

(12) UK Patent Application (19) GB (11) 2 162 347 A

(43) Application published 29 Jan 1986

(21) Application No 8516615

(22) Date of filing 1 Jul 1985

(30) Priority data

(31) 8419071 (32) 26 Jul 1984 (33) GB

(71) Applicant

The Secretary of Defence (United Kingdom),
Whitehall, London SW1A 2HB

(72) Inventor

Richard Anthony Evans

(74) Agent and/or Address for Service

Dr A W S Williams,
Procurement Executive, Ministry of Defence, Patents
1A4 Room 2014, Empress State Building, Lillie Road,
London SW6 1TR

(51) INT CL⁴

G06F 15/20

(52) Domestic classification

G4A CM UX

(56) Documents cited

None

(58) Field of search

G4A

(54) Language recognition apparatus

(57) Language recognition apparatus includes a cell (10) for comparing (14) input grammatical rule characters with a stored rule symbol (16) synchronously with comparison (20) of an input character with a second stored symbol (22). An AND gate (24) indicates simultaneous match, and effects output (38) of a third stored symbol (34) for comparison with the first stored symbol (16) synchronously with input of a second character for matching with the second symbol (22). A fourth stored symbol (42) is transferred (50) to a remote store for subsequent recall and comparison with the first stored symbol (16) in synchronism with third character input provided prior matches have occurred. The cell (10) tests successive input characters for compliance with a rule of grammar expressed by the stored symbols. Compliance is indicated by unrealisable recall of a symbol from the store in combination with lack of an input character. Several cells may be arranged to test for compliance with all rules of a grammar.

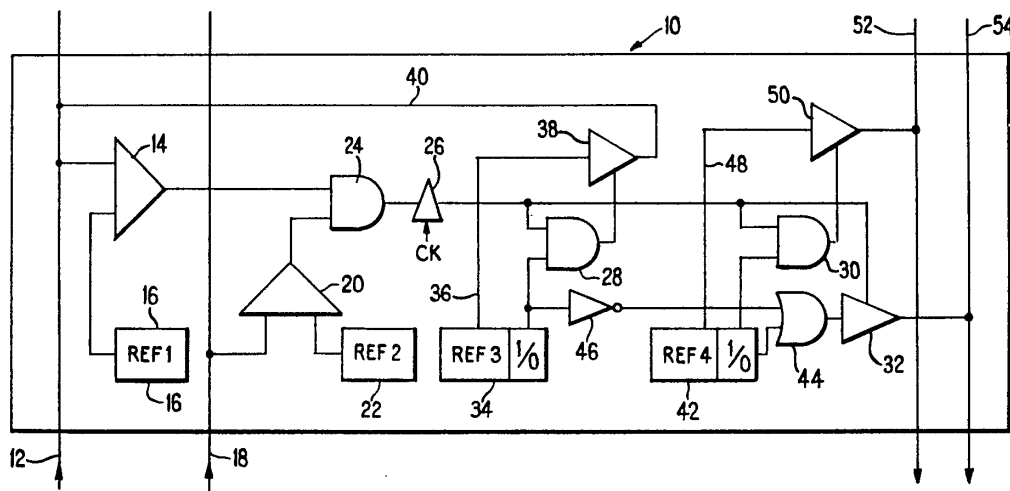


FIG. 1

GB 2 162 347 A



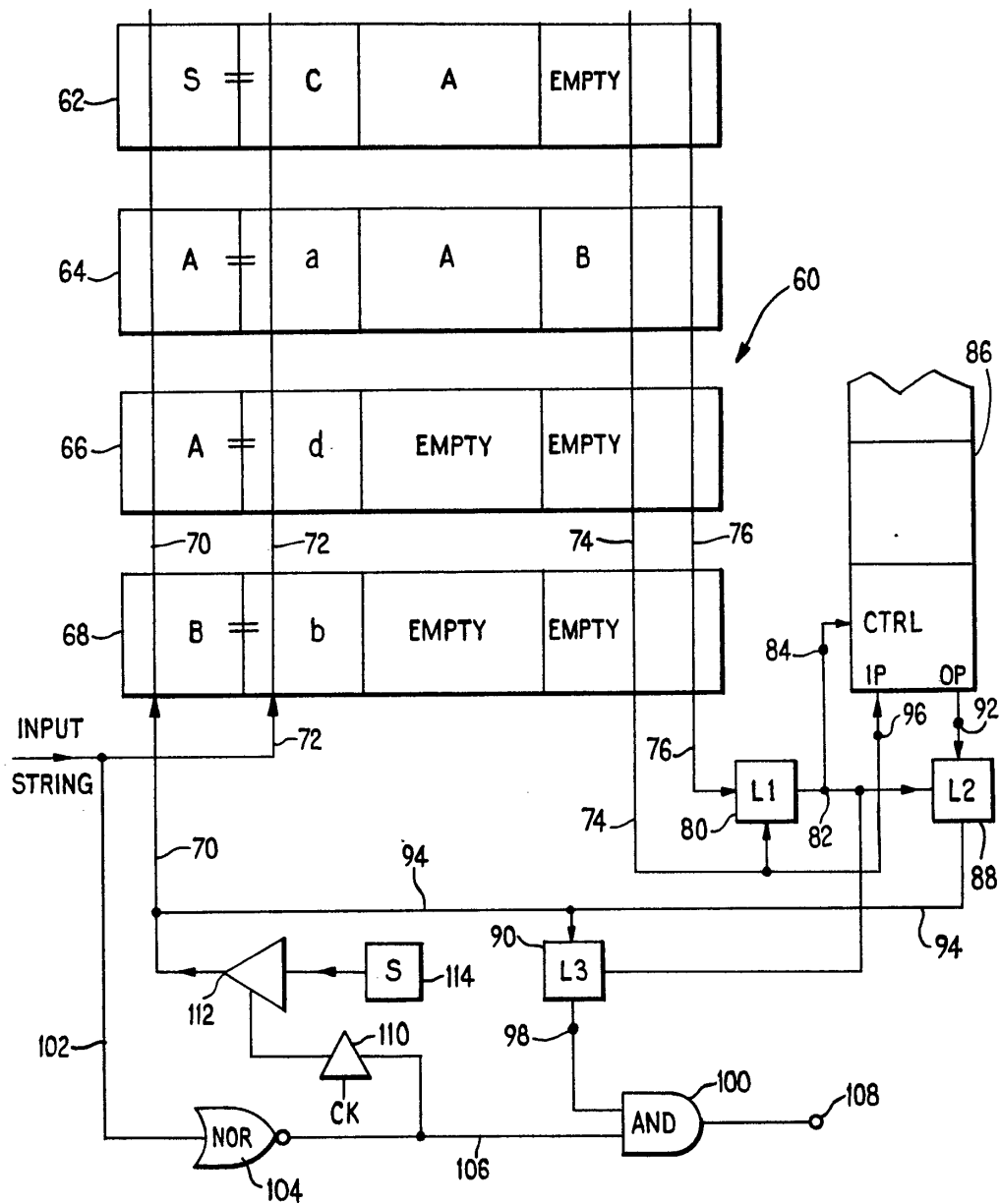


FIG. 2

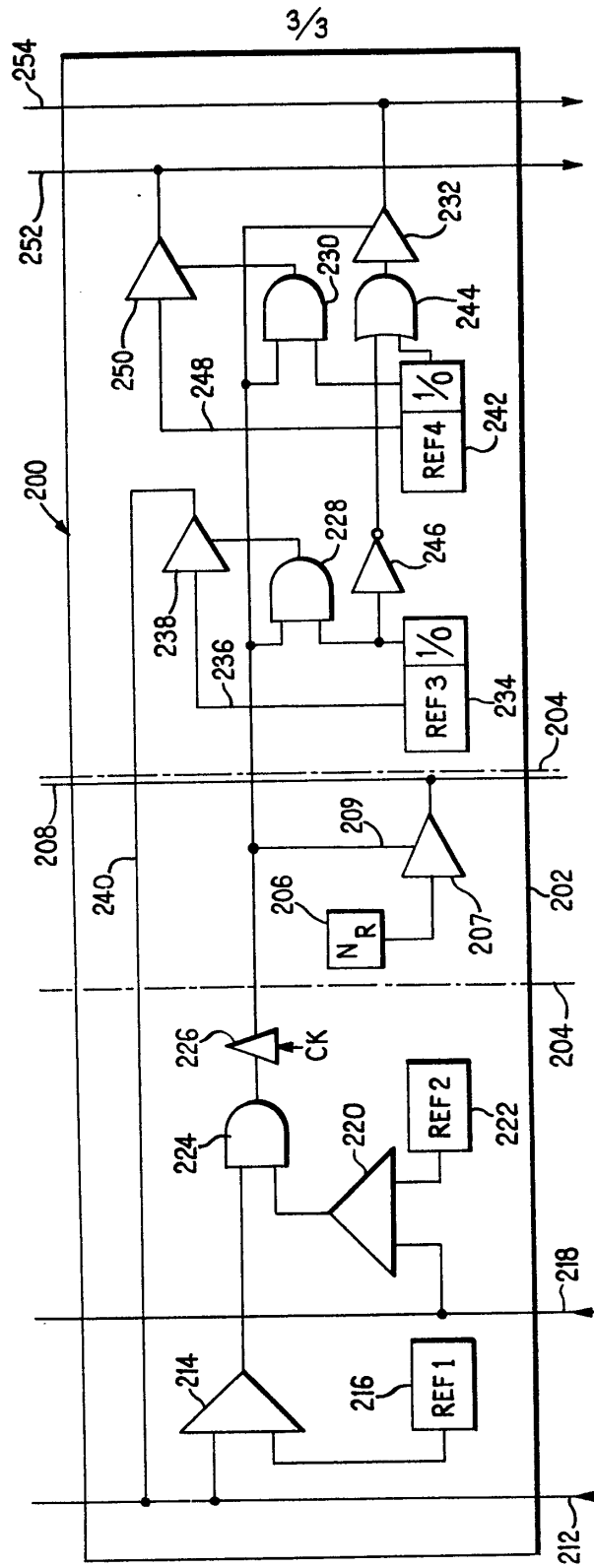


FIG. 3

SPECIFICATION

Language recognition apparatus

5 This invention relates to a language recognition apparatus for context-free languages expressed in Greibach's Normal form. 5

In mathematical terms, a grammar G is a finite set of rules of syntax or rewriting rules for generating sentences, ie strings of symbols or characters combined in a way which is syntactically correct. The language L(G) of a grammar G is the set of all such sentences which 10 can be generated from it by operation of its rewriting rules. 10

Generally, all possible grammars may be arranged in a hierarchy referred to as the Chomsky hierarchy, as follows:-

0. Unrestricted Grammars,
- 15 1. Context-Sensitive Grammars, 15
2. Context-Free Grammars, and
3. Regular Grammars.

Each of the grammar types 1, 2 and 3 is a subset of that immediately above it in the 20 hierarchy. In particular, a context-sensitive grammar (1) may contain rewriting rules or productions applicable only under the constraint that an appropriate context exists. A context-free grammar (2) is the special case of this that any context is sufficient. An unrestricted grammar (0) is the case where no constraints apply to the form of productions. 20

Unrestricted and context-sensitive grammars are generally relevant only to language theory, 25 whereas the present invention is concerned with grammars of the context-free regular kinds employed for example in computer programming languages. 25

A simple regular grammar G_R is defined by the expression:

$$30 \quad G_R = (N, \Sigma, P, S_0) \quad (1.1) \quad 30$$

where $N = \{S, A\}$, a set of non-terminal symbols; (1.2)

$\Sigma = \{a, b, c, d\}$, a set of terminal symbols; (1.3)

$P = \{S \rightarrow aS, S \rightarrow bA, S \rightarrow d, A \rightarrow c\}$, a set of productions 35 or rewriting rules for converting non-terminal symbols to terminal or irreducible symbols in one or more steps. 35

S = starting symbol (1.5)

G_R is termed a regular grammar because all of its productions are of the form $A \rightarrow cB$ or $A \rightarrow d$, 40 where A and B represent arbitrary non-terminals and c and d arbitrary terminals. The production set P allows replacement of the starting symbol S by any of aS, bA and d. When replaced by aS, the productions are applied to S again. When replaced by bA, the production A becomes c is applied. It can be seen that the complete set of terminal symbol strings obtainable from these productions, ie the language L(G_R), is given by:- 45

$$45 \quad L(G_R) = \{aa \dots ad, aa \dots abc, d, bd\} \quad (2) \quad 45$$

Expression (2) may be rewritten as a regular expression in conventional notation as:-

$$50 \quad L(G_R) = a^*(d + bc) \quad (3) \quad 50$$

where a^* represents an arbitrary number of occurrences of a including zero (star operation), and $(d + bc)$ indicates alternatives d or bc (union operation).

In the prior art, recognition devices of varying capabilities have been proposed for identifying 55 the regular expressions characteristic of respective regular languages. In VLSI '83 (Elsevier Science Publishers, BV, North-Holland), Curry and Mukhopadhyay described an integrated circuit for matching regular expressions consisting of character strings. The union operation can be accommodated, but not the star operation. In the Proceedings of VLSI '81 (Academic Press), Foster and Kung describe the use of integrated circuits for recognition of many regular 60 expressions. For any given expression, a respective design exercise is required to determine the interconnection scheme. In Applicant's co-pending United Kingdom patent application No 8406396 dated the 2nd of March 1984, a regular expression recognition apparatus is described which exhibits ease of design and recognition of all forms of regular expression. 60

In VLSI 83 (Elsevier Science Publishers B.V., North-Holland), K. Seo et al, describe a circuit 65 for parsing the Pascal computer language. The circuit incorporates programmable logic arrays. It 65

does not appear to be easily adaptable to parse other languages, and appears undesirably complex.

It is an object of the present invention to provide an alternative form of language recognition apparatus.

5 The present invention provides a language recognition apparatus incorporating a recognising circuit including:- 5

(1) means for comparing an input rule indication character with a first stored symbol,

(2) means for comparing a further input character with a second stored symbol,

(3) means for indicating synchronous matches between the input characters and the 10 respective first and second stored symbols, and 10

(4) means for transferring a third stored symbol for comparison as an input rule indicative character with the first stored symbol in response to match indication.

The invention tests successive input characters for compliance with a rule of grammar expressed by the stored symbols. This provides a very convenient and rapid means of testing for 15 grammatical correctness. The invention may include a plurality of recognising circuits each 15 arranged to test for compliance with a respective rule.

The recognising circuit may also include:-

(1) means for communicating a fourth stored symbol to a store in response to a match indication, and

20 (2) means for transferring a symbol from the store for comparison with the first stored symbol 20 in response to unavailability of any other rule indication character.

The apparatus of the invention preferably includes means for indicating grammatical correctness of input characters when:-

(1) the store is empty,

25 (2) the fourth symbol transferring means is operative, and 25

(3) there is no input character for comparison with the second stored symbol.

The invention may include means for inputting a string of sequential characters for comparison with the second stored symbol, and means for providing a starting symbol as a rule indication character for comparison with the first stored symbol in synchronism with comprison 30 of the leading string character with the second stored symbol. The recognising circuit preferably 30 includes parsing means for outputting a symbol characteristic of the circuit in response to synchronous matches between the input characters and the first and second stored symbols.

In a preferred embodiment, the invention includes a plurality of recognising circuits each having a respective unique set of stored symbols and being connected for simultaneous 35 comparison of successive pairs of input characters with each pair of first and second stored 35 symbols.

In order that the invention might be more fully understood, embodiments thereof will now be described, by way of example only, with reference to the accompanying drawings, in which:-

Figure 1 illustrates a recognising circuit of the invention

40 Figure 2 schematically shows a column of recognising circuits of the invention each arranged 40 to test for compliance with a respective rule of grammar, and

Figure 3 shows a further recognising circuit of the invention arranged for parsing.

Referring to Fig. 1, there is shown a rule recogniser cell 10. A rule bus 12 n bits wide passes 45 through the cell 10, and is connected to an n-bit first comparator 14, where n is the number of 45 bits of the digital characters in a rule of the relevant language. An n-bit store 16 is also 45 connected to the comparator 14, the store 16 containing a symbol (REF 1) representing the character on the left hand side of a rule or production. From Expression (1.4), this might be S of S→bA.

A signal input bus 18 m bits wide also passes through the cell 10, and is connected to an m- 50 bit second comparator 20. Here m is the number of bits of the digital characters of an input 50 symbol. An m-bit second store 22 is also connected to the comparator 20, and contains a second symbol (REF 2) representing the first character on the right hand side of a rule, ie b of S→bA in Expression (1.4). The first and second comparators 14 and 20 provide input signals to a first AND gate 24. Output signals from AND gate 24 pass to a latch 26, and thence as input 55 to second and third AND gates 28 and 30 and as control signals to a first bus driver 32. The 55 output of bus driver 32 is high impedance or the driver input signal for logic 1 or 0 control signals respectively. A third store 34 (n + 1) bits wide contains a least significant bit (1sb) of 1 or 0 indicating presence or absence of a symbol.

The n most significant bits (msbs) of the store 34 may be unfilled or contain a third symbol 60 (REF 3) representing a second right hand side character of a rule. The n msbs are connected by 60 an n-bit bus 36 to a second bus driver 38 equivalent to bus driver 32, and thence via a further n-bit bus 40 to the rule bus 12. The 1sb of store 34 provides a second input signal to second AND gate 28, the output signal from which controls bus driver 38.

A fourth store 42 n + 1 bits wide is equivalent to third store 34, and provides for a possible n 65 bit fourth symbol (REF 4) to be stored in its n msbs, together with a presence/absence 65

indicating lsb. REF 4 represents a third right hand side character of a rule. The lsb is connected as input to an OR gate 44, which receives a second input of the store 34 lsb after inversion by an inverter 46. The lsb also becomes a second input to second AND gate 30. The n msbs representing REF 4 if present are fed via an n-bit bus 48 to a third bus driver 50 controlled by an output from AND gate 30 and driving a stack input bus 52.

The one-bit output signal from OR gate 44 is input to the first bus driver 32, and passes thence to a stack control line 54 provided that the output of latch 26 is logic 1.

A system clock (not shown) controls the latch 26 and input of characters to signal input bus 18.

The Fig. 1 arrangement operates as follows. The stack input bus 52 and stack control line 54 communicate with a stack (not shown), a first in, last out store of the kind used in microprocessors. Table 1 sets out the stack functions executed in response to the combinations of states available to input bus 52 and control line 54.

Stack Input Bus	Stack Control line	Stack function
High Impedance	High Impedance	No Change
High Impedance	1	'POP' off stack
High Impedance	0	No Change
Any Value (not high Impedance)	1	'PUSH' on stack

TABLE 1

The cell 10 is arranged to compare a sequence of input signals or characters with symbols stored in stores 16, 22, 34 and 42 and defining a rule of grammar. It compares one character at a time, and the stack is arranged to store rule indicating characters temporarily for subsequent comparison.

The cell 10 is arranged to recognise any sequence of characters conforming to the rule:—

$$\text{REF1} \rightarrow (\text{REF2})(\text{REF3})(\text{REF4}) \quad (4)$$

where REFs 1, 3 and 4 are non-terminal symbols and REF 2 is a terminal symbol. Rule bus 12 and signal input bus 18 receive input of respective n-bit characters. The character input to rule bus 12 is compared with REF1 by comparator 14 on an initial clock cycle. If the characters do not match, a 0 is output to AND gate 24, which also provides 0 to latch 26. This occurs irrespective of the character on signal input bus 18, indicating that the cell rule of Expression (4) does not apply. On the second clock cycle, latch 26 outputs 0 and bus driver 32 presents a high impedance output to stack control line 54. In addition, this latch signal produces 0 outputs at AND gates 28 and 30, so that high impedance outputs are presented by bus drivers 38 and 50 to rule bus 12 and stack input bus 52 respectively. High impedance on stack input bus 52 and stack control bus 54 produces no change in the stack, as shown in Table 1.

On the second clock cycle, the cell 10 tests subsequent characters on rule bus 12 and signal bus 18. If the rule bus character matches the symbol stored at 16 but REF2 does not match that stored at 22, AND gate 24 receives inputs of 1 and 0. This produces a 0 output to latch 26, subsequently controlling bus drivers 38, 50 and 32 to present high impedance outputs to respective buses and line 12, 52 and 54. This results as before in no change in the stack on the third clock cycle.

If however both the rule and signal input bus characters match REF1 and REF2 respectively on the second clock cycle, AND gate 24 outputs a 1 to latch 26. On the third clock cycle, the 1 output of latch 26 reaches And gates 28 and 30 and the control input of bus driver 32. There are now three possible circuit operations according to the contents of stores 34 and 42. Both

may be empty, store 34 may hold a non-terminal symbol X with store 42 empty, or both may be full and contain non-terminal symbols X and Y respectively. Here X and Y are arbitrary non-terminal symbols. Referring to these alternatives as cases 1, 2 and 3, the corresponding lsbs of stores 34 and 42 will be 0 and 0, 1 and 0 or 1 and 1 respectively. Table 2 sets out the cases with their rules, characters REFs 1 to 4 and lsbs of stores 34 and 42. The situation store 34 empty but store 42 full does not arise, store 34 being filled first with any available non-terminal symbol.

Case	Rule	REF1	REF2	REF3	REF4	Store 34 lsb	Store 42 lsb
1	$S \rightarrow a$	S	a	Empty	Empty	0	0
2	$S \rightarrow aX$	S	a	X	Empty	1	0
3	$S \rightarrow aXY$	S	a	X	Y	1	1

TABLE 2

In Case 1, inverter 46 produces a 1 output from store 34 0 lsb. OR gate 44 provides a 1 output to driver 32, and this output is transferred to stack control line 54 since the driver control input from latch 26 is 1. AND gates 28 and 30 both receive 0 inputs from the lsbs of stores 34 and 42, and provide 0 control inputs to bus drivers 38 and 50 presenting high impedances to rule bus 12 and stack input bus 52. From Table 1, high impedance on the stack input bus 52 and 1 on the stack control line 54 instructs the stack to 'pop' off the last character (if any) input to it.

In Case 2, inverter 46 produces a 0 output from input of the lsb 1 in store 34. OR gate 44 receives two 0 inputs, since the lsb of REF4 is 0. Bus driver 32 accordingly places 0 on the stack control line 54, having received a 1 input from latch 26. Simultaneously, AND gate 30 receives 1 and 0 inputs from latch 26 and store 42 lsb respectively, and bus driver 50 receives a 0 control input. Stack input bus 52 therefore receives a high impedance state. From Table 1, no change in the stack takes place. Both inputs and therefore also the output of AND gate 28 are 1, as bus driver 38 places REF3 on rule bus 12. By virtue of the one clock cycle delay introduced by latch 26, REF3 appears on rule bus 12 in synchronism with input of the next character on signal input bus 18, and comparison takes place as before.

Regarding Case 3, OR gate 44 produces a 1 output by virtue of 0 and 1 inputs from inverter 46 and store 42 lsb. The control input to bus driver 32 is 1 as in Cases 1 and 2, and so driver 32 places a 1 on stack control line 54. As in Case 2, rule bus 12 receives REF3. AND gate 30 receives two 1 inputs, and accordingly controls store 42 to place REF4 on stack input bus 52. From Table 1, the combination of a 1 on stack control line 54 and any value on stack input bus 52 'pushes' REF4 on the stack. As in Case 2, REF3 is available on rule bus 12 for comparison with the next character on signal input bus 18. Unlike Case 2, REF4 is available on the stack for comparison with a subsequent input character.

Referring now to Fig. 2, there is shown a language recognition apparatus indicated generally by 60. It incorporates schematically illustrated recogniser cells 62 to 68 of the kind shown in Fig. 1, each cell being labelled with respective characters representing REFs 1 to 4 in accordance with the relevant rule. The apparatus 60 is arranged to recognise a context free grammar G_{CF} corresponding to language L_{CF} and expressed in Greibach's normal form, where:-

$$G_{CF} = (N, \Sigma, P, S), \quad (5.1)$$

$$S = \text{starting symbol} \quad (5.2)$$

$$N = \{S, A, B\} \text{ (non-terminal symbols)} \quad (5.3)$$

$$\Sigma = \{a, b, c, d\} \text{ (terminal symbols)} \quad (5.4)$$

$$P = \{S \rightarrow cA, A \rightarrow aAB, A \rightarrow d, B \rightarrow b\} \quad (5.5)$$

(rewriting rules or productions)

Each of the cells 62 to 68 is shown schematically labelled by the characters of the respective production to which it relates. Table 3 sets this out.

TABLE 3

5	Cell	REF1	REF2	REF3	REF4	5
10	62	S	c	A	empty	10
	64	A	a	A	B	
	66	A	d	empty	empty	
15	68	B	b	empty	empty	15

The cells 62 to 68 are arranged in a column with a common rule bus 70, signal input bus 72, stack input bus 74 and stack control line 76.

20 The order of the cells in the column is arbitrary. Stack control line 76 and stack input bus 74 are connected to a first logic circuit 80 arranged to generate the Table 1 logic functions at an output 82. The signal at 82 is fed to a control input 84 of a schematically illustrated stack 86, and also to second and third logic circuits 88 and 90. A stack output 92 is connected via circuit 88 to rule bus 70 and circuit 90 by an output bus 94. The stack input bus 74 is connected to a stack input 96. First logic circuit 80 has a two-bit output to indicate each of the four possibilities of Table 1. It controls second logic circuit 88 and stack control input 84 accordingly to provide stack functions 'pop', 'push' or no change as appropriate. The stack might be implemented as a left/right shift register controlled by logic circuitry to respond appropriately to a two-bit control input. This is well-known in the art. Second logic circuit 88 incorporates a bus driver (not shown) operative to place the stack output on rule bus 70 when the output of first logic circuit 80 indicates its receipt of a 'pop' condition. Third logic circuit 90 has a one-bit output which is normally 0, but becomes 1 when there is no value on bus 94 and the output of first logic circuit 80 corresponds to receipt of high impedance/logic 1, ie 'pop'. Logic circuitry for implementing the functions of circuits 80, 88 and 90 is straightforward and well known in the art, and so will not be described further. Third logic circuit output 98 is connected to one input of an AND gate 100. The signal input bus 72 is connected via a bus 102 to an n-bit NOR gate 104, whose output signal passes via a line 106 to provide a second input to AND gate 100. The output from AND gate 100 appears at 108. The output signal from NOR gate 104 is also fed to a latch 110 introducing a one clock cycle delay, and thence as control input to a bus driver 112. A store 114 holds the starting symbol S, and the bus driver 112 is connected to place this on the rule bus 70 on receipt of a logic 1 control input from latch 110.

The language recogniser circuit 60 operates as follows. Consider an input string of n-bit characters caabb applied to signal input bus 72, where each character has at least one non-zero bit. Absence of a character is indicated by n zero bits. Referring to Expression (5.5), the string caabb is generated by the sequence of productions set out in Table 4.

50	PRODUCTION	EFFECT	50
	S → cA	cA	
	A → aAB	caAB	
55	A → aAB	caaABB	55
	A → d	caadBB	
	B → b	caadbB	
60	B → b	caadbb	60

TABLE 4

Since the string is generated by applying rules or productions of G_{CF} , it should be recognised by the circuit 60. The string is input in a sequential, bit-parallel fashion to signal input bus 72. Prior to input of the string, n zero bits appear on the bus 72, and the output of n -bit NOR gate 104 is 1, as is the value held in latch 110. Bus driver 112 accordingly maintains the starting symbol S from store 114 on rule bus 70. When any character of the string appears on signal input bus 72, the corresponding non-zero input to NOR gate 104 produces a zero output. Bus driver 112 is accordingly in a high impedance state one clock cycle later, and does not place S on rule bus 70 until one clock cycle after characters have ceased to appear on signal input bus 72. The only time that bus driver 112 places S on rule bus 70 synchronously with a character on signal input bus 72 is when the first character of a string is input.

Input of the first character c of the string occurs on a first clock cycle while S appears on rule bus 70. These will match REFs 1 and 2 in cell 62, but produce non-matches in cells 64 to 68. Referring now also to Fig. 1, in cell 62 the output of AND gate 24 to latch 26 will be logic 1. On the second clock cycle, S is removed from rule bus 70, and the output of latch 26 becomes logic 1. Since REF3 is non-empty and REF 4 is empty, stack control bus 76 receives logic 0. In addition the cell 62 value of REF3, symbol A , is placed on rule bus 70 synchronously with input of the second character a of the string to signal input bus 72. Whilst this took place, the bus drivers 32, 38 and 50 of cells 64 to 68 were all in high impedance states effecting isolation from the rule and stack input buses and the stack control line.

When on the second clock cycle A and a respectively appear on the rule and signal input buses 70 and 72, REFs 1 and 2 in cell 64 produce a match and cells 62, 66 and 68 are isolated. Although REF1 of cell 66 matches A , its REF2 does not match a . On the third clock cycle, latch 26 in cell 64 outputs logic 1 and the cell's REF3 character A is transferred to rule bus 70. Simultaneously, symbol B or REF4 of cell 64 is output to stack input bus 74, and stack control bus 76 receives logic 1 from store 42 lsb; from Table 1, this combination on bus 74 and line 76 causes logic circuit 80 to apply the appropriate two-bit code to stack control input 84 so that character B is 'pushed' on to the stack 86 via input 96. Meanwhile, signal input bus 72 has received the third character a of the string, which coincides with character A on rule bus 70 as previously said. Cell 64 accordingly operates once more in the same way, a second character B being 'pushed' on to the stack 86 with character A being placed on rule bus 70 one clock cycle later (the fourth). On this clock cycle, the fourth character d of the string appears on signal input bus 72. The combination of A and d causes cell 66 to operate. This cell has REFs 3 and 4 both empty, so stack control line 76 receives logic 1 on the fifth clock cycle, is one clock cycle later, and stack input bus is at high impedance, from Table 1, B is 'popped' off the stack 86 synchronously with input of fifth string character b . This operates cell 68, which also has REFs 3 and 4 empty. The second B is therefore 'popped' off the stack 86 on the sixth cycle, whilst final string character b is input to signal input bus 72 and cell 68 operates once more.

On the seventh clock cycle there is no input string character, and all n bits of signal input bus 72 are 0. NOR gate 104 accordingly outputs logic 1 to AND gate 100. The concurrent operation of cell 68 provides logic 1 on stack control line 76 and high impedance on stack input bus 74, corresponding to the 'pop' instruction. Third 'logic circuit 90 detects the corresponding two-bit output of first logic circuit 80, and detects no rule symbol on bus 94. It therefore responds by outputting logic 1 to AND gate 100. Output 108 of AND gate 100 accordingly switches from its previous logic 0 state to logic 1, indicating that the input string has obeyed the rules of grammar G_{CF} set out in Expression (5.1) to (5.5); ie the string $caasbb$ is a 'sentence' of the context free language L_{CF} . Table 5 summarises the foregoing sequence of events. Although not apparent in this example, the combination of logic circuit 90, AND gate 100 and NOR gate 104 is arranged to avoid spurious recognition indications part way through an input string.

TABLE 5

5	Clock cycle	Rule Bus	Signal Input Bus	Stack Input Bus	Stack Control Line	Stack Contents	NOR Gate 104 Output	Logic Circuit 90 Output	AND Gate 100 Output	5
10	0	S	0	High Imped- ance	0	empty	1	0	0	10
15	1	S	c	High Imped- ance	0	empty	0	0	0	15
20	2	A	a	B	1	empty	0	0	0	20
25	3	A	a	B	1	B	0	0	0	25
30	4	A	d	High Imped- ance	1	BB	0	0	0	30
35	5	B	b	High Imped- ance	1	B	0	0	0	35
40	6	B	b	High Imped- ance	1	empty	0	0	0	40
45	7	0	0	High Imped- ance	1	empty	1	1	1	45

Referring to Fig. 3, there is shown a language recognised cell 200 of the invention for parsing a sentence of a language having a context-free grammar. The cell 200 is equivalent to cell 10 of Fig. 1 with an additional central section 202 between chain lines 204. Parts equivalent to those shown in Fig. 1 are like-referenced with the prefix 200.

The central section 202 has a store 206 connected to a bus driver 207 and thence to a parse output bus 208. The output signal from the latch 226 is connected via a line 209 as control input to the bus driver 207. The store 206 holds a number indicating which rewriting rule the cell 200 implements.

The cell 200 carries out all the operations of cell 10. An additional function of indicating application of the cell rule is provided. When latch 226 outputs logic 1 indicating matches obtained for REFs 1 and 2 on the previous clock cycle, bus driver 207 switches from high impedance to placing the number held in store 206 on parse output bus 208. Cell 200 will be one of a column (not shown) of like cells through which bus 208 passes. Matches to REFs 1 and 2 in each cell results in its rule number reaching parse output bus 208. The bus 208 accordingly outputs a series of numbers indicating the sequence of rules applied to check that the input string fed to signal input bus 218 was in accordance with the given grammar. This constitutes the parsing operation, which is defined as identifying the sequence of rules of grammar applied in generation of the input character string or sentence of the language.

The examples of the invention hereinbefore described are capable of recognising grammars

having the following forms of rewriting rules:-

$$Z_1 \rightarrow a_1 \quad (6.1)$$

$$Z_2 \rightarrow a_2 X_2 \quad (6.2)$$

$$5 \quad Z_3 \rightarrow a_3 X_3 Y_3 \quad (6.3)$$

5

where non-terminal symbols are upper case characters and terminals lower case. Expressions (6.1) to (6.3) represent context-free grammars having only one rule beginning with any given terminal symbol, and each rule contains a maximum of two non-terminal symbols into which an initial terminal may be rewritten. The computer software language Pascal has such a grammar. 10

The invention may be extended to grammars incorporating longer rules, ie $A \rightarrow ABCD \dots$, by making provision for extra storage and rules stacking in cell 10 or 200, stack 86 and associated logic. However, it would in general be necessary to provide for input string characters to undergo differing degrees of delay, such as by storing them on a further stack and 'popping' each one off as necessary. This enables cells to operate for differing length of time as required to apply long rewriting rules. 15

It will be appreciated that the invention provides the advantages of modular construction and ease of programmability. A language having a set of grammatical rules may be recognised by a column of equivalent cells with one cell per rule. The cells are programmed with their respective rules simply by inserting appropriate versions of REFs 1 to 4 or more. To implement this, stores 16, 22, 34 and 42 or their Fig. 3 equivalents may be a line of shift registers connected in series. Moreover, the shift registers of all cells 62 to 68 in Fig. 2 could be arranged in a line connected in series. Programming the cells then becomes a simple matter of inserting the appropriate sequence of symbols into the shift register line. 20

CLAIMS 25

1. Language recognition apparatus incorporating a recognising circuit including:-
 (1) means for comparing an input rule indication character with a first stored symbol,
 (2) means for comparing a further input character with a second stored symbol,
 30 (3) means for indicating synchronous matches between the input characters and the respective first and second stored symbols, and

(4) means for transferring a third stored symbol for comparison as an input rule indication character with the first stored symbol in response to match indication.

2. Language recognition apparatus according to Claim 1 wherein the recognising circuit additionally includes
 35 (1) means for communicating a fourth stored symbol to a store in response to a match indication, and

(2) means for transferring a symbol from the store for comparison with the first stored symbol in response to unavailability of any other rule indication character.

3. Language recognition apparatus according to Claim 2 including means for indicating grammatical correctness of input characters when:- 40

(1) the store is empty,
 (2) the fourth symbol transferring means is operative, and
 (3) there is no input character for comparison with the second stored symbol.

4. Language recognition apparatus according to Claim 1, 2 or 3 including means for inputting a string of sequential characters for comparison with the second stored symbol. 45

5. Language recognition apparatus according to Claim 4 including means for providing a starting symbol as an input rule indication character for comparison with the first stored symbol in synchronism with comparison of the leading string character with the second stored symbol.

6. Language recognition apparatus according to any preceding claim wherein the recognising circuit includes parsing means for outputting a symbol characteristic of the circuit in response to synchronous matches between the input characters and the first and second stored symbols. 50

7. Language recognition apparatus according to any preceding claim including a plurality of recognising circuits in accordance with Claim 1, 2 or 6, each circuit having a respective unique set of stored symbols and being connected for simultaneous comparison of successive pairs of input characters with each pair of first and second stored symbols. 55