The Computability of Stack Non-Underflow

D.T. GOODWIN
University of Keele, Newcastle, Staffs ST5 5BG

In a number of previous articles a model of arithmetic behaviour at run-time was set up and studied. In the last of these, Goodwin[3], sufficient conditions were presented under which an infinity of different program runs may use the stack without underflow. This paper shows these conditions to be necessary, so that it is computable at compile-time whether or not such an infinity exists.

1. INTRODUCTION

This paper continues and probably concludes a series of articles in which a model of arithmetic stack behaviour was set up and studied. The first of these was Goodwin[1] and the last was Goodwin[3]. It is supposed that a user's program is written in a high-level language which makes implicit or explicit use of an arithmetic stack. Any conventional high-level language is acceptable, and Pop2 is the most general case known – distinct in that it uses a stack explicitly and also in that its individual statements may cause a net gain or loss of items on the stack. (Equally general are micro-computer machine codes where a stack is often employed.) Under conditions always satisfied in (say) Algol, Fortran or Pascal, and frequently satisfied in Pop2, the user's particular program is shown to act as a context-free or BNF-type grammar whose function is to generate a sequence of symbols, as the program runs. These symbols denote commands to add or remove items from the arithmetic stack. The whole sequence or string of symbols varies depending on the data, so that in general an infinity of such strings has to be considered.

The subjects of the previous papers have been:

- 1977: Conditions for the stack-length to be totally bounded above and below.
- 1980: Conditions under which an infinity of program runs, not necessarily all, may use a bounded stack.
- 1985: Sufficient conditions under which an infinity of program runs may use a stack which is only bounded below.

This article shows that the final conditions already presented as sufficient are also necessary. This leads to the conclusion that it is computable at compile-time whether or not an infinity of program runs exists whose stack-lengths are at all times bounded below.

2. CONCEPTS AND NOTATION

This paper relies on the concepts and notation of its predecessors, but a very brief account of useful ideas is given below. Theorem and lemma numbers are continued, previous theorems and lemmas being distributed as follows:

(1977): Theorems 1–6, lemmas 1–4.
(1980): Theorems 7–16, lemmas 5–6.
(1985): Theorems 17–20, lemmas 7–12.

In the examples of formal grammars and strings, individual non-terminals are denoted by capitals, while small letters denote strings of terminals, or terminals and non-terminals, as explained in the context. Hopefully confusion with small letters that denote integers is minimal. Individual terminals have the effect of incrementing or decrementing the length of the stack. The significance of these actions is not relevant to the model being studied, but in the arithmetic stack application incrementing would take place if an item were fetched from store, while decrementing would correspond to a transfer of the top item of the stack to store, or to the combination of the top two items using an arithmetic operation. The particular grammar under discussion, which corresponds to the text of the user's particular program, is called G, and the examination of its properties corresponds to computation carried out at compile-time. The special non-terminal of G which is expanded to form a string of the language is called S.

A 'cycle' is used in the sense of a number of production-rule applications which together generate uNv from N. This is written N ⇒ uNv, where u and v are in general mixtures of terminals and non-terminals. The 'length' of a cycle C is written l(C), and is the net number of items that u and v together add to the stack. The left-hand-length lhl(C) of the cycle is the net number of items that u alone adds. These lengths may vary depending on the expansions into terminals of the non-terminals of u and v. $l_0(C)$ is a length of uv when all their non-terminals are expanded non-cyclically, ie with no repeating non-terminals down any branch of the tree of such an expansion. 'Basic' cycles are a finite set of cycles from which all cycles may be generated, by 'composition ' (ie combination) or repetition.

3. AN INVALUABLE TOOL

The following simple but powerful numeric lemma is the basis of all the subsequent theorems. No originality is claimed.

Lemma 13

Let an integer n be fixed, and let S be an infinite set of distinct vectors $X = [x_1, x_2, \ldots x_i, \ldots x_n]$, where each x_i, ($1 \leqslant i \leqslant n$), takes integer values which are bounded below. Then there exists an infinite sequence of vectors X_1, X_2, $\ldots X_p, \ldots X_q, \ldots$ all in S such that for all p, q, $p < q$, then:

x_i of $X_p = x_i$ of X_q for all i in $1 \leqslant i \leqslant n$, for which the set of x_i values in S is also bounded *above*.

x_i of $X_p < x_i$ of X_q for all i in $1 \leqslant i \leqslant n$ for which the set of x_i values in S is not bounded above.

Proof

There is a simple proof by induction on n. It is certainly true for $n = 1$, (when the x_i values must be unbounded above). The task now is to show it true for $n+1$ if it is true for n.

Consider the infinity of vectors $[x_1, x_2,...x_{n+1}]$. Then each vector has an n-vector $[x_2,...x_{n+1}]$ within it. Consider the set of all such vectors $[x_2,...x_{n+1}]$. Since some may coincide there are two cases:

1. The number of distinct $[x_2,...x_{n+1}]$ is finite. Then for at least one particular $[x_2,...x_{n+1}]$ vector there exists an infinity of original $(n+1)$-vectors which contain it, with an infinity of different x_1 values. The desired infinite sequence is obtained by listing these $(n+1)$-vectors in order of ascending x_1 values. The other x_i remain constant which is as the lemma states, since case 1 implies boundedness above for x_2 to x_{n+1}.
2. The number of $[x_2,...x_{n+1}]$ is infinite. Then the theorem can be applied to them, using the inductive hypothesis, and there exists a sequence $X_1,...X_p,...X_q,...$ of them. Consider now the $X_1,...$ associated with their original x_1 values – (called respectively $x_1(1)$, $x_1(2),...x_1(p),...x_1(q)...$). These constitute the sequence $(x_1(1),X_1)$, $(x_1(2), X_1)$, ... $(x_1(p), X_p)$,... , and again two cases are distinguishable:

(a) The x_1 values are unbounded above. Then choose any p and one can always find q, $q > p$, for which $x_1(q) > x_1(p)$. Now take the value of q and use it as p in a repetition of the above step – thus one can find an infinite increasing sequence of integers $p_1, p_2, ...p_r...$ whose x_1 values are strictly increasing. Thus the required infinite sub-sequence of vectors is $(x_1(p_1), X(p_1)), (x_1(p_2), X(p_2)),...$.

(b) The x_1 values are bounded above. Let the upper and lower bounds of x_1 be u and l respectively. Then there exists t, $l \leqslant t \leqslant u$, for which there exists an infinite number of p-values for which $x_1(p) = t$. This again defines the required infinite sub-sequence of vectors.

The lemma is therefore true for all n.

One useful aspect of this lemma is that any dependencies there may be between the components of the vectors do not affect the result. The existence of an infinity of vector-instances is a sufficiently powerful condition.

4. S-CYCLES ONLY IN THE S-CHAINED DERIVATION

A simple case of the general theory of Theorem 25 is first developed, so that the kernel of the argument may be presented without added complications.

Theorem 21

Let there be an infinity of non-underflowing strings, each with the property that in its parse-tree there is an instance of S which has a non-cyclic expansion into terminals, and in the infinity let the depth of such S-instances be unbounded. Then there exists a cycle C such that $lhl(C) \geqslant 0$ and $l(C) \geqslant 0$.

Proof

In the parse-tree of each non-underflowing string, let S* be an instance of S which expands non-cyclically. Hence there exist one or more cycles $S \Rightarrow u_1Sv_1$, $S \Rightarrow u_2Sv_2$, ... $S \Rightarrow u_iSv_i$, ... to $S \Rightarrow u_mSv_m$, not necessarily distinct, such that the base-node S expands thus:

$$\begin{aligned} S &\Rightarrow u_1Sv_1 \\ &\Rightarrow u_1u_2Sv_2v_1 \\ &\dots \\ &\Rightarrow u_1u_2...u_mS^*v_m...v_2v_1 . \end{aligned}$$

Here the u and v symbols denote strings of terminals and non-terminals. The integer m may vary from string to string, and denotes the depth of S* (ie the number of cycles between the first S-instance and S*). The idea of the proof is to show that certain sums of the lengths of the u_i and v_i are non-negative, and then to demonstrate the existence of a cycle which has just such a length and left-hand-length. Consider the lengths of u_i and v_i in $S \Rightarrow u_iSv_i$. Each non-terminal in u_i and v_i may expand into terminal strings whose lengths are generated by a finite number of formulae as in Lemma 6. Let such a non-terminal be N, and let $N \Rightarrow s(N)$ be one of the finite number of non-cyclic expansions of N into terminals. Let $C_1, ...C_j, ...C_n$ be a set of basic cycles of G. Then, rephrasing the result of Lemma 6:

$$l(N) = l(s(N)) + \sum_j x_{Nsj}\, l_0(C_j),$$

where the x_{Nsj} are non-negative integers distinct for every combination of N, s(N) and j. Whereas in Lemma 6 the formulae were used as generators of all possible lengths (by assigning all integers to the x variables), here the use is different – they are to be used to denote just those lengths that actually turn up when N is used in cycles in the given infinity of strings. Thus any particular x_{Nsj} may never take a certain value, and indeed some may never differ from zero, because for convenience all the zero-cyclic lengths $l_0(C_j)$ of G have been included, some of which may not be accessible in expansions of the non-terminal N. There is also no information about whether any of the x_{Nsj} are bounded above.

Returning again to the typical cycle $S \Rightarrow u_iSv_i$, seeking a formula for $l(u_i)$ means selecting one of a finite number of formulae for l(N), for each N in u_i. The difficulty of thinking about this is resolved by regarding the original cycle as a number of distinct cycles, one for each of the finite number of distinct combinations of l(s) values for all the non-terminals of u_i and v_i. (This had already been implicitly assumed in using the $l_0(C_j)$ values in the formulae). Suppose the original $S \Rightarrow u_iSv_i$ cycles have been renumbered to allow for this, and let $u_i = u_iN_1N_2...N_p...N_h$, and $v_i = v_i'M_1M_2...M_q...M_k$, where u_i' and v_i' are strings of terminals only. Then as various expansions of the non-terminals are considered, the lengths of u_i are given by:

$$\begin{aligned} l(u_I) &= l(u_I') + \sum_1^h l(N_s) \\ &= l(u_i') + \sum_{p-1}^{h} l(s(N_p)) + \sum_{p-1}^{h} \sum_j x_{Npj}\, l_0(C_j) \\ &= U_i + \sum_j x_{ij}\, l_0(C_j), \end{aligned}$$

for some integral constant U_i and new non-negative variables x_{ij}. (Again it is stressed that the x_{ij} may not take all integer values). Similarly denote:

$$l(v_i) = V_i + \sum_j y_{ij}\, l_0(C_j).$$

It was stated earlier that the cycles $S \Rightarrow u_i S v_i$ are not necessarily distinct. For any one string of the infinity of non-underflowing strings, let the i'th distinct cycle $S \Rightarrow u_i S v_i$ be repeated r_i times, possibly with different values of the x_{ij} and y_{ij}. Then the total length of all $l(u_i)$ for all these repeats, for fixed i, is:

$$\sum_{\substack{\text{all}\\ \text{repeats}}} l(u_i) = r_i\, U_i + \sum_j \Big(\sum_{\substack{\text{all}\\ \text{repeats}}} x_{ij}\Big) l_0(C_j)$$

and similarly for $\Sigma l(v_j)$.

Then returning to the original cycle $S \Rightarrow u_1 u_2 \ldots u_m S v_m \ldots v_1$,

$$l(u_1 \ldots u_m) = \sum_i \Big[\sum_{\substack{\text{all}\\ \text{repeats}}} l(u_i)\Big]$$

$$\sum_i r_i\, U_i + \sum_j \Big\{\sum_i \Big(\sum_{\substack{\text{all}\\ \text{repeats}}} x_{ij}\, l_0(C_j)$$

$$= \sum_i (r_i\, U_i) + \sum_j k_j\, l_0(C_j), \text{ say,}$$

where the variables over all the strings of the infinity are the r_i and the k_j. Similarly:

$$l(v_m \ldots v_1) = \sum_i (r_i\, V_i) + \sum_j n_j\, l_0(C_j), \text{ say,}$$

where the variables over the infinity are the r_i and the n_j.

Now let the lower bound of the stack-length be L. Then for every string considered:

$$l(u_i \ldots u_m) \geqslant L$$

$$\textit{and} \qquad l(u_1 \ldots u_m) + l(w) + l(v_m \ldots v_1) \geqslant L.$$

Consider the infinity of vectors $X = [m, r_1, r_2, \ldots, k_1, \ldots n_1, \ldots l(u_1, \ldots u_m),\ l(w),\ l(u_1 \ldots u_m) + l(v_m \ldots v_1)]$. The number of r values is fixed because this is the number of distinct cycles involving S. The number of k and n values is fixed because this is the number of distinct basic cycle-lengths in all G. Furthermore every component of X takes integer values only and is bounded below. Thus the conditions of Lemma 13 apply and there exist strings T_1 and T_2 and associated vectors X_1 and X_2 such that $[X_2]_q \geqslant [X_1]_q$ for every component q of X taken in turn. The only component known to be bounded above is l(w), for which the equality holds. Let the variables for T_1 and T_2 be pre-superfixed 1 and 2 respectively. Then:

$${}^2l(u_1 \ldots u_m) - {}^1l(u_1 \ldots u_m) = ({}^2r_i - {}^1r_i) U_i + \sum_j ({}^2k_j - {}^1k_j) l_0(C_j)$$

$$= {}^*r_i\, U_i + \sum_j {}^*k_j\, l_0(C_j)$$

(for constants $^*r_i$ and $^*k_j$ just shown to be non-negative)

$$\geqslant 0\,.$$

Similarly

$$\begin{aligned} &{}^2l(u_1 \ldots u_m) + 1(w) + {}^2l(v_m \ldots v_1) - {}^1l(u_i \ldots u_m) \\ &\quad - {}^1l(w) - {}^1l(v_m \ldots v_1) \\ &= \sum_j {}^*r_i (U_i + V_i) + \sum_j ({}^*k_j + {}^*n_j) l_0(C_j) \end{aligned}$$

(for non-negative $^*n_j$)

$$\geqslant 0\,.$$

These two sums just shown to be not less than zero are remarkably like the left-hand and total lengths of a cycle. To prove the theorem it would be sufficient to show that a cycle with these lengths could be constructed. If all the $^*r_i$ are non-zero this follows at once, and in Theorems 23 and beyond this will be assumed at the start. There is a problem however if for some i, $^*r_i = 0$. (Certainly $^*r_i$ cannot be zero for all i because the depth m has been included in the vector – it is given as unbounded, and hence the depths of T_1 and T_2 differ). It could then happen that the cycle $S \Rightarrow u_i S v_i$ contained in u_i the only instance (in all of the possible $u_1, u_2, \ldots u_m$) of a non-terminal on to which instances of a particular basic cycle C_j could be attached. Omission of the ith cycle from the composed cycle now being constructed would thus prevent the addition of the number $^*k_j l_0(C_j)$ to the left-hand and total cycle lengths. This addition would only be of value if $^*k_j l_0(C_j)$ were positive and could otherwise be ignored. Thus in the difficult case, for some i and j, there exists a cycle $S \Rightarrow u_i S v_i$, where $u_i = aNb$ for some strings a and b, and $N \Rightarrow uNv$, where $l(uv) = l_0(C_j) > 0$. It follows then that by repeated application of the cycle $N \Rightarrow uNv$, the cycle $S \Rightarrow u_i S v_i$ may have its left-hand and total lengths increased as much as desired, until both are non-negative, thus proving the theorem.

There remains the similar difficulty in the case where for some $i, j, {}^*r_i = 0$ and $^*n_j > 0$. It has already been dealt with if at the same time $^*k_j > 0$, but suppose $^*k_j = 0$. Then for some i and j there exists a cycle $S \Rightarrow u_i S v_i$, where $v_i = cNd$ for some strings c and d, and where $N \Rightarrow uNv$, where $l(uv) = l_0(C_j) > 0$. Using the C_j cycle, $l(v_j)$ can be increased as much as desired. If $l(u_i) \geqslant 0$, the theorem is now proved. If $l(u_i) < 0$, it is still possible to prove the theorem if ${}^2l(u_1 \ldots u_m) - {}^1l(u_1 \ldots u_m)$ t 0. Then the constructed cycle contains S with a positive left-hand-length and negative total length, while $S \Rightarrow u_i S v_i$ has a negative left-hand length, and arbitrarily large total length. Since both contain S they may be composed to form a single cycle, and with suitable multiples the resulting left-hand and total lengths can both be non-negative.

Summarising, there remains the case where the cycle being constructed has zero left-hand length, negative total length, $^*r_i = 0$ for some i, and $^*k_j = 0$ and $^*n_j > 0$ for the basic cycle $N \Rightarrow uNv$ of length $l(C_j) = l(uv) > 0$, where $S \Rightarrow u_i ScNd$ such that $l(u_i) < 0$. If now $l(u) \geqslant 0$, the theorem is proved using the C_j cycle, so suppose $l(u) < 0$. Then it is impossible to find any acceptable ratio in which to compose the constructed and i'th cycles, as happened before. In fact the situation is very puzzling since the intuitive conditions hopefully to be proved do not appear to hold. This is all resolved by showing that the state of affairs cannot in fact occur – it only presents itself in the argument because of the crudeness of the tests for underflow which were applied – the requirements were just that underflow should not occur after all the terminal symbols before w had been processed, and after all symbols whatsoever had been processed.

To show the above state must lead to underflow, consider the deposition of the symbols of the string T_1. The symbols before w reduce stack-length by a finite amount $r_i l(u_i)$, since r_i is bounded in the infinity. After w the contributions are from the $S \Rightarrow u_i S v_i$ instances and all the others, mingled in some order. The latter each contribute negatively – indeed positive contributions from the right-hand lengths of C_j instances cannot begin until after an unbounded number of negative C_j left-hand-lengths have been loaded. Thus underflow is certain to occur, in whatever order the cycles appear. The same argument would hold if more than one C_j cycle were involved at a time. The theorem is now proved.

An interesting aspect of the theorem and its proof is that it imposes no restrictions on the nature of the expansions of the left-hand and right-hand sides of the cycles of S, apart, of course, from underflow. Thus S-cycles can occur in them, and an unbounded profusion of branches, as well as cycles, may develop, when permitted by the grammar. (The author found this case a stumbling-block for a long time). These remarks apply as well to the more general theorems below.

5. T-CYCLES ONLY IN THE S-CHAINED DERIVATION

A small extension of Theorem 21 is now undertaken.

Theorem 22

Let there be an infinity of non-underflowing strings, each with the property that in its parse-tree there exists an instance T^* of a non-terminal T which has a non-cyclic expansion into terminals, such that $S \Rightarrow a_1 T b_1$ (non-cyclically) $\Rightarrow a_1 u T^* v b_1$. It is assumed as well that $l(a_1)$ is bounded in the infinity, and that the depth of T^* is unbounded. Then there exists a cycle C such that $lhl(C) \geqslant 0$ and $l(C) \geqslant 0$. (This reverts to Theorem 21 by choosing the subset of grammars in which S and T are identical).

Proof

The argument is as for Theorem 21, reading T in place of S. However the non-underflow conditions are now:

$$l(a_1) + l(u_1 \ldots u_m) \geqslant L$$
$$l(a_1) + l(u_1 \ldots u_m) + l(w) + l(v_m \ldots v_1) \geqslant L$$

from which $l(u_1 \ldots u_m)$ and $l(u_1 \ldots u_m) + l(v_m \ldots v_1)$ are bounded below as before. Hence the theorem follows.

The purpose of presenting this trivial extension is to discuss the variation in conditions. From the position of a_1 at the beginning of strings in the infinity, $l(a_1)$ must be bounded below to prevent underflow. The first inequality above only works if $l(a_1)$ is bounded above. If this condition is false, then T^* instances further to the left in the parse trees should have been chosen instead. However no such instances may be available. So far the cycles on the path to T^* have been restricted to those which involve T. Other non-terminals may appear, but always in between T-instances. Another way of expressing the same point is to say that so far T^*, the last instance of T, expands non-cyclically. In general there is an integer p (distinct from any p mentioned before) such that a path from S to a non-terminal T_p which expands non-cyclically is explained as follows. As discussed in and around Lemma 3, the grammar G is in one-to-one correspondence with a directed graph G_R, in which the nodes are the distinct non-terminals and terminals and each arc goes from a non-terminal to a symbol in one of the expansions using a production-rule. Then the path from S to T_p corresponds to a path in G_R from the S node to the T_p node. In general this path in G_R has circuits in it, there being only a finite number of S-to-T_p paths which are without circuits. Circuits cause repetition of non-terminals on the path. Sometimes also there are circuits within circuits, or 'second-level' circuits. For example, the trivial grammar:

$$\begin{array}{lll} S \to M & \left\{ \begin{array}{l} M \to N \\ N \to M \end{array} \right. & N \to N \\ M \to d & & \end{array}$$

yields the paths $S \to M \to N \to M \to \ldots \to M \to d$, where the bracketed rules introduce cycle-instances into the path. However each such cycle may be interrupted by an arbitrary number of second-level $N \leftarrow N$ cycles, thus:

$$S \to M \to N \to N \to M \to N \to N \to N \to M \to d.$$

Again in a more complex grammar a third level of cycle could be introduced, and so on.

To find the first level cycles proceed down the path from S to identify the first non-terminal T_1 which repeats at some later point on the path. (This could of course be S). Move down the path to locate the last instance of T_1. Thereafter begin again to identify the first non-terminal T_2 which repeats later. Now move to the last instance of T_2 and so on until the last instance of T_p is reached, after which no non-terminal is found which later repeats. The sequence S, $T_1, T_2, \ldots T_p$ is certainly finite, because the members of the sequence are distinct and all in G. In a similar way the identities of non-terminals which repeat in second, third and higher-level cycles can be determined, by searching for repeating instances inside respectively first, second and higher-level cycle instances which occur down the path.

Now let the first-level cycle-instances be $T_1 \Rightarrow u_{1s} T_1 v_{1s}$, where s refers to the particular cycle instance (out of the finite number that can arise), followed by $T_2 \Rightarrow u_{2s} T_2 v_{2s}, \ldots$ up to $T_p \Rightarrow u_{ps} T_p v_{ps}$. Denote all higher level cycles occurring within T_i instances by $N_{it} \Rightarrow u_{its} N_{it} v_{its}$. Here i varies indicating the top-level non-terminal, t varies to fix the second or higher-level non-terminal, and s indicates the choice of particular cycle out of the (finite) number which involve N_{it}. Denote by r_{is} the number of times the s'th cycle-variant of T_i ($T_i \Rightarrow u_{is} T_i v_{is}$) occurs on the path, and let r_{its} be the number of times the s'th cycle-variant of N_{it} ($N_{it} \Rightarrow u_{its} N_{it} v_{its}$) occurs in all the instances of T_i cycles. Also let $S \Rightarrow a_1 T_1 b_1$ for the first instance of T_1 on the path, and let $T_{i-1} \Rightarrow a_i T_i b_i$, for the last instance of T_{i-1} and the first of T_i.

A more general theorem can now be given, of which Theorems 21 and 22 are special cases.

6. THE THEOREM FOR A CHAIN OF CYCLES

Theorem 23

Let there be an infinity of non-underflowing strings, each with the property that in its parse-tree there exists a path from S to T_p as described above, where the last T_p instance expands non-cyclically to w, and where in the derivations $S \Rightarrow a_1T_1b_1$, $T_1 \Rightarrow a_2T_2b_2, \ldots T_{i-1} \Rightarrow a_iT_ib_i, \ldots T_{p-1} \Rightarrow a_{2p}T_{2p}b_{2p}$, the lengths $l(a_i)$ and $l(a_ib_i)$ for all i, $1 \leqslant i \leqslant p$, are bounded. Furthermore let each of the r_{is} and r_{its}, the cycle repetition variables, be unbounded in the infinity.

Then there exists a sequence of cycles $S_1 \Rightarrow x_1S_1y_1, \ldots S_i \Rightarrow x_iS_iy_i, \ldots S_p \Rightarrow x_pS_py_p$ for some non-terminals $S_1, \ldots S_i, \ldots S_p$ of G such that for some c_i, d_i, $S_1 \Rightarrow c_2S_2, \ldots S_{i-1} \Rightarrow c_iS_id_i, \ldots S_{p-1} \Rightarrow c_pS_pd_p$, and with the numeric properties that for all i and j, $1 \leqslant i, j \leqslant p$,

$$\sum_{S=1}^{i} l(x_s) > = 0 \quad \text{and} \quad \sum_{S=1}^{P} l(x_s) + \sum_{S=j}^{P} l(y_s) > = 0.$$

Proof

The general scheme is as in Theorem 21. First define the combined left-hand and right-hand lengths of all the T_q portion of the path as

$$L_q = \sum_s r_{qs}\, l(u_{qs}) + \sum_{t,s} r_{qts}\, l(u_{qts}).$$

and $$R_q = \sum_s r_{qs}\, l(v_{qs}) + \sum_{t,s} r_{qts}\, l(v_{qts}).$$

Then let $P_i = \sum_1^i L_q$ for all i, $1 = < i = < p$,

and $Q_j = P_p + \sum_{q=j}^{P} R_q$ for all j, $1 = < j = < p$.

This time the demands for non-underflow give the inequalities:

$$\sum_{q=1}^{i} l(a_q) + P_i \quad > = L, \text{ for } 1 = < \mathrm{i} = < p$$

$$\sum_1^P l(a_q) + l(w) + Q_p \quad > = L$$

$$\sum_1^P l(a_q) + l(w) + \sum_{q=j+1}^{P} (b_q) + Q_j > = L,$$
$$\text{for } 1 \leqslant j \leqslant p-1.$$

Since the $l(a_i)$, $l(b_i)$ and $l(w)$ are given as bounded, it follows that all the P_i and Q_j are bounded below. Also, following Theorem 21,

$$L_q = \sum_s r_{qs}\, U_{qs} + \sum_{s,t} r_{qst}\, U_{qst} + \sum_h k_{qh}\, l_0(C_h), \text{ say,}$$

where the U_{qs} and U_{qst}, as s and t vary, are the distinct lengths of non-cyclic expansions of the cycles involving T_s (and its higher-level included cycles), where the k_{qh} are the numbers of basic cycles C_j involved in the expansion of the whole q'th left-hand side. Similarly let

$$R_q = \sum_s r_{qs}\, v_{qs} + \sum_{s,t} r_{qsr}\, v_{qst} + \sum_h n_{sh}\, l_0(C_h).$$

Now consider the vector:

$X = [r_{11}, r_{12}, \ldots, r_{111}, \ldots, k_{11}, \ldots, n_{11}, \ldots,$
$\ldots r_{i1}, \ldots r_{i11}, \ldots, k_{i1}, \ldots, n_{i1}, \ldots,$
$\ldots$
$\ldots r_{p1}, r_{p2}, \ldots, r_{p11}, \ldots, k_{p1}, \ldots, n_{p1}, \ldots,$
$P_1, P_2, \ldots P_p, Q_1, Q_2, \ldots Q_p].$

Then all the components of X are integral and bounded-below, so that Lemma 13 applies, and there exist strings whose vectors X_1 and X_2 have components with the properties: ${}^1r_{ij} < {}^2r_{ij}$, for all i and j, and ${}^1r_{ijh} < {}^2r_{ijh}$ for all i, j and h, since all the r_{ij} and r_{ijh} are given unbounded above. Similarly ${}^1n_{ij} \leqslant {}^2n_{ij}$ and ${}^1k_{ij} \leqslant {}^2k_{ij}$. The most useful results are that ${}^2P_i - {}^1P_i \geqslant 0$ and ${}^2Q_j - {}^1Q_j \geqslant 0$ for all i and j, and the relations established between the r, k and n variables show that these partial sums always correspond to a constructible system of cycles, as required by the theorem. The S_i iidentify with the T_i, and higher-level cycles, when required, may be inserted into their own T_i cycle to form, for fixed i, a single composed cycle.

As discussed so far the theorem is very restrictive:

1. For each T_s, certain T_s cycles could well arise only a bounded number of times in each of all the infinity of paths considered. There might be a T_s whose total number of cycles was bounded.
2. Assumptions are made about the boundedness of the a_i and i b_i.

The first restriction is removed as follows. The infinity of paths from S to T_p can be regarded as having a common path executed in every case. On this path there are a finite number of points at which an arbitrary number of additional cycles can be inserted. The most obvious form of common path is a non-cyclic one. When admitting cycles with bounded numbers of repetitions, Lemma 13 allows an infinity of strings to be chosen in which the repetitions are constant. These also are admitted to the common path.

A general path in the infinity can thus be considered to have a finite number of 'constant regions' which are the portions of the common path. These constant regions are separated by 'unbounded regions' which are arbitrary insertions of unboundedly repeating cycles, governed only by the restriction of non-underflow. An interesting case is where a first-level T_i has unbounded repetitions, a second-level cycle is bounded, while a third is unbounded. What happens is that the second level instances are added to the common path, further instances of first and third-level cycles being inserted at quite different points on the path. Thus a chain of two cycles is established where in the last theorem one composed cycle would have sufficed.

Now the theorem can be restated with restriction 1 lifted, no further proof being given.

Theorem 24

Let there be an infinity of non-underflowing strings, each with the property that in the parse-tree there exists a path from S to T_p, where the last T_p instance expands to the terminal string w, where l(w) is bounded, and where in the derivations $S \Rightarrow a_1T_1b_1$, $T_1 \Rightarrow a_2T_2b_2, \ldots T_{i-1} \Rightarrow a_iT_ib_i, \ldots T_{p-i} \Rightarrow a_pT_pb_p$, the lengths $l(a_i)$ and $l(a_ib_i)$ are bounded, for all i in $1 \leqslant i \leqslant p$. However this time for some i, T_{i-1} and T_i may be identical non-terminals, as determined in the above discussion, and the expansion $T_{i-1} \Rightarrow a_iT_ib_1$ would then have a number of T_i or other cycles in it, the number being fixed, for this given i, throughout the infinity of strings. The integer p now measures the number of 'unbounded regions' down the path. Then, as before, there exists a finite sequence of cycles $S_1 \Rightarrow x_1S_1y_1, \ldots S_i \Rightarrow x_iS_iy_i, \ldots S_q \Rightarrow x_qS_qy_q$ for some non-terminals $S_1, \ldots S_1, \ldots S_q$ (not necessarily distinct) of G such that for some c_i and d_i, $S_1 = c_2S_2d_2, \ldots S_{i-1} \Rightarrow c_iS_id_i, \ldots S_{q-1} \Rightarrow c_qS_qd_q$ and with the numeric properties that for all i and j, $1 \leqslant i, j \leqslant q$,

$$\sum_{s=1}^{i} l(x_s) > = 0, \text{ and} \sum_{s=1}^{q} l(x_s) + \sum_{s=j}^{q} l(y_s) > = 0.$$

(It must be emphasised that the notions of 'unbounded' and 'constant' regions refer to the numbers of cycles on the path being considered, not to their lengths – ie not to whether any left or right-hand branches are bounded or unbounded).

7. THE THEOREM FOR A TREE OF CYCLES

The second restriction of the last section was that the $l(a_i)$, $l(a_ib_i)$ and l(w) had to be bounded. However in general the a_i etc will have non-terminals which expand unboundedly in the infinity. In all the a_i and b_i there are only a finite number of such non-terminal instances, and the expansion of each can be developed exactly as the expansion of S has been treated so far. Where unboundedness occurs to the left or right, these sub-branches can also be developed similarly. There is only a finite number of branches, each containing only a finite number of unbounded regions.

Consider the order in which terminals of a string whose parse-tree corresponds to the structure are deposited on the stack. When as usual the base-node S is at the top, deposition occurs in the order the terminals are encountered as one inspects the periphery of the tree in an anti-clockwise direction. As in Theorems 21 to 24, in Theorem 25 the concern will be to form a finite list of partial sums of the lengths of the left-hand and right-hand sides of the unbounded regions. Each of these sums will then have a lower bound placed on it, so the partial sums of one, two and more left-hand and right-hand sides of unbounded regions will have to be made from the beginning of deposition, and in 'deposition order'.

Theorem 25

Let there be an infinity of non-underflowing strings which have an identical tree-structure of bounded and unbounded regions. Then there exists an identical (and finite) tree of cycles whose partial sums of left-hand and right-hand lengths, as determined by deposition order, are in no case less than zero.

Proof

If written for a particular tree-structure, the proof would be analogous to Theorem 23, but with a far more unwieldy suffix notation!

8. GENERAL INFINITIES OF NON-UNDERFLOWING STRINGS

In each of the theorems so far only a specific class of non-undeflowing infinities has been treated. The general result is:

Theorem 26

Let there be an infinity of non-underflowing strings. Then there exists a tree of cycles whose partial sums of left-hand and right-hand lengths, as determined by deposition order, are in no case less than zero.

Proof

Most of the full procedure for finding such a cycle-tree structure has already been described. It is now sufficient to show that for any non-underflowing infinity of strings a sub-infinity can be found with a common S to T_p path of constant and unbounded p regions. The previous work shows how this simple result is used repetitively to develop the whole common tree of constant and unbounded regions. Furthermore it is just path-determination that needs discussion below – division of a path into regions has already been described.

Consider in the graph G_R the finite set of all paths (with no R circuits) from S to every non-terminal. Consider also the infinity of all paths, in the given non-underflowing infinity of strings, from S to non-terminals which are expanded non-cyclically. When all cycles are removed from such paths there is sure to be an infinity of the latter paths which correspond exactly with one of the set from G_R. Many such paths from G_R may thus have an infinity associated with it, which is not surprising – just choose one such to determine a suitable infinity of strings.

Theorem 27

There exists an infinity of non-underflowing strings if and only if either:

1. There exists a cycle $N \Rightarrow uNv$, not necessarily basic, for which $u \geqslant 0$ and $u+v = 0$, – or
2. there exists a cycle-tree of depth two, (in which the upper cycle's u, v etc are unsuffixed, the lower cycles being suffixed from left to right), for which $u > 0$ and $u_i < 0$ for all i, $1 \leqslant i \leqslant n$, $R > 1$, $R_i > 1$ for all i, and $R \leqslant R_1R_2 \ldots R_n$, for some $n \geqslant 1$.

Proof

Sufficiency is proved first – if the first condition is true the result follows at once by repetition of the cycle to generate all the infinity of strings required. If the second condition is true then the result follows from Theorem 19. *Necessity*: Theorem 26 shows the existence of a cycle-tree with non-negative partial sums. By constructing equal repetitions of all the cycles in the tree an infinity of different strings are generated, all non-underflowing. Thus the conditions of Theorem 20 are satisfied, which leads to the required result.

9. COMPILE-TIME COMPUTABILITY

The final theorem, the aim of the whole article, can now be presented.

Theorem 28

It is computable by examining the grammar G whether or not an infinity of non-underflowing strings can be generated.

Proof

By Theorem 27 if a cycle or cycle-tree of depth two as described there, can be found in the grammar, then the infinity can be generated. The argument following Theorem 17 shows that only basic cycles need be examined in the search for sufficient conditions. Hence the search is finite, and either succeeds or fails. From Theorem 27 also, failure to detect these simple conditions means that the desired infinity cannot be generated.

10. CONCLUSION

Happily all questions have turned out to be computable.

Acknowledgment

This series of articles would never have been completed without my wife Valerie's enthusiasm and encouragement.

REFERENCES

1. D. T. Goodwin (1977) Conditions for Underflow and Overflow of an Arithmetic Stack, *The Computer Journal* **Vol. 20**, pp.56–62.
2. D. T. Goodwin (1980) Partial Non-Underflow and Non-Overflow of an Arithmetic Stack, *The Computer Journal* **Vol. 23**, pp.153–160.
3. D. T. Goodwin (1985) Further Study of a Stack-Length Model, *The Computer Journal* **Vol. 28**, No.1, pp. 9–16.