systems Division, Programs Directorate
Fort Monmouth, NJ 07703

18 April 1980

SDL -- A String Description Language

Contents

SDL -- A String Description Language

The Rationale for SDL

String Description Language, hereafter referred to as SDL, is a language which enables one to describe the format of character strings. It gives one the ability to precisely, or, if desired, imprecisely, describe a generalized character string. For example, if the user wishes to describe all character strings which consist of an arbitrary number (from 1 to n) of 'A's followed by an arbitrary number of 'B's, he may describe this precisely in SDL as follows:

```
<ABSTRING> : 'AB' ! 'A'* <ABSTRING> 'B'*
```

As the user can see, this is a rather elegant and simple description of such character strings, while the English words used to describe such strings are somewhat awkward.

SDL was developed by the author as a result of his lack of satisfaction with other string description languages such as BNF (Backus-Naur Form) and Extended BNF. SDL is a combination of what the author considers to be the good points of several string description languages and the features he finds lacking in them.

Particularly, the features carried over into SDL from other string description languages include the relatively simple form of the Extended BNF statements, the BNF and Extended BNF operators (with some modification), the form of constant strings employed by Extended BNF, and recursion.

Features incorporated into SDL which are not found in BNF and Extended BNF include the multiplicity form of the '*' and '+' operators, the use of SDL-defined non-terminal variables, and the ability to imprecisely define strings using an English description.

SDL, as described in this document, is a refinement of the SDL described in the author's MS thesis "ARIAN -- An Implementation of a Microcomputer Operating System."

SDL Variables and Constants

All constants in SDL are string constants, which take the form of zero or more characters of the ASCII character set enclosed in single quotes. The double single quote ('') is used to represent a single quote constant. Examples of SDL constants are 'abc', 'hello', ' ', '0', '1234ABC', and '*:bc1,.'.

A variable in SDL is a symbol which may be assigned a string value. Two types of variables are used in SDL -- terminal and non-terminal variables. A non-terminal variable is a string of alphanumeric characters enclosed in angle brackets (<>), and it is used to represent certain special characters (such as carriage return) and characters or strings the user wishes to use to help define

other non-terminal and terminal variables in his SDL equations. A terminal variable is a string of alphanumeric characters (the first of which must be alphabetic) which is not enclosed in angle brackets. Terminal variables are final products; they may not be used to help define other variables. Non-terminal variables may be used to help describe themselves in their SDL expression. This is a recursive definition. The first example with <ABSTRING> is a recursive definition.

The underscore (_) can be used in variable names to improve readability; if used, its location in the variable name is as significant as the location of any other character in the variable name.

SDL defines a number of non-terminal variables which may be used in SDL equations. These non-terminals are:

1. <cr> or <CR> -- the ASCII carriage return character,
2. <bs> or <BS> -- the ASCII backspace character,
3. <lf> or <LF> -- the ASCII line feed character,
4. <null> or <NULL> -- the ASCII null character,
5. <empty> or <EMPTY> -- the empty or null string,
6. <bel> or <BEL> -- the ASCII BEL character (ctrl-G),
7. <ff> or <FF> -- the ASCII form feed character,
8. <tab> or <TAB> -- the ASCII tab character.

These SDL-defined non-terminal variables may be used in any SDL expression without being defined elsewhere. These symbols may also be redefined by the user at his discretion, but care must be taken not to think of their original SDL definitions if the redefinition of the variable assigns it a meaning different from its SDL meaning.

SDL Operators

SDL allows the user to employ several operators in the SDL expressions he creates. These operators are:

Symbol	Name of Operator
"	double quote
*	asterisk
+	the plus symbol
!	the exclamation mark
?	the question mark
...	the mark of ellipsis
(	the left parenthesis
)	the right parenthesis
:	colon

The uses of these SDL operators are as follows:

1. double quote - this is used to enclose a description phrase. A description phrase is an English phrase used to describe a variable; an example of a description phrase is "THE DIGITS 0 AND 1".

2. asterisk - this is one of SDL's two multiplication symbols. It means that the preceeding symbol or expression may be repeated zero or more times. For example, <digit>* means that <digit> may be repeated zero or more times. If the asterisk is followed by a number, the number specifies the upper limit of the multiplication. For example, <digit>*4 says that there may be zero to 4 digits in the string.

3. the plus symbol - this is the other multiplication symbol. It is used exactly like the asterisk, but it specifies one or more. The number following it may also be present, indicating an upper limit. For example, <digit>+4 says that there may be from 1 to 4 <digit>s in the string.

4. the exclamation mark - this is the logical OR operator in SDL. Used in expressions, it means that the right-hand side or the left-hand side are possible. For example, <alpha> ! <digit> says that whatever is being discussed may be an <alpha> or a <digit>.

5. question mark - this means that the preceeding symbol or expression is optional. For example, <alpha>? means that an <alpha> is optional.

6. the marks of ellipsis - implies that the symbols or characters are continuous by ASCII order. For example, 'A' ! ... ! 'F' refers to 'A', 'B', 'C', 'D', 'E', 'F'.

7. parentheses - used as grouping symbols. They may be nested as deeply as desired.

8. colon - this is the assignment symbol. It must be present in every SDL statement, and it assigns the expression on its right to the variable on its left.

SDL Statements

An SDL description of a string or set of strings consists of a group of SDL statements. An SDL statement consists simply of a variable, which may be used to represent the string in question, followed by a colon and an SDL expression. An SDL expression is a grouping of variables, operators, description phrases, and ellipses which describes a string. In this grouping each variable is separated by one or more delimiters, where a delimiter is an operator, a space, or a new line.

Examples

The following is a group of examples of SDL statements:

1. <digit> : '0' ! ... ! '9'

   This statement defines <digit> to be any of the characters '0' to '9'.

2. <hexdigit> : <digit> ! 'A' ! ... ! 'F'

   This statement defines <hexdigit> to be a <digit> or one of the characters 'A' to 'F'.

3. HEX_ADDRESS : <hexdigit>*4

   HEX_ADDRESS is defined as a string of from 0 to 4 <hexdigit>s. For example, HEX_ADDRESS may be "", "ABC", "0123", "00".

4. NUMBER : <digit>+4

   NUMBER is defined as a string of from 1 to 4 <digit>s. Note that NUMBER and HEX_ADDRESS are terminal variables and may not be used in expressions other than those like " VALUE

: TERMINAL ".

5. ITIS : <alpha>+ '.'?

ITIS is defined as a string of one or more <alpha>s followed optionally by a period.

6. ENGLISH_SENTENCE : "ANY COMBINATION OF ASCII CHARACTERS"

ENGLISH_SENTENCE is defined by a description phrase.

7. THING : <digit>+4 ! <hexdigit>+4

THING is defined as being either from 1 to 4 <digit>s or from 1 to 4 <hexdigit>s.

8. <neat> : 'A' ! 'B' ! 'A' <neat> 'B'

<neat> is recursively defined. Strings like 'A', 'B', 'AAB', 'AAAABBB', and 'ABBBBB' are possible.

The user will note that spaces are never implied in SDL. They must be placed in the string explicitly.

SDL -- A String Description Language

Summary

SDL is a recursive String Description Language which is used to define strings of ASCII characters. It permits implicit and explicit definition of strings, and it can be generally used to describe almost any language or set of strings. Derived from BNF and Extended BNF, it contains many features from these languages as well as several extensions made by the author.

Some features of SDL are:

1. SDL-defined non-terminal variables which need not be defined by the user.

2. The use of ellipsis to indicate items left out of a definition.

3. Multiplication of symbols or expressions with * or + may be followed by a number to indicate the limit of multiplication.

4. Recursive definition which allows a variable to be defined in terms of itself.

5. The use of a description phrase in a variable definition.

Like BNF and Extended BNF, SDL may be used to describe context free languages. It was not intended to be used as such, and SDL lacks some of the formality of BNF and Extended BNF, but the value of this application is present. SDL was designed primarily to be used as its name implies -- to describe character strings, and, as the user may note, a computer language can be referred to in terms of character strings.

References

Conn, Richard. "ARIAN -- An Implementation of a Microcomputer Operating System." May, 1978; MS Thesis, Graduate College, University of Illinois, Urbana, IL 61801. 187 pp. Also available through Defense Documentation Center, Alexandria, VA, under AD Number ADA060210.

Conn, Richard. "The Monitor Command System User's Manual." Feb, 1978; Department of Computer Science, University of Illinois, Urbana, IL 61801; Technical Report Number UIUCDCS-R-78-912 and Engineering Report Number UILU-ENG 78 1705. 31 pp.