Date: 2024-05-03

Turing Machines

Abstract

Developments are here described in the analysis technique for non-terminating Turing Machines (TM's) that I described earlier. The main idea is the introduction of IRR patterns i.e. constraints satisfied by large sets of IRRs (Irreducible Regular Rules) and the logical relationships between them as a result of the general method for deriving IRR's from others described in my earlier paper. These logical relationships will be referred to as IGR's (IRR Generating Rules). IGR's have been reduced to their minimal form in a way analogous to the way in which regular rules were reduced to IRR's by taking out symbol strings that played no essential role. In the case of IGR's these symbol strings (actually pairs) will be referred to as context pairs. A new version of my computer program extending the previous analysis is described and is freely available that generates these IGR's up to a given length of IRR's that they generate. The results show repetition of the left hand halves (Left IGR's or LIGR's) of IGR's associated with different right hand halves. Because the LIGR's can be derived independently of the right hand halves of IGR's, this should be done separately and can be done using the currently known IRR's as previously described in my earlier papers. The LIGR's can be used to calculate all the IRR's of a TM. A procedure for the generation of all the LIGR's for a TM has been suggested and is expressed here by a detailed analysis of a TM though not yet as computer code.

Mathematics Subject Classification: 68Q25

Keywords: Turing Machine (TM), Irreducible Regular Rule (IRR), IRR Generating Rule (IGR), LIGR (Left IGR).

1 Introduction

I have done a lot of work on this since I last updated it and I decided to update it now despite not being entirely consistent because some updates are in progress. Main changes:

Table 95 was reorganised renumbering the LIGR's, and the related text has been consolidated to remove repetitions.

Table 1 shows that Table 95 is incomplete and so is Table 3.

Table 1 can be reorganised (Table 5) in terms of LIGR's and RIGR's(Right IGR's).

This document is a work in progress. As such it is incomplete and still has errors and omissions. When brought to a state where I cannot easily find any improvements it will form my next paper on Turing Machine analysis.

Section 2 is a quite dense summary of the previous methods that lays the foundations of the developments to be described in this paper. Section 3 introduces IRR patterns (IRRP's) as sets of IGR's conforming to the pattern. They have some common symbols in the origin and the RHS of the IRR and allow for any LHS. Section 4 introduces IGR's in terms of IRRP's and illustrates the fact that IRR's of any length can be derived from sequences of IGR's by a sequence of substitutions. In Section 5 the detailed description of an IGR is given and proves the generation of IRR's from IGR's. A computer algorithm is described for generating them all for a TM and its results are shown for an example TM. In Section 6 a necessary condition in the relation "can be followed by" for IGR's is found. Further results are found for the set of IGR's that can follow a sequence of IGR's following each other (i.e. substituted into each other) hand calculation of which suggests a method for generating all of them for a TM thugh this appears practically impossible for the example TM beause of te large number of cases to be considered. In Section 7 left IGR's or LIGR's are introduced because in section 6 the LHS and RHS of an IGR can be developed independently. An algorithm is illustrated by example for finding all the LIGR's for a TM based on the above ideas and results.

A lot of material has been removed to 2017's Notes on Turing Machines. These notes are now mostly superseded, but there may be a little there that is of use.

Comments are welcome. Please send them to john.h.nixon1@gmail.com

2 Basic definitions and summary of the existing method to generate the IRR's for a TM

A configuration set (CS) for a TM is a set of complete configurations (tape symbols with pointer position indicated, and the machine state of the TM) such that the CS is specified by giving a finite set of symbols in a set of contiguous pointer positions together with the machine state and such that the pointer position is where one of the given symbols is given or adjacent to one. In a CS all possible configurations that are consistent with the specified symbols and machine state are included. The notation is the specified symbol string with the pointer indicated by an underscore (it is just off the end of the symbol string) or an underline and the machine state on the left. For example with machine states 1,2,3, etc. and symbols lower case letters the following are CSs: $\texttt{2a}\underline{\texttt{b}}\texttt{ca}$, $\texttt{1\_aabbcac}$. The length of the CS is the length of the symbol string which is finite.

A computation rule or rule is a pair of CS's linked by $\rightarrow$ indicating the forward direction of the computation. A reducible rule is one that has symbols that play no part in the computation i.e. any extra symbols added on the left or right of the strings at the left and right hand sides of a computation rule. From the definitions of regular rules (RR) and irreducible regular rules (IRR) in [1], any computation of the TM that ends with the pointer just off the end (i.e. adjacent to a symbol at the end) of the string of symbols specified at the start can be represented by RR's chained together as a sequence of CS's starting with one of length 1, where for each step in the chain a new symbol is read at a position where no symbol has yet been read at the pointer, thus the length of the string of symbols increases by 1 for each RR unless a stationary cycle occurs that ends such a sequence. All such CS's are by definition reachable. All single TM steps are RR's. If the RR is of type LR or RL as designated earlier (now the position of the pointer in the origin (see below) is included so these are now RLR and LRL respectively) the pointer swaps ends at that step of the chain and these RR's are also irreducible RR's (IRR's) because if the pointer swaps ends there are no redundant symbols i.e. the rule is irreducible. There are also IRR's that that don't swap ends. If a CS called "origin" is included with the LHS and RHS of the IRR it can be written in the triplet form as $\texttt{origin} \rightarrow \texttt{LHS} \rightarrow \texttt{RHS}$ for which the abbeviated form $\texttt{origin} \rightarrow\rightarrow \texttt{RHS}$ will be used if the LHS is not specified hence the changed designation of the type of an IRR. An origin (there could be many for the same LHS) of an IRR is a CS obtained by running the TM backwards starting from the LHS to a point such that the pointer position is at the opposite end of the string from where it is in the LHS.

If an RR is of type LRR or RLL it is related to an irreducible form (a possibly shorter IRR which only involves the symbols passed by the pointer during its execution) as follows. Suppose the RR is represented by $\texttt{m} \rightarrow \texttt{n} \rightarrow \texttt{o}$ where $m < n < o$ and $n + 1 = o$ where italics represent the corresponding pointer positions for the CS's in typewriter font. Then the RR has type LRR because the start and end points of the i.e. the LHS and RHS have the pointer at the right hand end of the string. The rule $\texttt{n} \rightarrow \texttt{o}$ can be represented as $\texttt{n}' \rightarrow \texttt{p}' \rightarrow \texttt{o}'$ without any redundant symbols where $m \leq p \leq n < o$ and the primes indicate shortening of the strings by deleting the symbols below position p i.e. p is the leftmost pointer position in the computation from $\texttt{n}$ to $\texttt{o}$. $n = p$ holds if and only if $\texttt{n}' = \texttt{p}'$ and $\texttt{n}' \rightarrow \texttt{o}'$ is a single TM step. If $n \neq p$ the rule $\texttt{n}' \rightarrow \texttt{p}'$ shows that $\texttt{p}'$ is reachable therefore $\texttt{n}' \rightarrow \texttt{p}' \rightarrow \texttt{o}'$ represents an IRR of type RLR, and of course the mirror image result applies to IRR's of type RLL.

In general let $\texttt{X}$ be a member of IRR(n) i.e. the set of all IRR's with CS's of length n. Then $\texttt{X}$ can be represented as $\texttt{A} \rightarrow \texttt{B} \rightarrow \texttt{C}$ where the pointer swaps ends between $\texttt{A}$ and $\texttt{B}$ (thus this is either $\texttt{1} \rightarrow \texttt{n}$ or $\texttt{n} \rightarrow \texttt{1}$ and is referred to as condition 1) to establish the reachability of `B` necessary for `X` to be an IRR. There may be more than one such CS `A` for a single `B` and the set of all such `A` will be denoted by $\texttt{O}_1(\texttt{B})$ (the same as `S(B)` in [2]), the 1 referring to the backward searching algorithm that terminates in condition 1 (see [2] section 2.2). Likewise if it terminates in condition 2 i.e. the pointer comes back to where it was at the start of the backward search, the set of such endpoints will be denoted as $\texttt{O}_2(\texttt{B})$, but these do not confer reachability and will not be referred to as origins. If the pointer is at the right in `A` and at the left in `B` then at `C` it can be at the left or right so that `X` must be represented as either of the triplet forms $\texttt{n} \to \texttt{1} \to \texttt{0}$ or $\texttt{n} \to \texttt{1} \to \texttt{n}+\texttt{1}$ and having types RLL and RLR respectively. Likewise if the pointer is at the left in `A` (the mirror image forms), `X` must be represented by either of $\texttt{1} \to \texttt{n} \to \texttt{n}+\texttt{1}$ or $\texttt{1} \to \texttt{n} \to \texttt{0}$, having types LRR and LRL respectively.

If `B` is reachable but forward computaton from it leads to a CS that has arisen before in this computation, this is an stationary cycle and the type of the IRR is then of type LC or RC. If the reverse computation path from `B` leads to a stationary cycle, then this cycle must include `B` to avoid a branch point in the forward computation that would not then be unique. Thus likewise the IRR is of type LC or RC.

From the definition of RR in the first paragraph of this section, if in the backward search from the LHS of an IRR, the pointer again reaches the same position it had in the LHS (condition 2), however much further back the backward search were to continue, it would not be possible from this alone to show that this LHS is indeed the LHS of an IRR. This is because if the computation is again run forward, this LHS has the pointer at the same point as a previous CS and is therefore not shown to be one of the list of CS's playing the special role in the above definition, though it could possibly be shown to be one as a result of another backward search path. This is the justification of the terminating condition 2 in the backward search algorithm. See [2] page 30.

This proves that

Lemma 2.1. *The triplets* $\texttt{1} \to \texttt{n} \to \texttt{0}$, $\texttt{1} \to \texttt{n} \to \texttt{n}+\texttt{1}$, *and* $\texttt{n} \to \texttt{1} \to \texttt{n}+\texttt{1}$, *and* $\texttt{n} \to \texttt{1} \to \texttt{0}$ *representing TM computations each form an IRR* (*type LRL, LRR, RLR or RLL respectively*) *if and only if the origin indicated* (*the first member*) *is the first CS arrived at with the pointer in that position after tracing the computation back from the LHS* (*the middle member*) *and the pointer does not occur again in the position it had in the LHS in the reverse computation path from the LHS i.e there is no other CS* `1` *or* `n` *between the* `1` *and the* `n` *in the triplet forms above. Furthermore any IRR of length* n *of one of the types RLL, RLR, LRL, LRR has one of these forms.*

Generating all the IRR's based on Theorem 9.1 of [1] starts with all single TM steps in the above notation $\texttt{1} \to \texttt{0}$ (i.e. $\underline{\texttt{x}} \to \_\texttt{x}$) or $\texttt{1} \to \texttt{2}$ (i.e. $\underline{\texttt{x}} \to \texttt{x}\_$)

where x's represents an arbitrary symbol that could be different for each use. Every possible single symbol (called α) is added at the pointer position in each RHS, and in each case the computaton is taken as far as possible to get the new RHS unless a stationary cycle occurs. The resulting rule $\mathtt{LHS} \to \mathtt{RHS}$ is an IRR if it irreducible i.e. connot be expressed with shorter strings of symbols. In the first case, adding α on the left and continuing the computation as far as possible gives results either of the form (i) $\mathtt{2} \to \mathtt{1} \to \mathtt{3}$ or (ii) $\mathtt{2} \to \mathtt{1} \to \mathtt{0}$ i.e. $\alpha\underline{\mathtt{x}} \to \underline{\alpha}\mathtt{x} \to \mathtt{xx\_}$ or $\mathtt{\_xx}$ respectively unless a stationary cycle is obtained. The results in case (i) are IRR's by Lemma 2.1 because there can be no other CS's between the 2 and the 1 which is a single TM step. The results from (ii) are IRR's if and only if the first move beyond the 1 is to 2 i.e. the computation has the form $\mathtt{2} \to \mathtt{1} \to \mathtt{2} \to \mathtt{0}$ because this ensures that the rule $\mathtt{1} \to \mathtt{0}$ contained in (ii) of length 2 is irreducible. Likewise for the mirror image case starting with a rule of the form $\mathtt{1} \to \mathtt{2}$ adding the α on the right and continuing gives results of the form $\mathtt{1} \to \mathtt{2} \to \mathtt{0}$ ($\in$ IRR(2)) or of the form $\mathtt{1} \to \mathtt{2} \to \mathtt{3}$ ($\in$ IRR(2) if and only if the first move from the 2 is to 1).

Consider extending this to the general case of generating all the IRR Y of length $n+1$ based on the single IRR X of length $n \geq 2$ and having the form $\mathtt{n} \to \mathtt{1} \to \mathtt{n+1}$, which can be also be written as $\mathtt{A} \to \mathtt{B} \to \mathtt{C}$ for some CS's A,B, and C. First the computation $\mathtt{A}\alpha \to \mathtt{B}\alpha \to \mathtt{C}\alpha$ holds where α is any symbol the TM uses. Clearly by Theorems 5.4 and 9.1 of [1] every such IRR Y can be obtained starting from the LHS $\mathtt{B}\alpha$ if an appropriate α can be found. The symbol α must be chosen so that $\mathtt{B}\alpha$ is reachable i.e. $\mathtt{O}_1(\mathtt{B}\alpha) \neq \emptyset$. These are all the terminal CS's of length $n+1$ from the backward searching algorithm starting from $\mathtt{B}\alpha$ and ending in condition (1). Each of these branches has a point where the pointer first reaches n and this CS is $\mathtt{A}\alpha$ because the α has yet played no part, so $\mathtt{O}_1(\mathtt{B}\alpha) = \mathtt{O}_1(\mathtt{A}\alpha)$, thus the backward search algorithm is applied to $\mathtt{A}\alpha$, and identifying all possible values of α i.e. the values of α for which $\mathtt{O}_1(\mathtt{A}\alpha) \neq \emptyset$ by generating all its origins for each such α. Also the forward computation from $\mathtt{C}\alpha$ is continued as far as possible to generate the RHS of Y and hence what its type is (LRR, LRL, RLR, RLL, LRC, or RLC) the last two cases coming from a stationary cycle in the forward computation from $\mathtt{C}\alpha$.

If the pointer is at the left in A and C and the right in B (the mirror image case) the added arbitrary symbol α will be on the left. This procedure for generating the all the IRR Y of length $n+1$ like this from an IRR X of length n, including the mirror image case where the triplet form of X is $\mathtt{1} \to \mathtt{n} \to \mathtt{0}$, will denoted by the function F. F applied to an IRR of type LC or RC is the empty set. This proves that

Theorem 2.2. *Every member of IRR($n+1$) can be obtained using* F *from some* X $\in$ *IRR(n) of type RLR or LRL for $n \geq 2$. Also, because forward computation is unique e.g. the RHS of an IRR is uniquely determined by is*

LHS, but the origins may be more than one, the sets of IRR obtained like this for different `X` *(different* `B`*) must be disjoint i.e.* $\mathtt{F}^{-1}$ *applied to a member of IRR$(n+1)$ is a unique member of IRR(n). The first part can be written symbolically as*

$$\mathit{IRR}(n+1) = \bigcup_{\mathtt{X}\in\mathtt{IRR}(n)} \{\mathtt{F(X)}\} \tag{1}$$

The following is the general outline showing an IRR triplet of length n of type RLR (type RLR with origin having the pointer at the right) and the possible types of result (except the cases where a stationary cycle occurs) of this argument for a given symbol α that could include a new IRR triplet of length $n+1$.

$$\left.\begin{array}{lr}\text{Cannot be used to prove reachability of } \mathtt{1} & \mathtt{1} \rightarrow \\ \text{Proves reachability of } \mathtt{1} & \mathtt{n+1} \rightarrow \end{array}\right\} \mathtt{n} \rightarrow \mathtt{1} \rightarrow \mathtt{n+1} \left\{\begin{array}{ccl}\rightarrow & \mathtt{0} & \text{type RLL}\\ \rightarrow & \mathtt{n+2} & \text{type RLR}\end{array}\right. \tag{2}$$

The 3 central CS's refer to a member of IRR(n), and the leftmost, central, and rightmost CS's refer to the corresponding member of IRR(n+`1`) if reachability of the CS `1` in the centre of (2) is found. The corresponding mirror image result for the LRL case is as follows:

$$\left.\begin{array}{lr}\text{Proves reachability of } \mathtt{n+1} & \mathtt{1} \rightarrow \\ \text{Cannot be used to prove reachability of } \mathtt{n+1} & \mathtt{n+1} \rightarrow \end{array}\right\} \mathtt{2} \rightarrow \mathtt{n{+}1} \rightarrow \mathtt{1} \left\{\begin{array}{ccl}\rightarrow & \mathtt{0} & \text{type LRL}\\ \rightarrow & \mathtt{n+2} & \text{type LRR}\end{array}\right. \tag{3}$$

In this case note that because the α is added on the left, all the pointer positions in the IRR of length n have been increased by 1 when they appear in the embedded IRR of length n, so originally they would have been $\mathtt{1} \rightarrow \mathtt{n} \rightarrow \mathtt{0}$. The above procedure allows the generation of all the IRR of a TM up to any given length and has been implemented [3].

3 IRR patterns (IRRP's) and IGR's

The derivation of IRR's from other ones (length n) following the procedure `F` described above was found to often take the same form independent of n provided n is large enough. Then the obvious step is to describe these general results termed IRR generating rules (IGR's) so that they can be easily applied in any given case. These results have an LHS and an RHS and the existence of a member of IRR(n) matching the LHS implies the existence of a corresponding member of IRR(n+1) matching each of the parts of the RHS. Each of these parts and the LHS take the form of a generalised IRR in which the symbol α appears and two arbitrary strings, $\mathtt{T}_1$ in the origin and $\mathtt{T}_2$ in the RHS, and the

LHS (middle member) is omitted so that any LHS is matched. These general forms for the IRR's were termed IRR patterns (IRRP's).

The analysis techniques were applied to the following TM (4) which was generated randomly with 5 states and 5 symbols. This TM, being much larger than any that I have analysed before, has proved to be a much more challenging case.

$$\begin{array}{lllll}
\mathtt{1\underline{a}} \to \mathtt{2\_d} & \mathtt{2\underline{a}} \to \mathtt{1c\_} & \mathtt{3\underline{a}} \to \mathtt{4c\_} & \mathtt{4\underline{a}} \to \mathtt{3\_b} & \mathtt{5\underline{a}} \to \mathtt{2\_e} \\
\mathtt{1\underline{b}} \to \mathtt{4\_d} & \mathtt{2\underline{b}} \to \mathtt{4\_c} & \mathtt{3\underline{b}} \to \mathtt{4\_c} & \mathtt{4\underline{b}} \to \mathtt{4b\_} & \mathtt{5\underline{b}} \to \mathtt{3\_e} \\
\mathtt{1\underline{c}} \to \mathtt{3\_a} & \mathtt{2\underline{c}} \to \mathtt{1d\_} & \mathtt{3\underline{c}} \to \mathtt{2\_a} & \mathtt{4\underline{c}} \to \mathtt{3c\_} & \mathtt{5\underline{c}} \to \mathtt{3a\_} \\
\mathtt{1\underline{d}} \to \mathtt{2b\_} & \mathtt{2\underline{d}} \to \mathtt{1a\_} & \mathtt{3\underline{d}} \to \mathtt{5\_c} & \mathtt{4\underline{d}} \to \mathtt{5\_c} & \mathtt{5\underline{d}} \to \mathtt{4\_a} \\
\mathtt{1\underline{e}} \to \mathtt{2b\_} & \mathtt{2\underline{e}} \to \mathtt{3\_c} & \mathtt{3\underline{e}} \to \mathtt{3b\_} & \mathtt{4\underline{e}} \to \mathtt{5a\_} & \mathtt{5\underline{e}} \to \mathtt{3a\_}
\end{array} \tag{4}$$

For example it is known that at least one IRR for this TM matches

$$\mathtt{1\underline{d}aT}_1 \to\!\!\to \mathtt{4\_caT}_2. \tag{5}$$

Using backward TM steps from (4) gives

$$\begin{array}{llll}
\text{deriving the origin} & \text{old RHS} & \text{RHS} & \alpha \\
\alpha\mathtt{A} & \alpha\mathtt{C} & & \\
\mathtt{1}\alpha\mathtt{\underline{d}aT}_1 \left\{ \begin{array}{l} \overset{\alpha=\mathtt{a}}{\longleftarrow} \mathtt{2\underline{d}daT}_1 \\ \overset{\alpha=\mathtt{c}}{\longleftarrow} \mathtt{2\underline{a}daT}_1 \\ \overset{\alpha=\mathtt{d}}{\longleftarrow} \mathtt{2\underline{c}daT}_1 \end{array} \right. & \begin{array}{l} \mathtt{4\underline{a}caT}_2 \\ \mathtt{4\underline{c}caT}_2 \\ \mathtt{4\underline{d}caT}_2 \end{array} & \begin{array}{l} \mathtt{3\_bcaT}_2 \\ \mathtt{1abc}\underline{\mathtt{T}_2} \\ \mathtt{5\_ccaT}_2 \end{array} & \begin{array}{l} \mathtt{a} \\ \mathtt{c} \\ \mathtt{d} \end{array}
\end{array} \tag{6}$$

of which the result for $\alpha = \mathtt{c}$ has an RHS given where the pointer is at the first symbol of $\mathtt{T}_2$. The results of $\mathtt{F}$ are written as follows

$$\begin{array}{l}
\mathtt{2\underline{d}daT}_1 \to\!\!\to \mathtt{3\_bcaT}_2 \\
\mathtt{2\underline{a}daT}_1 \to\!\!\to \mathtt{1abc}\underline{\mathtt{T}_2} \\
\mathtt{2\underline{c}daT}_1 \to\!\!\to \mathtt{5\_ccaT}_2
\end{array} \tag{7}$$

for $\alpha = \mathtt{a}, \mathtt{c}, \mathtt{d}$ respectively, and the complete IGR can be written as

$$\mathtt{1\underline{d}aT}_1 \to\!\!\to \mathtt{4\_caT}_2 \left\{ \begin{array}{l}
\overset{\mathtt{a}}{\Rightarrow} \mathtt{2\underline{d}daT}_1 \to\!\!\to \mathtt{3\_bcaT}_2 \\
\overset{\mathtt{c}}{\Rightarrow} \mathtt{2\underline{a}daT}_1 \to\!\!\to \mathtt{1abc}\underline{\mathtt{T}_2} \\
\overset{\mathtt{d}}{\Rightarrow} \mathtt{2\underline{c}daT}_1 \to\!\!\to \mathtt{5\_ccaT}_2
\end{array} \right. \tag{8}$$

Note that in this argument, adding the arbitrary symbol α on the left (because the pointer in the origin is on the left) maintains the pointer being on the right hand end of the string of symbols in the LHS (not shown), and this property is implicit in an IRRP with the pointer at the left of the origin CS. The result of this argument is that if an IRR of length n conforms to (5), then there are 3 more IRR's of length $n + 1$ corresponding to (7) for $\alpha \in \{\mathtt{a}, \mathtt{c}, \mathtt{d}\}$ respectively. The second of these results in (7) has the pointer in the RHS on the right, so this IRR has type LRR and cannot be used to derive other IRR's. These are examples of rules that generate IRR's of length $n+1$ from other IRR's of length n. An IGR is defined as a logical implication having an IRRP on the left and sets of IRRP's on the right, one set for each value of α and the logical deduction follows the general procedure outlined in Theorem 2.2. Thus the strings with given symbols on the right of the implications are one symbol longer than those on the left.

To illustrate how an IRR can be derived from a member of IRR(2) and a sequence of IGR's, consider the following IRR of length 6 that was chosen from the computer program output and represented in Origin $\to$ LHS $\to$ RHS format as follows:

$$\mathtt{3\underline{a}ecccb} \to \mathtt{1cadbd\underline{b}} \to \mathtt{2dbdbdb\_}. \tag{9}$$

The derivation of the first rule of (9) in single TM steps is

$$\begin{array}{l}
\mathtt{3\underline{a}ecccb}\\
\mathtt{4c\underline{e}cccb}\\
\mathtt{5ca\underline{c}ccb}\\
\mathtt{3caa\underline{c}cb}\\
\mathtt{2ca\underline{a}acb}\\
\mathtt{1cac\underline{a}cb}\\
\mathtt{2ca\underline{c}dcb}\\
\mathtt{1cad\underline{d}cb}\\
\mathtt{2cadb\underline{c}b}\\
\mathtt{1cadbd\underline{b}}
\end{array}. \tag{10}$$

Each time the pointer moves to where it has not been before while going backwards from the LHS, the derivation (10) generates IRR's as follows, followed by (9) in triplet and the abbreviated notation:

$$\begin{array}{ll}
\mathtt{2\underline{c}b} \to \mathtt{1d\underline{b}} \to \mathtt{5\_cd} \in \mathtt{IRR(2)} & \mathtt{2\underline{c}b} \to\to \mathtt{5\_cd}\\
\mathtt{1\underline{d}cb} \to \mathtt{1bd\underline{b}} \to \mathtt{3\_ecd} \in \mathtt{IRR(3)} & \mathtt{1\underline{d}cb} \to\to \mathtt{3\_ecd}\\
\mathtt{2\underline{c}dcb} \to \mathtt{1dbd\underline{b}} \to \mathtt{5\_cecd} \in \mathtt{IRR(4)} & \mathtt{2\underline{c}dcb} \to\to \mathtt{5\_cecd}\\
\mathtt{4\underline{e}cccb} \to \mathtt{1adbd\underline{b}} \to \mathtt{2\_ececd} \in \mathtt{IRR(5)} & \mathtt{4\underline{e}cccb} \to\to \mathtt{2\_ececd}
\end{array}. \tag{11}$$

This splitting up of the derivation of (9) results from the repeated application of `F` to (11).1. The abbreviated forms in (11) can be obtained by applying in order the following results to the first of these IRR's $\mathtt{2\underline{c}b} \to\to \mathtt{5\_cd}$.

$$\begin{array}{l}
\texttt{2}\underline{\texttt{T}}_1 \twoheadrightarrow \texttt{5\_T}_2 \overset{\texttt{b}}{\Rightarrow} \left.\begin{array}{l}\texttt{1}\underline{\texttt{d}}\texttt{T}_1\\ \texttt{1}\underline{\texttt{e}}\texttt{T}_1\end{array}\right\} \twoheadrightarrow \texttt{3\_eT}_2 \\[2ex]
\texttt{1}\underline{\texttt{T}}_1 \twoheadrightarrow \texttt{3\_T}_2 \left\{\begin{array}{l}\overset{\texttt{a}}{\Rightarrow} \texttt{2}\underline{\texttt{d}}\texttt{T}_1 \twoheadrightarrow \texttt{4c}\underline{\texttt{T}}_2\\ \overset{\texttt{c}}{\Rightarrow} \texttt{2}\underline{\texttt{a}}\texttt{T}_1 \twoheadrightarrow \texttt{2\_aT}_2\\ \overset{\texttt{d}}{\Rightarrow} \texttt{2}\underline{\texttt{c}}\texttt{T}_1 \twoheadrightarrow \texttt{5\_cT}_2\end{array}\right. \\[2ex]
\texttt{2}\underline{\texttt{c}}\texttt{dT}_1 \twoheadrightarrow \texttt{5\_T}_2 \overset{\texttt{a}}{\Rightarrow} \left.\begin{array}{l}\texttt{4}\underline{\texttt{e}}\texttt{ccT}_1\\ \texttt{4}\underline{\texttt{e}}\texttt{ecT}_1\\ \texttt{5}\underline{\texttt{c}}\texttt{adT}_1\\ \texttt{5}\underline{\texttt{e}}\texttt{adT}_1\end{array}\right\} \twoheadrightarrow \texttt{2\_eT}_2 \\[2ex]
\texttt{4}\underline{\texttt{T}}_1 \twoheadrightarrow \texttt{2\_eT}_2 \overset{\texttt{c}}{\Rightarrow} \left.\begin{array}{l}\texttt{3}\underline{\texttt{a}}\texttt{T}_1\\ \texttt{4}\underline{\texttt{b}}\texttt{T}_1\end{array}\right\} \twoheadrightarrow \texttt{2db}\underline{\texttt{T}}_2
\end{array} \quad . \tag{12}$$

For the initial steps in the derivation of (9), the following subcases of successive members of (12) need to be applied in this order: 1, 3, 1, 1.

Equation (12) contains examples of IGR's which allow one IRR to be derived from another by substituting for the $\texttt{T}_1$ and $\texttt{T}_2$ as the example shows, and express the application of the function $\texttt{F}$ in Theorem 2.2 in a simpler form. The IGR's have two lengths, one associated with $\texttt{T}_1$ and one associated with $\texttt{T}_2$ and these are defined as the lengths of the corresponding strings on the RHS of the IGR thus for example the lengths of the IGR's in (12) will be donoted by $(1,1),(1,1),(3,1),(1,2)$ respectively. The equations (12) are in the shortest forms possible as can be verified from their derivations.

The symbols above the implication signs are the symbol added next to the pointer in the origin (α) in the derivation of the IRR's from other ones as described in Section 2, and are the first 4 symbols of the LHS of IRR (9) taken in reverse order. The results on the right in (12) are all the results that can be derived from their LHS for that value of α and that length, though the third example is quite complicated and has other values of α ie. $\texttt{b}$ and $\texttt{c}$ with different lengths of results.

The IRRP on the RHS of the last member of (12) is of type LRR and can be seen to not generate a new IRR directly. Applying the last member of (12) to the last IRR of (11) gives the initial result

$$\texttt{3}\underline{\texttt{a}}\texttt{ecccb} \rightarrow \texttt{1cadbd}\underline{\texttt{b}} \rightarrow \texttt{2dbc}\underline{\texttt{c}}\texttt{ecd} \tag{13}$$

which is not an IRR. Taking this computation as far a possible has to be an IRR (in this case having non-extendable type LRR) which is

$$\texttt{3}\underline{\texttt{a}}\texttt{ecccb} \twoheadrightarrow \texttt{2dbdbdb\_} \tag{14}$$

and has $\alpha = \texttt{c}$ and is in agreement with (9). The derivations of (11) and (14) illustrate the general procedure for deriving any IRR by repeated applications of $\texttt{F}$ i.e. applying a sequence of IGR's starting from a member of IRR(2).

This example suggests that if all the IGR's needed to generate the IRR($n+1$) from IRR(n) were obtained, these could have lengths much less than $n+1$ and be fewer in number than the IRR($n+1$), and this might give a more compact way to represent the action of the TM. This will be followed up later, but many details need to be given first.

Every IGR represents the process of deriving a member of IRR($\mathtt{n}+\mathtt{1}$) from a member of IRR($\mathtt{n}$). Therefore every such IGR can be obtained from another IGR (representing the process for deriving the member of IRR($\mathtt{n}$) from a member of IRR($\mathtt{n}-\mathtt{1}$)) by an appropriate specialisation by adding the context symbols, applying $\mathtt{F}$, then removing any redundant symbols as before. But the number of such context symbols needed seems to be unlimited.

4 General Definition of IGR's

The general form of the derivation of an IRR from an existing one ($\mathtt{F}$) can be expressed in detail as follows. Start with the IRR pattern (IRRP) of type LRL

$$\mathtt{t}_1\underline{\mathtt{y}_1}\ldots\mathtt{y}_n\mathtt{T}_1 \rightarrow\rightarrow \mathtt{t}_2\_\mathtt{z}_1\ldots\mathtt{z}_n\mathtt{T}_2 \tag{15}$$

in which $\mathtt{T}_1$ and $\mathtt{T}_2$ have been omitted for brevity in much of this section. Here $n \geq 2$ and the $\mathtt{t}$'s are machine states and $\mathtt{y}$'s and $\mathtt{z}$'s are symbols.

Then proceed with $\mathtt{F}$ i.e. add the symbol α to both sides where the pointer is in the RHS then the backward search gives the following types of results (excluding the stationary cycles) which can be classified according to the rightmost position j_1 of the pointer relative to the symbol $\mathtt{y}_1$

$$\mathtt{t}_1\alpha\underline{\mathtt{y}_1}\ldots\mathtt{y}_n \leftarrow \begin{cases} \mathtt{t}'_1\underline{\alpha'}\mathtt{y}_1\ldots\mathtt{y}_n \text{ for } j_1 = 0 \\ \mathtt{t}'_1\underline{\alpha'}\mathtt{y}'_1\ldots\mathtt{y}'_{j_1+1}\mathtt{y}_{j_1+2}\ldots\mathtt{y}_n \text{ for } 1 \leq j_1 \leq n-2 \\ \mathtt{t}'_1\alpha\mathtt{y}'_1\ldots\mathtt{y}'_{n-1}\underline{\mathtt{y}'_n} \text{ for } j_1 = n-1 \end{cases} \tag{16}$$

where the primes indicate a possible change in the symbol or state by the TM. For the case $n=1$, j_1 must be 0. Note that the form $\mathtt{t}'_1\underline{\alpha'}\mathtt{y}'_1\mathtt{y}_2\ldots\mathtt{y}_n$ cannot arise because a single backward step to the right followed by two backward steps to the left could possibly alter $\mathtt{y}_1$ and $\mathtt{y}_2$ whereas a single backward step to the left has $j_1 = 0$ as above.

The point of the classification is to enumerate all the different types of case that can arise after all the symbols that are not altered in the derivation are abstracted out. They are not mentioned explicitly and they form part of an arbitrary string (in this case $\mathtt{T}_1$). The last reverse computation step in the last case giving $j_1 = n-1$ cannot not lead to a new IRR because this path and the CS reached does not imply the reachability of the LHS and so does not generate an IRR. If the LHS is reachable it must be because there is another origin with $j_1 < n-1$. Therefore this case must be omitted for the purpose of generating IGR's, so j_1 can be restricted to the range $0 \leq j_1 \leq n-2$.

Similarly, for the computation of the new RHS, the results can be classified (again excluding stationary cycles) by the rightmost position j_2 of the pointer. So that this parameter also starts at 0, the pointer starts at position 0 at α and ends at position -1 if it goes left and ends at position $n+1$ if it goes right giving the possibilities

$$\mathtt{t}_2\underline{\alpha}\mathtt{z}_1 \ldots \mathtt{z}_n \rightarrow \begin{cases} \mathtt{t}'_2\underline{\ }\alpha'\mathtt{z}'_1 \ldots \mathtt{z}'_{j_2}\mathtt{z}_{j_2+1} \ldots \mathtt{z}_n \text{ where } 0 \leq j_2 \leq n \text{ or} \\ \mathtt{t}'_2\alpha'\mathtt{z}'_1 \ldots \mathtt{z}'_n\underline{\ } \text{ if } j_2 = n+1 \end{cases} . \tag{17}$$

This works whenever $n \geq 1$.

If a stationary cycle occurred in (16) it would be noted, but it would have no effect on the general form of the possible results except that none of the forms of endpoint in (16) might result, because a stationary cycle would result in a closed circuit in the reverse search path from which paths ending in a type of endpoint in (16) or none could diverge. As noted earlier this would imply $\mathtt{t}_1\alpha\underline{\mathtt{y}_1} \ldots \mathtt{y}_n$ is in the closed circuit (to avoid a branch point in the forward computation implying it is not unique) so the derived IRR would have type RC. This implies a stationary cycle in the result of (17).

The minimum number of symbols needed for the representation of (16) is easily seen to be

$$r_1 = \begin{cases} 1 & \text{for } j_1 = 0 \\ j_1 + 2 & \text{otherwise} \end{cases} \tag{18}$$

provided $0 \leq j_1 \leq n-2$. Similarly, the minimum number of symbols needed for the representation of the result of (17) is

$$r_2 = \min(j_2 + 1, n + 1). \tag{19}$$

The length of an IGR consists of the pair (r_1, r_2).

From (16) and (17) the remaining four combinations can be summarised as

$$\begin{gathered} \left\{ \begin{matrix} \mathtt{t}_1\underline{\mathtt{T}_1} \\ \mathtt{t}_1\underline{\mathtt{y}}_1 \ldots \mathtt{y}_{j_1+1}\mathtt{T}_1 \end{matrix} \right\} \rightarrow\rightarrow \left\{ \begin{matrix} \mathtt{t}_2\underline{\ }\mathtt{z}_1 \ldots \mathtt{z}_{j_2}\mathtt{T}_2 \\ \mathtt{t}_2\underline{\ }\mathtt{z}_1 \ldots \mathtt{z}_n\mathtt{T}_2 \end{matrix} \right\} \overset{\alpha}{\Rightarrow} \\ \left\{ \begin{matrix} \mathtt{t}'_1\underline{\alpha'}\mathtt{T}_1 \\ \mathtt{t}'_1\underline{\alpha'}\mathtt{y}'_1 \ldots \mathtt{y}'_{j_1+1}\mathtt{T}_1 \end{matrix} \right\} \rightarrow\rightarrow \left\{ \begin{matrix} \mathtt{t}'_2\underline{\ }\alpha'\mathtt{z}'_1 \ldots \mathtt{z}'_{j_2}\mathtt{T}_2 \\ \mathtt{t}'_2\alpha'\mathtt{z}'_1 \ldots \mathtt{z}'_n\underline{\mathtt{T}_2} \end{matrix} \right\} \end{gathered} . \tag{20}$$

In this statement the top and bottom parts on the left of $\rightarrow\rightarrow$ can be combined independently with the top and bottom parts on the right of $\rightarrow\rightarrow$ i.e. there are four combinations possible. Of these the distinctions on the left of $\rightarrow\rightarrow$ do not change the type of the new IRR this being respectively LRL and LRR for the top and bottom parts on the right of `to` $\rightarrow$. The IRR"s are also distinguished by different pairs (j_1, j_2). The type of an IGR is defined as the type of the IRR that it generates.

The corresponding right-left reversed results starting from an IRRP of type RLR also involve the parameters j_1 and j_2 obtained similarly but counting leftwards. Thus starting from

$$\mathtt{t}_1\mathtt{y}_n \dots \underline{\mathtt{y}_1} \rightarrow\rightarrow \mathtt{t}_2\mathtt{z}_n \dots \mathtt{z}_{1\text{-}} \tag{21}$$

likewise the following types of results are obtained which can be classified according to the leftmost position relative to $\mathtt{y}_1$, (j_1) of the pointer. This satisfies $0 \leq j_1 \leq n-1$ and gives the following:

$$\mathtt{t}_1\mathtt{y}_n \dots \underline{\mathtt{y}_1}\alpha \leftarrow \begin{cases} \mathtt{t}'_1\mathtt{y}_n \dots \mathtt{y}_1\underline{\alpha'} \text{ for } j_1 = 0 \\ \mathtt{t}'_1\mathtt{y}_n \dots \mathtt{y}_{j_1+2}\mathtt{y}'_{j_1+1} \dots \mathtt{y}'_1\underline{\alpha'} \text{ for } 1 \leq j_1 \leq n-2 \\ \mathtt{t}'_1\underline{\mathtt{y}'_n}\mathtt{y}'_{n-1} \dots \mathtt{y}'_1\alpha \text{ for } j_1 = n-1 \end{cases} \tag{22}$$

Naturally, (18) and (19) and are still valid and all the types of result in (20) have corresponding mirror image forms.

These types of result in (20) are expressed with the shortest strings of symbols possible (i.e. the $\mathtt{y}$'s and $\mathtt{z}$'s). The strings $\mathtt{T}_1$ and $\mathtt{T}_2$ being arbitrary, so can be replaced by any strings. They do not have to have the same length.

These together with their left-right reversed forms are all the different types of IGR's possible.

Simple examples of these are in (12), and (16) and (17) indicate the general method for deriving them which is as follows. After the symbol α has been added to the origins on the left, reverse steps of the TM are made recursively, making sure that all possible reverse steps at each stage are done and stopping only when further reverse steps are impossible without the knowledge of what the strings $\mathtt{T}_1$ and $\mathtt{T}_2$ are, as described in Section 2.

Thus an IGR is defined to have no redundant symbols where the pointer does not reach during its derivation. This is analogous to IRR's being irreducible. In the derivation of the IGR from an IRR of length n, the backward search to obtain the new origins and in the forward computation to obtain the new RHS, the pointer can obviously never move outside the strings of lengths r_1 and r_2 introduced above except for the last TM step in the forward computation. In addition all these positions of the pointer are reached during the derivation, the string of length r_1 for the derivation of a new origin and the string of length r_2 for the derivation of the new RHS.

If the pointer ends up at one end of the string $\mathtt{T}_2$ (indicated by $\underline{\mathtt{T}_2}$), the pointer position is clear from the context. The pair of strings of symbols $(\mathtt{T}_1, \mathtt{T}_2)$ of lengths $(n+1-r_1, n+1-r_2)$ respectively in (20) that are not passed by the pointer during the derivation of an IGR from an IRR of length

n that is the basis of its LHS will be removed and listed as "context pairs" so that the result is presented in its minimal form i.e. as an IGR in computer output.

A IGR could be defined to include all the possible results that can be derived for any possible value of α (an IGR member), i.e. all the possible origins for each α, but if there is not likely to be confusion I will refer to such statements as IGR's as was done above. Thus an IGR would be the union over α of the IGR members. An IGR member has the form (IRRP,α) $\Rightarrow$ set of IRRP's, so the above results in (12) could be described as IGR members. Thus it would be possible for different RHS's of the IGR to have different values of (r_1, r_2) corresponding to different values of α, but these will be separated into different IGR's in the computer output.

There can be a problem that occurs in the computer representation of the IGR's after the context strings have been separated out, which is to determine whether the original IRRP on its left is of type LRL or RLR. Provided $n > \mathtt{1}$, it is not immediately obvious which is the case because the pointer positions and the parameter j_1 can be counted going either way, for example compare (16) with (22). The way it works is that a CS in the computer program output is represented as $\mathtt{CS}(\mathtt{t}, \mathtt{p}, \mathtt{l}, \mathtt{string})$ where $\mathtt{t}$ is the machine state, $\mathtt{p}$ is the pointer position counted from the left and is one for the symbol on the left, and is 0 for the position just to the left of this symbol, and is $\mathtt{l}+1$ for the position just to the right of the string, where $\mathtt{l} = n$ is the length of the string. The string is spelled out inside quote marks in printed output. After the context strings have been split out of the derived IGR, the pointer position in the origin of the IRRP set on the LHS of (15) is 1 by convention if the original IRRP (see (16)) (the LHS of the new IGR) was of type LRL or LRR because the pointer starts at 1 and is not affected by the truncation of the symbols from the right. If the original IRRP was of type RLR or RLL, the pointer position in its origin (LHS of (21)) is initially by convention at n (i.e. the right hand end) and is reduced as a result of splitting out the context symbols. This for $j_1 = 0$ is position n minus the length of the string of symbols removed also n i.e. 0, and is n minus the length of $\mathtt{y}_n \dots \mathtt{y}_{j_1+2}$ otherwise, which is $j_1 + 1$. This value can never be 1, so the value 1 is characteristic of the original IGR being of type LRL. This implies that the value $\mathtt{p} = 1$ in an origin CS on the LHS of an IGR indicates, provided $n > 1$, that the context strings ($\mathtt{T}_1$ and $\mathtt{T}_2$) are added on the right, and on the left otherwise. For the case $n = 1$ this is obvious from the RHS of the IRRP on the LHS of the IGR. which is of the form $\mathtt{t}_{2\_}\mathtt{z}_1$ or $\mathtt{t}_2\mathtt{z}_{1\_}$ according to whether the IGR is of type LRL or RLR respectively. This shows that this obvious convention for defining the pointer positions in the different cases distinguishes the LRL, LRR from the RLR, RLL types of IGR.

The above argument shows, when combined with Theorem 2.2, that
(1) every IRR of length $n + 1$ of type RL can be derived by $\mathtt{F}$ from another

IRR of length n of type LRL by an IGR with parameters (r_1, r_2) of type (20).1 (LRL (20)) in satisfying $1 \leq r_1 \leq n$ and $1 \leq r_2 \leq n+1$ as described and
(2) every IRR of length $n+1$ of type LRR can be derived by `F` from another IRR of length n of type LRL by an IGR of type (20).2 (LRR in (20)) (with parameters r_1 and r_2 such that $1 \leq r_1 \leq n$ and $r_2 = n+1$.

These can be applied recursively to show that

Theorem 4.1. *any extendable IRR (type LRL or RLR) of length ≥ 3 can be obtained from a member of IRR(2) by a sequence of substitutions of IGR's as described here under case (1). Any non-extendable IRR (type RLL or LRR) can be obtained from a member of IRR(2) by the above substitutions (0 or more) followed by a single substitution step under case (2).*

This theorem is illustrated by the example at the beginning of this section. This suggests the obvious process for generating the set of all the IGR's could start as follows after finding all the members of IRR(2). Essentially this was the method used in the latest version of the program [5] to generate Table 1.

Find all the members of IRR(3) and the IGR's used to generate them from the IRR(2). These will be IGR's of lengths $(1,1)$, $(1,2)$, $(1,3)$, $(2,1)$, $(2,2)$, and $(2,3)$. Likewise the IRR(4) can be obtained from the IRR(3) and the IGR's summarising this can be added while not duplicating any IGR's already found etc.. This can be repeated to generate up to the IRR(n). After a while hopefully to generate the IRR($n+1$) from the IRR(n) will not require any IGR's that have not already been obtained for n sufficiently large.

In the remainder of the paper the following example was studied because the results from (4) became very complicated.

$$\begin{aligned}
&\texttt{1}\underline{\texttt{a}} \to \texttt{2b\_}\\
&\texttt{1}\underline{\texttt{b}} \to \texttt{3\_b}\\
&\texttt{1}\underline{\texttt{c}} \to \texttt{1b\_}\\
&\texttt{2}\underline{\texttt{a}} \to \texttt{3b\_}\\
&\texttt{2}\underline{\texttt{b}} \to \texttt{2c\_}\\
&\texttt{2}\underline{\texttt{c}} \to \texttt{1\_c}\\
&\texttt{3}\underline{\texttt{a}} \to \texttt{1\_a}\\
&\texttt{3}\underline{\texttt{b}} \to \texttt{1\_a}\\
&\texttt{3}\underline{\texttt{c}} \to \texttt{3c\_}
\end{aligned} \tag{23}$$

The results for the IGR's from TM 23 were as follows in Table 1 giving the maximum length of the computation rules as 10.

Table 1: IGR's generated by the computer program

1	$1\underline{T_1} \to\!\!\to 1\_T_2 \overset{\underline{b}}{\Rightarrow} 1\underline{c}T_1 \to\!\!\to 3\_bT_2$
2	$1\underline{c}aT_1 \to\!\!\to 1\_T_2 \overset{\underline{b}}{\Rightarrow} \left.\begin{matrix}2\underline{a}ca\\2\underline{a}cb\end{matrix}\right\}T_1 \to\!\!\to 3\_bT_2$
3	$1\underline{c}aaT_1 \to\!\!\to 1\_T_2 \overset{\underline{b}}{\Rightarrow} \left.\begin{matrix}1\underline{a}baa\\1\underline{a}bab\end{matrix}\right\}T_1 \to\!\!\to 3\_bT_2$
4	$1\underline{c}ababT_1 \to\!\!\to 1\_T_2 \overset{\underline{b}}{\Rightarrow} 1\underline{a}bbccbT_1 \to\!\!\to 3\_bT_2$
5	$1\underline{c}ababcT_1 \to\!\!\to 1\_T_2 \overset{\underline{b}}{\Rightarrow} 1\underline{a}bbccacT_1 \to\!\!\to 3\_bT_2$
6	$1\underline{c}abcT_1 \to\!\!\to 1\_T_2 \overset{\underline{b}}{\Rightarrow} \left.\begin{matrix}2\underline{a}ccac\\2\underline{a}ccbc\end{matrix}\right\}T_1 \to\!\!\to 3\_bT_2$
7	$1\underline{c}cT_1 \to\!\!\to 1\_T_2 \overset{\underline{b}}{\Rightarrow} 1\underline{a}bcT_1 \to\!\!\to 3\_bT_2$
8	$2\underline{T_1} \to\!\!\to 1\_T_2 \overset{\underline{b}}{\Rightarrow} 1\underline{a}T_1 \to\!\!\to 3\_bT_2$
9	$3\underline{T_1} \to\!\!\to 1\_T_2 \overset{\underline{b}}{\Rightarrow} 2\underline{a}T_1 \to\!\!\to 3\_bT_2$
10	$3\underline{T_1} \to\!\!\to 1\_aT_2 \overset{\underline{c}}{\Rightarrow} 3\underline{c}T_1 \to\!\!\to 2bb\underline{T_2}$
11	$1\underline{c}cT_1 \to\!\!\to 1\_ababT_2 \overset{\underline{c}}{\Rightarrow} 2\underline{b}bcT_1 \to\!\!\to 3bbbbb\underline{T_2}$
12	$1\underline{c}aT_1 \to\!\!\to 1\_ababaT_2 \overset{\underline{c}}{\Rightarrow} \left.\begin{matrix}3\underline{c}ca\\3\underline{c}cb\end{matrix}\right\}T_1 \to\!\!\to 3\_bababaT_2$
13	$1\underline{c}aaT_1 \to\!\!\to 1\_ababaT_2 \overset{\underline{c}}{\Rightarrow} \left.\begin{matrix}2\underline{b}baa\\2\underline{b}bab\end{matrix}\right\}T_1 \to\!\!\to 3\_bababaT_2$
14	$1\underline{c}ababT_1 \to\!\!\to 1\_ababaT_2 \overset{\underline{c}}{\Rightarrow} 2\underline{b}bbccbT_1 \to\!\!\to 3\_bababaT_2$
15	$1\underline{c}ababcT_1 \to\!\!\to 1\_ababaT_2 \overset{\underline{c}}{\Rightarrow} 2\underline{b}bbccacT_1 \to\!\!\to 3\_bababaT_2$
16	$1\underline{c}abcT_1 \to\!\!\to 1\_ababaT_2 \overset{\underline{c}}{\Rightarrow} \left.\begin{matrix}3\underline{c}cccac\\3\underline{c}ccbc\end{matrix}\right\}T_1 \to\!\!\to 3\_bababaT_2$
17	$1\underline{c}cT_1 \to\!\!\to 1\_ababaT_2 \overset{\underline{c}}{\Rightarrow} 2\underline{b}bcT_1 \to\!\!\to 3\_bababaT_2$
18	$2\underline{T_1} \to\!\!\to 1\_ababaT_2 \overset{\underline{c}}{\Rightarrow} 2\underline{b}T_1 \to\!\!\to 3\_bababaT_2$
19	$1\underline{c}cT_1 \to\!\!\to 1\_ababcT_2 \overset{\underline{c}}{\Rightarrow} 2\underline{b}bcT_1 \to\!\!\to 3bbbbbc\underline{T_2}$
20	$1\underline{c}aT_1 \to\!\!\to 1\_abcT_2 \overset{\underline{c}}{\Rightarrow} \left.\begin{matrix}3\underline{c}ca\\3\underline{c}cb\end{matrix}\right\}T_1 \to\!\!\to 1bbbb\underline{T_2}$
21	$2\underline{T_1} \to\!\!\to 1\_cT_2 \overset{\underline{c}}{\Rightarrow} 2\underline{b}T_1 \to\!\!\to 1bb\underline{T_2}$
22	$1\underline{T_1} \to\!\!\to 3\_T_2 \overset{\underline{b}}{\Rightarrow} 1\underline{c}T_1 \to\!\!\to 1\_aT_2$
23	$1\underline{c}aT_1 \to\!\!\to 3\_T_2 \overset{\underline{b}}{\Rightarrow} \left.\begin{matrix}2\underline{a}ca\\2\underline{a}cb\end{matrix}\right\}T_1 \to\!\!\to 1\_aT_2$
24	$1\underline{c}aaT_1 \to\!\!\to 3\_T_2 \overset{\underline{b}}{\Rightarrow} \left.\begin{matrix}1\underline{a}baa\\1\underline{a}bab\end{matrix}\right\}T_1 \to\!\!\to 1\_aT_2$
25	$1\underline{c}ababT_1 \to\!\!\to 3\_T_2 \overset{\underline{b}}{\Rightarrow} 1\underline{a}bbccbT_1 \to\!\!\to 1\_aT_2$
26	$1\underline{c}ababcT_1 \to\!\!\to 3\_T_2 \overset{\underline{b}}{\Rightarrow} 1\underline{a}bbccacT_1 \to\!\!\to 1\_aT_2$
27	$1\underline{c}abcT_1 \to\!\!\to 3\_T_2 \overset{\underline{b}}{\Rightarrow} \left.\begin{matrix}2\underline{a}ccac\\2\underline{a}ccbc\end{matrix}\right\}T_1 \to\!\!\to 1\_aT_2$
28	$1\underline{c}cT_1 \to\!\!\to 3\_T_2 \overset{\underline{b}}{\Rightarrow} 1\underline{a}bcT_1 \to\!\!\to 1\_aT_2$
29	$2\underline{T_1} \to\!\!\to 3\_T_2 \overset{\underline{b}}{\Rightarrow} 1\underline{a}T_1 \to\!\!\to 1\_aT_2$
30	$3\underline{T_1} \to\!\!\to 3\_T_2 \overset{\underline{b}}{\Rightarrow} 2\underline{a}T_1 \to\!\!\to 1\_aT_2$
31	$2\underline{T_1} \to\!\!\to 3\_baT_2 \overset{\underline{c}}{\Rightarrow} 2\underline{b}T_1 \to\!\!\to 3bbb\underline{T_2}$
32	$1\underline{c}aT_1 \to\!\!\to 3\_babT_2 \overset{\underline{c}}{\Rightarrow} \left.\begin{matrix}3\underline{c}ca\\3\underline{c}cb\end{matrix}\right\}T_1 \to\!\!\to 3\_babaT_2$
33	$1\underline{c}aaT_1 \to\!\!\to 3\_babT_2 \overset{\underline{c}}{\Rightarrow} \left.\begin{matrix}2\underline{b}baa\\2\underline{b}bab\end{matrix}\right\}T_1 \to\!\!\to 3\_babaT_2$
34	$1\underline{c}ababT_1 \to\!\!\to 3\_babT_2 \overset{\underline{c}}{\Rightarrow} 2\underline{b}bbccbT_1 \to\!\!\to 3\_babaT_2$

35 $\mathtt{1\underline{c}ababcT_1} \rightarrow\rightarrow \mathtt{3\_babT_2} \overset{\mathtt{c}}{\Rightarrow} \mathtt{2\underline{b}bbccacT_1} \rightarrow\rightarrow \mathtt{3\_babaT_2}$

36 $\mathtt{1\underline{c}abcT_1} \rightarrow\rightarrow \mathtt{3\_babT_2} \overset{\mathtt{c}}{\Rightarrow} \left.\begin{matrix}\mathtt{3\underline{c}ccac}\\ \mathtt{3\underline{c}ccbc}\end{matrix}\right\} \mathtt{T_1} \rightarrow\rightarrow \mathtt{3\_babaT_2}$

37 $\mathtt{1\underline{c}cT_1} \rightarrow\rightarrow \mathtt{3\_babT_2} \overset{\mathtt{c}}{\Rightarrow} \mathtt{2\underline{b}bcT_1} \rightarrow\rightarrow \mathtt{3\_babaT_2}$

38 $\mathtt{2\underline{T_1}} \rightarrow\rightarrow \mathtt{3\_babT_2} \overset{\mathtt{c}}{\Rightarrow} \mathtt{2\underline{b}T_1} \rightarrow\rightarrow \mathtt{3\_babaT_2}$

39 $\mathtt{3\underline{T_1}} \rightarrow\rightarrow \mathtt{3\_babT_2} \overset{\mathtt{c}}{\Rightarrow} \mathtt{3\underline{c}T_1} \rightarrow\rightarrow \mathtt{3\_babaT_2}$

40 $\mathtt{1\underline{T_1}} \rightarrow\rightarrow \mathtt{1T_{2\_}} \left\{\begin{matrix}\overset{\mathtt{a}}{\Rightarrow} \left.\begin{matrix}\mathtt{3T_1\underline{a}}\\ \mathtt{3T_1\underline{b}}\end{matrix}\right\} \rightarrow\rightarrow \mathtt{2T_2b\_}\\ \overset{\mathtt{c}}{\Rightarrow} \mathtt{2T_1\underline{c}} \rightarrow\rightarrow \mathtt{1T_2b\_}\end{matrix}\right.$

41 $\mathtt{1\underline{T_1}} \rightarrow\rightarrow \mathtt{2T_{2\_}} \overset{\mathtt{a}}{\Rightarrow} \left.\begin{matrix}\mathtt{3T_1\underline{a}}\\ \mathtt{3T_1\underline{b}}\end{matrix}\right\} \rightarrow\rightarrow \mathtt{3T_2b\_}$

42 $\mathtt{3\underline{T_1}} \rightarrow\rightarrow \mathtt{2T_{2\_}} \overset{\mathtt{b}}{\Rightarrow} \mathtt{1T_1\underline{b}} \rightarrow\rightarrow \mathtt{2T_2c\_}$

43 $\mathtt{1\underline{T_1}} \rightarrow\rightarrow \mathtt{3T_{2\_}} \overset{\mathtt{c}}{\Rightarrow} \mathtt{2T_1\underline{c}} \rightarrow\rightarrow \mathtt{3T_2c\_}$

44 $\mathtt{3T_1bb\underline{b}} \rightarrow\rightarrow \mathtt{3T_{2\_}} \overset{\mathtt{c}}{\Rightarrow} \left.\begin{matrix}\mathtt{2T_1aab\underline{c}}\\ \mathtt{2T_1abb\underline{c}}\end{matrix}\right\} \rightarrow\rightarrow \mathtt{3T_2c\_}$

45 $\mathtt{1\underline{T_1}} \rightarrow\rightarrow \mathtt{2T_2b\_} \overset{\mathtt{c}}{\Rightarrow} \mathtt{2T_1\underline{c}} \rightarrow\rightarrow \mathtt{3\underline{T_2}bc}$

46 $\mathtt{1\underline{T_1}} \rightarrow\rightarrow \mathtt{2T_2c\_} \overset{\mathtt{c}}{\Rightarrow} \mathtt{2T_1\underline{c}} \rightarrow\rightarrow \mathtt{1T_2bb\_}$

47 $\mathtt{3\underline{T_1}} \rightarrow\rightarrow \mathtt{3T_2c\_} \overset{\mathtt{b}}{\Rightarrow} \mathtt{1T_1\underline{b}} \rightarrow\rightarrow \mathtt{2T_2bb\_}$

48 $\mathtt{1\underline{T_1}} \rightarrow\rightarrow \mathtt{2T_2bb\_} \overset{\mathtt{c}}{\Rightarrow} \mathtt{2T_1\underline{c}} \rightarrow\rightarrow \mathtt{1\underline{T_2}abc}$

49 $\mathtt{3\underline{T_1}} \rightarrow\rightarrow \mathtt{3T_2bb\_} \overset{\mathtt{b}}{\Rightarrow} \mathtt{1T_1\underline{b}} \rightarrow\rightarrow \mathtt{1\underline{T_2}aba}$

50 $\mathtt{3\underline{T_1}} \rightarrow\rightarrow \mathtt{3T_2cb\_} \overset{\mathtt{b}}{\Rightarrow} \mathtt{1T_1\underline{b}} \rightarrow\rightarrow \mathtt{3T_2bbb\_}$

51 $\mathtt{3T_1bb\underline{b}} \rightarrow\rightarrow \mathtt{3T_2cb\_} \overset{\mathtt{a}}{\Rightarrow} \left.\begin{matrix}\mathtt{3T_1aab\underline{a}}\\ \mathtt{3T_1abb\underline{a}}\\ \mathtt{3T_1aab\underline{b}}\\ \mathtt{3T_1abb\underline{b}}\end{matrix}\right\} \rightarrow\rightarrow \mathtt{3T_2bbb\_}$

52 $\mathtt{3\underline{T_1}} \rightarrow\rightarrow \mathtt{3T_2bbb\_} \overset{\mathtt{b}}{\Rightarrow} \mathtt{1T_1\underline{b}} \rightarrow\rightarrow \mathtt{3\underline{T_2}baba}$

53 $\mathtt{1\underline{T_1}} \rightarrow\rightarrow \mathtt{3T_2bbbbb\_} \overset{\mathtt{a}}{\Rightarrow} \left.\begin{matrix}\mathtt{3T_1\underline{a}}\\ \mathtt{3T_1\underline{b}}\end{matrix}\right\} \rightarrow\rightarrow \mathtt{3\underline{T_2}bababa}$

54 $\mathtt{3\underline{T_1}} \rightarrow\rightarrow \mathtt{3T_2bbbbb\_} \overset{\mathtt{b}}{\Rightarrow} \mathtt{1T_1\underline{b}} \rightarrow\rightarrow \mathtt{3\underline{T_2}bababa}$

55 $\mathtt{3T_1bb\underline{b}} \rightarrow\rightarrow \mathtt{3T_2bbbbb\_} \overset{\mathtt{a}}{\Rightarrow} \left.\begin{matrix}\mathtt{3T_1aab\underline{a}}\\ \mathtt{3T_1abb\underline{a}}\\ \mathtt{3T_1aab\underline{b}}\\ \mathtt{3T_1abb\underline{b}}\end{matrix}\right\} \rightarrow\rightarrow \mathtt{3\underline{T_2}bababa}$

Theorem 4.1 demonstrates the importance of derivations of IRR's using chains of IGR's substituted into each other. Connected with this is the relation 'can be followed by' which restricts the possible sequences of substitutions of IGR's. This is given in Table 2 and requires a match on the LHS and on the RHS in which the machine state and the symbol strings must match, as well as the direction for adding α, and the first IGR must be of extendable type i.e. it must generate IRR's of type LRL or RLR. In Table 2 the numbers refer to IGR's in Table 1. The letters if present refer to the part of the IGR associated with that letter as the symbol above $\Rightarrow$ i.e. the symbol called α, otherwise the whole IGR is referred to. The following number refers to the sub-part of the IGR with that numbered origin in the RHS of the IGR. On the RHS of Table 1 (to the right of $\rightarrow$) all parts and sub-parts of an IGR referenced are included. Every IGR on the left of $\rightarrow$ can be followed by any IGR on the right of $\rightarrow$ in the same row.

Table 2: The relation ‘can be followed by’

1 →	22, 23, 24, 25, 26, 27, 28, 32, 33, 34, 35, 36, 37
3b1, 3b2, 4, 5, 7, 8 →	22
2b1, 2b2, 6b1, 6b2, 9, 13c1, 13c2, 14, 15, 17, 18, 33c1, 33c2, 34, 35, 37, 38 } →	29, 31, 38
12c1, 12c2, 16c1, 16c2, 36c1, 36c2, 39 →	30, 39
22 →	1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 13, 14, 15, 16, 18, 19
23b1, 23b2, 27b1, 27b2, 30 →	8, 18
24b1, 24b2, 25, 26, 28, 29 →	1
32c1, 32c2 →	29, 38
40a1, 40a2 →	42
41a1 →	49, 50, 52, 54
41a2 →	44, 49, 50, 51, 52, 54, 55
42 →	41, 46
47 →	41, 45, 48
50 →	43, 53
51a1, 51a2, 51a3 →	49, 52, 54
51a4 →	44, 49, 52, 54, 55

By examining these IGR’s in Table 1 and the compatibility relations in Table 2 the following facts become evident:

1. There are a relatively small number of distinct origins of the LHS’s of these IGR’s. Each of these together with the value of α gives rise to the same origin of the RHS of the IGR regardless of the RHS of the LHS of the IGR. For example $\mathtt{1\underline{c}aT_1}$ with $\alpha = \mathtt{b}$ is always associated with $\left.\begin{array}{l}\mathtt{2\underline{a}ca}\\ \mathtt{2\underline{a}cb}\end{array}\right\}\mathtt{T_1}$ in IGR’s 2 and 23.

2. IGR’s can be chained together by substitutions for the arbitrary strings $\mathtt{T_1}$ and $\mathtt{T_2}$.

3. In the chain of substitutions, there is a restriction on which IGR can follow another IGR; this results from the structure of the IGR’s themselves. This information is given in Table 2.

4. Other restrictions result from the way in which sequences of substitutions operate.

5. By carrying out `F` to the RHS of an IGR, it is sometimes possible to deduce that no previous IGR’s to a sequence of them can affect which IGR’s can follow the sequence.

6. If by carrying out `F` to any sequence of IGR’s to find which IGR’s can be next in the sequence, there always results IGR’s that have already been listed, then it would show that the set of IGR’s found is sufficient to generate all the IRR’s for the TM.

5 A modification of F to be applied to generating IGR's from IGR's

An obvious step to take is to attempt to directly derive Table 1 starting from all the IGR's needed to derive the IRR(2) from the TM table without generating all the IRR's. The method is based on the procedure I called $\mathtt{F}$ and will be modified as follows and denoted by $\mathtt{F}^*$ because instead of explicit sequences of symbols, arbitrary strings $\mathtt{T}_1$ and $\mathtt{T}_2$ are involved. In the backward search to get the new origins, if the pointer reaches adjacent to the arbitrary string $\mathtt{T}_1$, the two cases $\mathtt{T}_1 = \varepsilon$ and $\mathtt{T}_1 \neq \varepsilon$ need to be considered separately where ε is the empty string. In the first case the computation halts because this is what happens with $\mathtt{F}$, but in the second case, the backward seach it continues but the CS's with the pointer adjacent to $\mathtt{T}_1$ are recorded, until it is stopped by (i) no further backward TM steps being possible, (ii) the pointer reaching α or (iii) a stationary cycle is found. In the forward computaton of the new RHS, the presence of the arbitrary string $\mathtt{T}_2$ merely stops the computation where the pointer reaches the end of $\mathtt{T}_2$.

Another reason for a change to $\mathtt{F}$ is because for the simplest case with $r_1 = 1$ i.e. an IGR of the form $\cdots \Rightarrow \mathtt{t}\underline{\mathtt{y}}\mathtt{T}_1 \rightarrow \cdots$ to start with, $\mathtt{F}^*$ starts from the backward search beginning with $\mathtt{t}\alpha\underline{\mathtt{y}}\mathtt{T}_1$ from which at most one reverse TM step can be done giving say $\mathtt{t}'\underline{\alpha'}\mathtt{y}\mathtt{T}_1$, so the symbol $\mathtt{y}$ is not involved and can be put into $\mathtt{T}_1$, so the derived IGR has $\mathtt{t}'\underline{\alpha'}\mathtt{T}_1'$, and $r_1' = 1$ as is r_1 thus derived IGR's must always have $r_1 = 1$ which is obviously not going to generate all the results in Table 1. To solve this while maximising the generality of the results i.e. involving the contexts to the minimal extent possible, try involving the context symbol on the left only if $r_1 = 1$.

Therefore the general procedure I propose for a TM is to first generate the set IRR(2). These can be abbreviated to IGR's with length (1,1) by taking out the context pair and will be put into the set $\mathtt{S}$ but they should be kept in initially. Then the following procedure is repeated until hopefully it comes to an end with no more members to be processed. Take the next unmarked member of $\mathtt{S}$, mark it and apply $\mathtt{F}^*$ to it generating new IGR's (for both cases $\mathtt{T}_1 = \varepsilon$ and $\mathtt{T}_1 \neq \varepsilon$) that are added to $\mathtt{S}$ if either are different from all the existing members of $\mathtt{S}$.

On studying how this works, a problem was found arising when applying $\mathtt{F}^*$ to IGR 40.1 in Table 1, which itself arises with the context pair $(\mathtt{c},\mathtt{b})$ i.e. the IRR $\mathtt{3c\underline{d}} \to\!\!\to \mathtt{2bb\_}$. The origins are $\mathtt{3T_1\underline{a}}$ and $\mathtt{3T_1\underline{b}}$ which are written more succinctly as $\mathtt{3T_1\underline{d}}$ and the context symbol on the left is $\mathtt{c}$. So putting this back in starting the procedure $\mathtt{F}^*$ gives the CS $\mathtt{3T_1c\underline{d}}\alpha$ from which the backward search gives

$$\mathtt{3T_1c\underline{d}}\alpha \begin{cases} \overset{\alpha=\mathtt{b}}{\longleftarrow} \mathtt{1T_1cd\underline{b}} \\ \leftarrow \mathtt{3T_1\underline{c}d}\alpha \overset{\mathtt{d}=\mathtt{b},\mathtt{T_1}\neq\varepsilon}{\longleftarrow} \mathtt{1T_1c\underline{b}}\alpha \begin{cases} \overset{\alpha=\mathtt{a}}{\longleftarrow} \mathtt{3T_1cb\underline{d}} \\ \overset{\alpha=\mathtt{c}}{\longleftarrow} \mathtt{2T_1cb\underline{c}} \end{cases} \end{cases} .$$

On the right $\mathtt{F}^*$ gives $\begin{cases} \mathtt{2T_2bb\underline{a}} \to \mathtt{3T_2bbb\_} \\ \mathtt{2T_2bb\underline{b}} \to \mathtt{2T_2bbc\_} \\ \mathtt{2T_2bb\underline{c}} \to \mathtt{1\underline{T_2}abc} \end{cases}$ which can be shortened, by absorbing symbols that cannot be changed because the pointer does not reach them into $\mathtt{T_1}$ or $\mathtt{T_2}$, gives the following:

$$\begin{array}{l} \mathtt{3T_1c\underline{d}} \to\!\!\to \mathtt{2T_2\_} \overset{\alpha=\mathtt{a},\mathtt{T_1}\neq\varepsilon}{\Rightarrow} \mathtt{3T_1cb\underline{d}} \to\!\!\to \mathtt{3T_2b\_} \\ \mathtt{3T_1c\underline{d}} \to\!\!\to \mathtt{2T_2\_} \overset{\mathtt{T_1}=\varepsilon}{\Rightarrow} \mathtt{3\underline{c}d}\alpha \\ \mathtt{3\underline{T_1}} \to\!\!\to \mathtt{2T_2\_} \overset{\alpha=\mathtt{b},\mathtt{T_1}\neq\varepsilon}{\Rightarrow} \mathtt{1T_1\underline{b}} \to\!\!\to \mathtt{2T_2c\_} \\ \mathtt{3T_1c\underline{d}} \to\!\!\to \mathtt{2T_2bb\_} \overset{\alpha=\mathtt{c},\mathtt{T_1}\neq\varepsilon}{\Rightarrow} \mathtt{2T_1cb\underline{c}} \to\!\!\to \mathtt{1\underline{T_2}abc} \end{array} . \qquad (24)$$

The results (24).1, (24).3 and (24).4 are IGR's expressed more fully involving the condition $\mathtt{T_1} \neq \varepsilon$ and (24).2 just gives an RCS. The computer program output shows that the application of $\mathtt{F}^*$ to IGR 40.1 does not appear in the analysis of this TM because IGR 40.1 only appears once in the set of full IGR's (with contexts) having the context pair $(\mathtt{c},\mathtt{b})$. This implies that the above analysis only applies with $\mathtt{T_1} = \varepsilon$ invalidating the reasoning requiring the right hand halves of (24).1 and (24).4 to be IRR patterns involved in the analysis of TM1.

****

Temporarily disregarding property 1, and in the hope that Table 1 would have property 6, manual calculation was started beginning with IGR 22 because from Table 2, IGR 22 clearly plays an important role. By restricting at first consideration to IGR $\mathtt{1}$ followed by IGR 22 denoted by $\mathtt{1} \cdot 22$ fewer possibilities will result for the following IGR's. It was soon found that this is likely to get very unwealdy because of the large number of cases to be considered, Nevertheless is was instructive to try. The first case to be considered was what IGR's that can follow $\mathtt{1} \cdot 22$? The sequence of IGR's $\mathtt{1} \cdot 22$ is $\mathtt{1\underline{T_1}} \to\!\!\to \mathtt{1\_T_2} \overset{\mathtt{b}}{\Rightarrow} \mathtt{1\underline{c}T_1} \to\!\!\to \mathtt{3\_bT_2} \overset{\mathtt{b}}{\Rightarrow} \mathtt{1\underline{c}cT_1} \to\!\!\to \mathtt{1\_abT_2}$ obtained by substituting $\mathtt{cT_1}$ for $\mathtt{T_1}$ and $\mathtt{bT_2}$ for $\mathtt{T_2}$ in IGR 22. The result of this is a composite IGR. The IRR's that it generates are a subset of the IRR's generated by looking for which IGR's follow IGR 22 alone. By trying to apply $\mathtt{F}$ to this general form, results dependent on the arbitrary strings $\mathtt{T_1}$ and $\mathtt{T_2}$ will be produced. This starts by considering what CS's can lead to $\mathtt{1}\alpha\underline{\mathtt{c}}\mathtt{cT}_1$ for any symbol α. It is easy to see that

$$\mathtt{1}\alpha\underline{\mathtt{c}}\mathtt{cT}_1 \begin{cases} \overset{\alpha=\mathtt{b}}{\leftarrow} \mathtt{1}\underline{\mathtt{c}}\mathtt{ccT}_1 \\ \leftarrow \mathtt{2}\alpha\mathtt{c}\underline{\mathtt{c}}\mathtt{T}_1 \end{cases} . \tag{25}$$

in one TM step in either case. The first of these will lead to the IRRP $\mathtt{1}\underline{\mathtt{c}}\mathtt{ccT}_1 \to\to \mathtt{3\_babT}_2$ because $\mathtt{1}\underline{\mathtt{b}}\mathtt{abT}_2 \to \mathtt{3\_babT}_2$. The strings $\mathtt{ccT}_1$ and $\mathtt{abT}_2$ are not changed because the pointer does not enter them in the derivation of the IRRP, so the IGR used is $\mathtt{1}\underline{\mathtt{T}_1} \to\to \mathtt{1\_T}_2 \overset{\mathtt{b}}{\Rightarrow} \mathtt{1}\underline{\mathtt{c}}\mathtt{T}_1 \to\to \mathtt{3\_bT}_2$ i.e. IGR $\mathtt{1}$ in Table 1. The second result of (25) has reached condition 2 in the backward search if $\mathtt{T}_1$ is the empty string, which implies that there is no point in continuing the backward search further in that case.

If $\mathtt{T}_1$ is not the empty string, the general reasoning indicates that $\mathtt{T}_1$ needs to be specialised further by prepending the sequence of IGR's $\mathtt{1}\cdot\mathtt{22}$ with others, however the backward search can be logically continued giving

$$\mathtt{2}\alpha\mathtt{c}\underline{\mathtt{c}}\mathtt{T}_1 \leftarrow \mathtt{2}\alpha\underline{\mathtt{b}}\mathtt{cT}_1 \begin{cases} \overset{\alpha=\mathtt{b}}{\leftarrow} \mathtt{1}\underline{\mathtt{a}}\mathtt{bcT}_1 \\ \overset{\alpha=\mathtt{c}}{\leftarrow} \mathtt{2}\underline{\mathtt{b}}\mathtt{bcT}_1 \end{cases} \tag{26}$$

which is independent of $\mathtt{T}_1$ because the first of these reverse steps from $\mathtt{2}\alpha\mathtt{c}\underline{\mathtt{c}}\mathtt{T}_1$ cannot lead to any other result than the one indicated (because there is no TM step ending in $\mathtt{2\_}\beta$ no matter what the symbol β in $\mathtt{T}_1$ is). This shows that if $\mathtt{T}_1$ is not the empty string, the result will always be that condition 1 is reached, giving another IGR.

Returning to the general argument,

taking a further step back in the sequence of IGR's to be considered gives for example $\mathtt{22}\cdot\mathtt{1}\cdot\mathtt{22}$. This sequence gives

$$\mathtt{1}\underline{\mathtt{T}_1} \to\to \mathtt{3\_T}_2 \overset{\mathtt{22b}}{\Rightarrow} \mathtt{1}\underline{\mathtt{c}}\mathtt{T}_1 \to\to \mathtt{1\_aT}_2 \overset{\mathtt{bb}}{\Rightarrow} \mathtt{1}\underline{\mathtt{c}}\mathtt{ccT}_1 \to\to \mathtt{1\_abaT}_2 \tag{27}$$

the second part of which comes from $\mathtt{1}\cdot\mathtt{22}$ above. The symbols above the symbols $\Rightarrow$ respectively indicate IGR 22 with $\alpha = \mathtt{b}$ and as above $(\mathtt{1}\cdot\mathtt{22})$ with two steps of IGR's with $\alpha = \mathtt{b}$. Applying $\mathtt{F}$ to this starts by the backward search from $\mathtt{1}\alpha\underline{\mathtt{c}}\mathtt{ccT}_1$ giving (e.g. using (25))

$$\mathtt{1}\alpha\underline{\mathtt{c}}\mathtt{ccT}_1 \begin{cases} \leftarrow \mathtt{2}\alpha\mathtt{c}\underline{\mathtt{c}}\mathtt{cT}_1 \leftarrow \mathtt{2}\alpha\underline{\mathtt{b}}\mathtt{ccT}_1 \begin{cases} \overset{\mathtt{b}}{\leftarrow} \mathtt{1}\underline{\mathtt{a}}\mathtt{bccT}_1 \\ \overset{\mathtt{c}}{\leftarrow} \mathtt{2}\underline{\mathtt{b}}\mathtt{bccT}_1 \end{cases} . \\ \overset{\mathtt{b}}{\leftarrow} \mathtt{1}\underline{\mathtt{c}}\mathtt{cccT}_1 \end{cases} \tag{28}$$

Combining this with $\mathtt{1}\underline{\mathtt{b}}\mathtt{abaT}_2 \to \mathtt{3\_babaT}_2$ and $\mathtt{1}\underline{\mathtt{c}}\mathtt{abaT}_2 \to \mathtt{3babb}\underline{\mathtt{T}_2}$ and absorbing any unchanged symbols into $\mathtt{T}_1$ or $\mathtt{T}_2$ because the pointer has not reached them gives the results $\mathtt{1}$, $\mathtt{7}$ and

$$\mathtt{1}\underline{\mathtt{c}}\mathtt{cT}_1 \to\to \mathtt{1\_abaT}_2 \overset{\mathtt{c}}{\Rightarrow} \mathtt{2}\underline{\mathtt{b}}\mathtt{bcT}_1 \to\to \mathtt{3bbcb}\underline{\mathtt{T}_2}. \tag{29}$$

[As an aside comment, Actually `17` is a special case of `11` which is itself a special case of (29) formed by successively decreasing the length of the string in the RHS by 1. Because in these three cases, `17` uniquely has the pointer finishing at the α end of the string in RHS of the RHS, such a sequence as (29), `11` and `17` cannot be continued by specialising $\mathtt{T}_2$ and continuing the computation to the end in the RHS so any sequence of IGR's ending with `17`]

In 28 because of the absence of a branch of the backward search taking the pointer to the opposite end of the string from α, it implies that any special cases of $\mathtt{T}_1$ that would result from prior IGR's in the sequence could not affect the new origins of IGR's that could be next in the sequence, only the RHS's could vary. This is because the general form of the derivation of a new origin follows the pattern in (28) whatever substitutes for $\mathtt{T}_1$. Because `1` can also be preceded by `24b1`, `24b2`, `25`, `26`, `28`, `29`, these cases could now be considered in turn preceding $\mathtt{1} \cdot \mathtt{22}$.

These results are very complicated and the way forward seems unclear, because in the derivation of new IGR's by applying `F`, both the new origins then the new RHS's have to be found and there are a lot of different combinations of cases that can arise. Also the number of cases to be considered seems prohibitively large based on the relation 'can be followed by' in Table 2.

6 Further simplification and LIGR's

Returning to property 1 of Table 1, it appears that the left and right hand halves of the RHS of each IGR can be derived independently (it is only α that connects them), and the left hand sides (the origin of the LHS and the origin of the RHS) have a lot of repetition, many appearing multiple times, thus the presentation in Table 1 is far from optimal although the list of IGR's given there for generating any IRR from the IRR(2) does now seem to be complete and can can be derived systematically up to any given length of the IRR's.

What is hinted at above is that there could be an alternative algorithm to generate the IGR's directly from each other by applying `F` in these general cases for arbitrary strings $\mathtt{T}_1$ and $\mathtt{T}_2$. This became extremely complicated because of dealing with new origins and RHS's together. It is this that I want to arrive at by just considering at first the derivation of new origins for the IGR's, for which I introduce the new concept of LIGR (Left IGR) because the RHS's can always be filled in later just with forward computations.

The aim of this section will be to demonstrate this algorithm before formulating it precisely and hopefully show that if it does come to an end, then the LIGR's so obtained from a Turing Machine will be sufficient rules to allow the computation of all the IRR's to any level desired. Unfortunately in the present example it has as yet proved too difficult to complete this analysis.

6.1 LIGR's

An LIGR or left IGR is the origin of the LHS of an IGR, and the origin of the RHS of the same IGR, combined with the symbol α. For example IGR's `2`, `23` have the common LIGR

$$\mathtt{1}\alpha\underline{\mathtt{c}}\mathtt{aT_1} \overset{\alpha=\mathtt{b}}{\longleftarrow} \left.\begin{matrix}\mathtt{2\underline{a}ca}\\ \mathtt{2\underline{a}cb}\end{matrix}\right\} \mathtt{T_1}. \tag{30}$$

Similarly

$$\mathtt{1}\alpha\underline{\mathtt{c}}\mathtt{aT_1} \overset{\alpha=\mathtt{c}}{\longleftarrow} \left.\begin{matrix}\mathtt{3\underline{c}ca}\\ \mathtt{3\underline{c}cb}\end{matrix}\right\} \mathtt{T_1} \tag{31}$$

is common to IGR's `12` and `20`. These examples show that in common with IGR's, LIGR's can have parts (labelled by α) and sub-parts that will be labelled in lexicographical order of the strings with the most significant symbol (sorted first) being where the pointer is. Also the symbol α may be omitted for brevity because it is always at the opposite end of the string from $\mathtt{T_1}$, which in the context of LIGR's will be called just `T` because $\mathtt{T_2}$ is not involved. For example if (30) and (31) are treated as parts of the complete LIGR `X` then (30) is `X.b` and (31) is `X.c`. The length of an LIGR will be the length of the symbol string on its right, which is one more than the length of the symbol string on the left assuming α is ommited. This notation with the reverse facing arrow will be used because as usual the arrows ($\leftarrow$ or $\rightarrow$) indicate the direction of the computation of the TM as distinct from logical derivation indicated by $\Rightarrow$. Thus LIGR's are also reverse computation rules, but very special ones because they arise in the context of IGR's.

There are obvious advantages of treating IGR's in this way as can be seen in the drastically shortened list of results (12 LIGR's from Table 1 not counting parts and sub-parts separately). Moreover if an LIGR (on its LHS) matches the origin of an IRR, `F` applied to this IRR has as origins the result of the substitution for $\mathtt{T_1}$ in the RHS of the LIGR, and its RHS can be computed directly from the original RHS using alpha of the LIGR. Thus the RHS's can be filled in later and do not need to be recorded in the rule for generating new IRR's from existing ones.

6.2 Sequences of LIGR's and `F`

A sequence of LIGR's is a chain of LIGR's that can follow each other written with $\cdot$ between them so for example if

$$\mathtt{L_1} = \mathtt{1\underline{c}aT} \overset{\mathtt{b}}{\longleftarrow} \left.\begin{matrix}\mathtt{2\underline{a}ca}\\ \mathtt{2\underline{a}cb}\end{matrix}\right\} \mathtt{T}. \tag{32}$$

$$\mathtt{L_2} = \mathtt{2\underline{T}} \left\{\begin{matrix}\overset{\mathtt{b}}{\longleftarrow} \mathtt{1\underline{a}T}\\ \overset{\mathtt{c}}{\longleftarrow} \mathtt{2\underline{b}T}\end{matrix}\right. \tag{33}$$

$$\mathtt{L}_3 = \mathtt{1\underline{T}} \overset{\mathtt{b}}{\leftarrow} \mathtt{1\underline{c}T} \tag{34}$$

then

$$\begin{aligned} \mathtt{L}_{1.2} &= \mathtt{1\underline{c}aT} \overset{\mathtt{b}}{\leftarrow} \mathtt{2\underline{a}cbT} \\ \mathtt{L}_{2.1} &= \mathtt{2\underline{T}} \overset{\mathtt{b}}{\leftarrow} \mathtt{1\underline{a}T} \end{aligned} \tag{35}$$

and the chain $\mathtt{L}_{1.2} \cdot \mathtt{L}_{2.1} \cdot \mathtt{L}_3$ is the the result of the three substitutions of the LHS performed in that sequence giving

$$\mathtt{1\underline{c}aT} \leftarrow \mathtt{2\underline{a}cbT} \leftarrow \mathtt{1\underline{a}acbT} \leftarrow \mathtt{1\underline{c}aacbT} \tag{36}$$

so the combination is $\mathtt{1\underline{c}aT} \overset{\mathtt{bbb}}{\leftarrow} \mathtt{1\underline{c}aacbT}$.

Similarly to the way in which $\mathtt{F}$ was applied to sequences of IGR's combined together, this can obviously be done for sequences of LIGR's.

The result of $\mathtt{F}$ applied to a sequence of LIGR's of length 1 cannot give rise to any residual CS's because there is not enough "room" and can be affected by adding an extra LIGR to the beginning of the sequence. To show this suppose an LIGR or a sequence of LIGR's combined as above $\mathtt{X}_1$ with α on the left has a sub-part of the form

$$\mathtt{s}_1\underline{\mathtt{y}_1}\ldots\mathtt{y}_\mathtt{r}\mathtt{T} \overset{\alpha}{\leftarrow} \mathtt{s}_2\underline{\mathtt{z}_1}\ldots\mathtt{z}_{\mathtt{r}+1}\mathtt{T}. \tag{37}$$

Likewise let $\mathtt{X}_2$ be

$$\mathtt{s}'_1\underline{\mathtt{y}'_1}\ldots\mathtt{y}'_{\mathtt{r}'}\mathtt{T} \overset{\alpha'}{\leftarrow} \mathtt{s}'_2\underline{\mathtt{z}'_1}\ldots\mathtt{z}'_{\mathtt{r}'+1}\mathtt{T}. \tag{38}$$

Then for the sequence $\mathtt{X}_2 \cdot \mathtt{X}_1$ to be possible requires, $\mathtt{s}'_2 = \mathtt{s}_1$ and $\underline{\mathtt{y}_1}\ldots\mathtt{y}_\mathtt{r}$ is a substring of $\underline{\mathtt{z}'_1}\ldots\mathtt{z}'_{\mathtt{r}'+1}$ on the left, and α is on the left for $\mathtt{X}_2$ too. The result of $\mathtt{X}_2 \cdot \mathtt{X}_1$ is

$$\mathtt{s}'_1\underline{\mathtt{y}'_1}\ldots\mathtt{y}'_{\mathtt{r}'}\mathtt{T} \overset{\alpha'}{\leftarrow} \mathtt{s}'_2\underline{\mathtt{z}'_1}\ldots\mathtt{z}'_{\mathtt{r}'+1}\mathtt{T} = \mathtt{s}_1\underline{\mathtt{y}_1}\ldots\mathtt{y}_\mathtt{r}\mathtt{z}'_{\mathtt{r}+1}\ldots\mathtt{z}'_{\mathtt{r}'+1}\mathtt{T} \overset{\alpha}{\leftarrow} \mathtt{s}_2\underline{\mathtt{z}_1}\ldots\mathtt{z}_{\mathtt{r}+1}\mathtt{z}'_{\mathtt{r}+1}\ldots\mathtt{z}'_{\mathtt{r}'+1}\mathtt{T}. \tag{39}$$

Comparing (37) with (39) shows the effect on $\mathtt{X}_1$ of preceding it with $\mathtt{X}_2$ which is to add extra symbols next to $\mathtt{T}$ in its right hand member. Therefore the results of the backward search starting from the RHS of (37) are reproduced when started from the RHS of (39) and shortened to the shortest form provided the pointer ends up at α; these give rise to LIGR's. In addition there may be some extra LIGR's resulting from the pointer reaching the extra symbols which may be classified by the position of the rightmost symbol reached. Crucially, this happens only when $\mathtt{F}$ applied to $\mathtt{X}_1$ leads to cases in the backward search when the pointer ends up at the opposite end of the string from α i.e. condition 2 is reached because the above searches can then be truncated when they reach the opposite end from α. These were termed residual CS's (RCS's) because they are cases that do not lead directly to any more IGR's and LIGR's but indicate the possibility of them if the sequence of LIGR's to which $\mathtt{F}$ is applied increases in length as a result of a preceding LIGR appended to the sequence in question.

Finally, there may be results of this where the pointer ends up at the opposite end of the string from α i.e. the pointer goes right when the rightmost symbol is reached. These are the new RCS's in the result of $\mathtt{F}$ applied to (39) and will be designated as $\mathtt{F}_2(\mathtt{X}_2 \cdot \mathtt{X}_1)$. Therefore it makes sense to introduce $\mathtt{\Delta F}_1$ as the set of extra LIGR's from $\mathtt{F}$, as a result of adding an extra LIGR $\mathtt{Y}_1$ at the beginning of the sequence where $\mathtt{S} = \mathtt{X}_1 \cdot \mathtt{X}_2 \ldots \cdot \mathtt{X}_n$ is a sequence of LIGR's:

$$\mathtt{\Delta F}_1(\mathtt{Y}_1, \mathtt{S}) = \mathtt{F}_1(\mathtt{Y}_1 \cdot \mathtt{S}) \setminus \mathtt{F}_1(\mathtt{S}). \tag{40}$$

In this notation the result of the preceding paragraph can be written as

$$\mathtt{F}_2(\mathtt{S}) = \emptyset \Rightarrow \mathtt{\Delta F}_1(\mathtt{Y}_1, \mathtt{S}) = \emptyset. \tag{41}$$

The converse is not true because it could be that there are some reverse search paths that go beyond the symbols in $\mathtt{S}$ but none of them go back to α. In this case $\mathtt{F}_2(\mathtt{Y}_1 \cdot \mathtt{S}) \neq \emptyset$ or some of these reverse search paths just reach an end because at some point no reverse TM step is possible. Here the result of $\mathtt{F}$ applied to a sequence of LIGR's was split into two components $\mathtt{F} = (\mathtt{F}_1, \mathtt{F}_2)$. Also the result of $\mathtt{F}$ for a collection of LIGR's in a sequence is defined as the result of $\mathtt{F}$ for the combined LIGR, which actually only depends on the its RHS and results from applying the backward search algorithm to it. A consequence of this is that the arguments of $\mathtt{F}$ can be written in different ways e.g. the sequence of LIGR's can be replaced by the equivalent sequence of symbols in the RHS of the combined LIGR.

6.3 Evaluating (40) one extra symbol at a time

The following is a description of the above calculation taken one symbol at a time. It can be applied when several symbols are added in one step from a single LIGR as was the original intention, or when a sequence of LIGR's is added that each contribute just one symbol etc..

The result of (40) can obviously be obtained by adding each symbol one at a time and accumulating the results for each extra symbol added. Suppose the extra symbols added to the starting CS $\mathtt{s}_2\underline{\mathtt{z}_1} \cdots \mathtt{z}_{r+1}$ from $\mathtt{S}$ as a result of preceding it with $\mathtt{Y}_1$ are $\mathtt{z}'_{r+1} \ldots \mathtt{z}'_{r'+1}$ then one can write:

$$\begin{aligned} \mathtt{F}_1(\mathtt{Y}_1 \cdot \mathtt{S}) &= \mathtt{F}_1(\mathtt{s}_2\underline{\mathtt{z}_1} \ldots \mathtt{z}_{r+1}\mathtt{z}'_{r+1} \ldots \mathtt{z}'_{r'+1}) \\ &= \left[\bigcup_{i=r+1}^{i=r'+1} \mathtt{\Delta F}_1(\mathtt{z}'_i, \mathtt{s}_2\underline{\mathtt{z}_1} \ldots \mathtt{z}_{r+1}\mathtt{z}'_{r+1} \ldots \mathtt{z}'_{i-1})\right] \cup \mathtt{F}_1(\mathtt{s}_2\underline{\mathtt{z}_1} \ldots \mathtt{z}_{r+1}) \end{aligned} \tag{42}$$

where the first term of the union is (40) and the second term is $\mathtt{F}_1(\mathtt{S})$. Each step corresponding to one term in the multiple union uses the RCS's from the previous step. These RCS's with the single extra symbol are the starting points of the continuing backward search which would of course stop if at some point there were no more RCS's. In more detail, in order to calculate

$$\mathtt{\Delta F_1(z'_i, s_2\underline{z_1} \dots z_{r+1}z'_{r+1} \dots z'_{i-1})} \tag{43}$$

which is by definition

$$\mathtt{F_1(s_2\underline{z_1} \dots z_{r+1}z'_{r+1} \dots z'_i) \setminus F_1(s_2\underline{z_1} \dots z_{r+1}z'_{r+1} \dots z'_{i-1})}, \tag{44}$$

in the backward search the pointer must reach $\mathtt{z'_i}$ before ending up at the right or left end (otherwise duplicate results are obtained that are to be eliminated), therefore the backward search can start from

$$\mathtt{F_2(s_2\underline{z_1} \dots z_{r+1}z'_{r+1} \dots z'_{i-1})\underline{z'_i}} \tag{45}$$

where the last symbol is concatenated to the residual results of $\mathtt{F_2}$. If the pointer reaches α the result is a new LIGR otherwise it gives an RCS which is in

$$\mathtt{F_2(s_2\underline{z_1} \dots z_{r+1}z'_{r+1} \dots z'_i)}. \tag{46}$$

Putting $i = r + 1$ initially, then this shows that the backward search starts from $\mathtt{F_2(s_2\underline{z_1} \dots z_{r+1})\underline{z'_{r+1}}}$ i.e. the set of RCS's from the initial backward search for $\mathtt{S}$ each appended with the first extra symbol $\mathtt{z'_{r+1}}$ on the right. If the pointer reaches α as the backward search continues, this gives a new LIGR otherwise it gives an RCS in $\mathtt{F_2(s_2\underline{z_1} \dots z_{r+1}z'_{r+1})}$ if the pointer reaches the opposite end of the string of symbols, or comes to a point where the backward search can go no further or end in an infinite stationary loop. Then for $i = r + 2$, the backward search starts from this appended with $\mathtt{z'_{r+2}}$ on the right. Again the backward search continues either the pointer reaches α giving a new LIGR, or away from it giving an RCS in $\mathtt{F_2(s_2\underline{z_1} \dots z_{r+1}z'_{r+1}z'_{r+2})}$ etc.. This continues until all the new symbols have been added and all possible backward search paths are followed at each stage. If at any stage there are no RCS's in $\mathtt{F_2}$ it terminates. All the new LIGR's are accumulated and any final RCS's are noted. Naturally, there is an equivalent version of this if α is on the right.

It follows from this that the calculation of the combined results of $\mathtt{\Delta F}$ applied to any sequence of LIGR's that can precede $\mathtt{S}$ is also given by going back one symbol at a time and finding $\mathtt{\Delta F}$ in each case without any restriction on the new symbol added other than up to that point $\mathtt{F_2} \neq \emptyset$. This might be a better general algorithm than the one following, but in the example for TM 23 I use a mixture of both strategies.

7 The procedure for finding the LIGR's for a TM

This algorithm is extremely complicated and is very hard to describe so the reader is not expected to understand this immediately. For this reason I attempt to do so here in this section and then the algorithm is applied to TM (23) in detail in Section 7.1 when it should become clearer.

Algorithm 7.1. *The algorithm* $\mathtt{F}$ *applied repeatedly can generate all the IRR's starting with members of IRR(2) therefore the first step is to find* $\mathtt{L}$ *the set of LIGR's corresponding to the formation of the members of IRR(3) from those of IRR(2) by applying* $\mathtt{F}$*. Doing this starts with putting the arbitrary symbol* α *at one end of the pair of symbols in the origin of the member of IRR(2) and the backward search starts with the pointer at the middle symbol. A single reverse TM step gives a member of IRR(3) if the pointer moves to* α *and if it goes the other way condition 2 occurs giving an RCS, so only one symbol is involved therefore the reduced form (the LIGR) that results (if the pointer goes to* α*) must have length 1. The RHS of each of these LIGR's of length 1 already is an RCS because the pointer is already adjacent to* $\mathtt{T}$*.*

Next the LIGR's involved in forming the IRR$(n+1)$ *from the IRR*(n) *need to be found for all* $n \geq 3$*. This can be done by searching for all LIGR's that can follow sequences of LIGR's that have already been found. This involves applying* $\mathtt{F}$ *to an arbitrary sequence* $\mathtt{S}$ *of such of LIGR's substitiuted into each other as in Section 6.2. This has to be done until closure i.e. until no more LIGR's can be found if this is repeated one more time. For this, as shown in Section 6.2,* $\Delta\mathtt{F}_1$ *and* $\mathtt{F}_2$ *need to be found only if the previous* $\mathtt{F}_2 \neq \emptyset$

i.e. only if $\mathtt{F}$ *applied to* $\mathtt{S}$ *gives* $\mathtt{F}_2(\mathtt{S}) \neq \emptyset$*, carry out* $\mathtt{F}$ *to obtain* $\Delta\mathtt{F}_1(\mathtt{X}\cdot\mathtt{S})$ *and* $\mathtt{F}_2(\mathtt{X}\cdot\mathtt{S})$ *for each LIGR* $\mathtt{X}$ *that can precede* $\mathtt{S}$ *where the requirement for precedence is given in* (39)*. This calculation can start from the previous* $\mathtt{F}_2$ *with the substitution made for* $\mathtt{T}$ *indicated by* $\mathtt{X}$*. This condition will be met for each of the initial set of LIGR's in* $\mathtt{L}$ *because as shown above their RHS's each have the form of an RCS. Any new LIGR's from* $\Delta\mathtt{F}_1(\mathtt{X}\cdot\mathtt{S})$ *are added to* $\mathtt{L}$*. For any members of* $\mathtt{F}_2(\mathtt{X}\cdot\mathtt{S})$ *apply this algorithm recursively i.e. with* $\mathtt{X}\cdot\mathtt{S}$ *taking the place of* $\mathtt{S}$ *above etc.. It is expected that this algorithm will terminate because the matching criterion for new LIGR's that can be prepended to the sequence gets increasingly stringent as the length of the strings increases. If this happens repeat this algorithm the "grand search" with the new enlarged set* $\mathtt{L}$ *of LIGR's. This should in fact be just to add to the previous results instead of repeating them because the previous LIGR's are still in* $\mathtt{L}$*. Repeat this "grand search" until the set of extra LIGR's added to* $\mathtt{L}$ *in one cycle is empty.*

Definition 7.2. *A finite set of LIGR's* $\mathtt{Z}$ *is closed under* $\mathtt{F}$ *if for any sequence of members of* $\mathtt{Z}$ *that can be substituted into one another in sequence as in* (36)*, the backward search* $\mathtt{F}$ *applied to the result of this generates results that are each a member of the set* $\mathtt{Z}$*.*

Theorem 7.3. *If there is a finite set* $\mathtt{Z}$ *of LIGR's for a Turing Machine that is closed under* $\mathtt{F}$ *as described above and includes all the LIGR's involved in obtaining the set IRR(3) from the set IRR(2) then every IRR for that TM can be obtained from a member of IRR(2) by a sequence of applications of LIGR's each in* $\mathtt{Z}$ *as described in Section 6.2.*

It is also obvious that no LIGR's could be removed from this set `Z` and `Z` still have this property because members of `Z` are only put there when they are required.

Proof. Consider deriving the members of IRR(4) from IRR(3). This can be done by applying `F` to each member of IRR(3) and accumulating all the results. Applying `F` gives results that come from members of `Z` because any LIGR following an LIGR in `Z` is also in `Z` by the closure property. Likewise deriving members of IRR(5) involve applying `F` to members of IRR(4) that themselves have been derived using a pair of LIGR's. Again all such derivations of LIGR's that can follow this sequence are in `Z` by the closure property etc.. □

This all assumes that `Z` is finite. The case if the closure algorithm does not terminate leading to an infinite set `Z` closed under `F` might be interesting.

The remainder of this section contains the application of Algorithm 7.1 to the example (23). The results are not all presented in the order in which they were derived i.e. there are forward references to LIGR's that have not yet been derived. This is because `L` is increasing in size as the algorithm proceeds and to avoid duplication, the results are presented assuming the current `L` is the final one and in the order determined by the grand search i.e."depth first" (if not the results will be added to). This all assumes that the final `L` is finite.

It will be useful to use the symbol `d` in the remainder of this paper to mean either `a` or `b`. This arose as an abbreviation useful for TM (23). Each instance of `d` will be independent of any other one in a CS, so that all combinations are possible. If the combinations that are possible are a subset of these, this is more complicated and alternatives will be given in braces, but this does not usually happen. This will allow many cases to be considered simultaneously so speeding up the computations. When doing reverse computations, all possible reverse steps from any of the combinations will be included.

That there is a finite number of LIGR's has been strongly suggested in the present case by the computer results that established Table 1 using any value of the maximum length of the strings involved (n) between 10 and 16 and show the same result. The computations rapidly increase in number and time taken as n increases.

Checking the arguments requires the derivation route from the initial LIGR's in `L` to all the final ones which will be given so that the results can all be checked.

The following are the 7 LIGR's that arise from the derivation of the IRR(3) from the IRR(2):

Why not start with the IGR(2) that generate IRR(2)?

$$\begin{array}{ll}
1 & 2\underline{\mathtt{T}} \overset{\mathtt{b}}{\leftarrow} 1\underline{\mathtt{a}}\mathtt{T} \\
2 & 3\underline{\mathtt{T}} \overset{\mathtt{b}}{\leftarrow} 1\mathtt{T}\underline{\mathtt{b}} \\
3 & 1\underline{\mathtt{T}} \overset{\mathtt{b}}{\leftarrow} 1\underline{\mathtt{c}}\mathtt{T} \\
4 & 2\underline{\mathtt{T}} \overset{\mathtt{c}}{\leftarrow} 2\underline{\mathtt{b}}\mathtt{T} \\
5 & 1\underline{\mathtt{T}} \overset{\mathtt{c}}{\leftarrow} 2\mathtt{T}\underline{\mathtt{c}} \\
6 & 1\underline{\mathtt{T}} \overset{\mathtt{a}}{\leftarrow} 3\mathtt{T}\underline{\mathtt{a}} \\
7 & 1\underline{\mathtt{T}} \overset{\mathtt{a}}{\leftarrow} 3\mathtt{T}\underline{\mathtt{b}}
\end{array} \cdot \tag{47}$$

The relation "can possibly be preceded by" on the LIGR's is also being discovered continually as the LIGR's are being discovered. The criterion is that the RHS of the preceding LIGR must match the LHS of the original LIGR in state, string of symbols and direction. The relation "can possibly be preceded by" initially among the 7 LIGR's is given by

$$\begin{array}{ll}
1 & 4 \\
2 & 6,7 \\
3 & 1,3 \\
4 & 4 \\
5 & 2 \\
6 & 2 \\
7 & 2
\end{array} . \tag{48}$$

Note that these LIGR's all appear in the final set but the numbering is slightly changed in the final set due to the use of `d` meaning `a` or `b` and other LIGR's including those of length 1 being found. While carrying out the main argument in Section 7.1 the following results emerged and are collected here for convenience because they may need to be referred to anywhere throughout the main argument. They arose as induction hypotheses to deal with endlessly repeating situations that arose during the application of Algorithm 7.1 to TM 23 or other complex situations arising there and so simplify the presentation of this.

Lemma 7.4. *The reversed TM cannot cross the symbol* `a` *going left, and*

Lemma 7.5. *The reversed TM cannot cross the pairs of symbols* `cx` *going right where* `x` *is any symbol*

Proof. There is no reverse TM step of the form $\mathtt{xa\_} \leftarrow \mathtt{CS}$ therefore the pointer cannot get left of the `a` which is maintained. Also if the pointer were to reach just left of the symbol `c` a further reverse step to the right is only possible if it is to the CS $2\underline{\mathtt{c}}$ (using $2\underline{\mathtt{c}} \rightarrow \mathtt{1\_c}$ in reverse). The next reverse TM step must be to the left if at all because there are no TM steps of the form $\mathtt{CS} \rightarrow \mathtt{2\_x}$ where `x` is any symbol. Thus the symbols `cx` are maintained and the pointer has not crossed them. □

Lemma 7.6. *The backward search from any CS of the form* $\mathtt{1Tadb\underline{a}aa}\alpha$ *cannot lead to any new LIGR's or RCS's provided the string* $\mathtt{T}$ *contains the symbol* $\mathtt{a}$.

Proof. This is by continuing the backward search from there. This gives the following tree

$$\mathtt{1Tadb\underline{a}aa}\alpha \leftarrow \begin{cases} \mathtt{1Tad\underline{c}a}^3\alpha \leftarrow \begin{cases} \mathtt{3Tadc\underline{d}a}^2\alpha \leftarrow \mathtt{3Tad\underline{c}da}^2\alpha \leftarrow \begin{cases} \mathtt{2Ta\underline{a}cda}^2\alpha \\ \mathtt{1Tadc\underline{b}a}^2\alpha * \end{cases} \\ \mathtt{1Ta\underline{c}ca}^3\alpha \end{cases} \\ \mathtt{3Tadba\underline{d}a}\alpha \end{cases} \tag{49}$$

$$\mathtt{1Tadc\underline{b}a}^2\alpha \leftarrow \mathtt{3Tadcb\underline{d}a}\alpha \leftarrow \mathtt{2Tadc\underline{a}da}\alpha \leftarrow \mathtt{2Tad\underline{b}ada}\alpha \leftarrow \mathtt{1Ta\underline{a}bada}\alpha \tag{50}$$

where the computation stopped whenever either no reverse TM step is possible, or when by Lemmas (7.4) or (7.5) the pointer cannot go beyond the string as a result of continued backward searching. Because all branches of the tree do eventually lead to a halt, no LIGR's or RCS's can result from further backward searching. □

Lemma 7.7. *Backward searching starting from any CS of the form* $\mathtt{1Td\underline{c}abada}\alpha$ *leads to exactly the following set of CS's regardless of the arbitary string* $\mathtt{T}$ *in addition to possible CS's with the pointer at the left depending on* $\mathtt{T}$*:*

$$\begin{gathered} \mathtt{1Tdca}^2\mathtt{dbd\underline{b}} \\ \mathtt{3Tdca}^3\mathtt{db\underline{d}} \\ \mathtt{2Tdca}^3\mathtt{db\underline{c}} \\ \mathtt{1Tdcdbdbd\underline{b}} \\ \mathtt{3Tdcdcbdb\underline{d}} \\ \mathtt{2Tdcdcbdb\underline{c}} \\ \mathtt{3Tdcdbadb\underline{d}} \\ \mathtt{2Tdcdbadb\underline{c}} \end{gathered} \tag{51}$$

These are related to the set of LIGR's in 95.20-23*.*

Proof. The backward search stops if either (1) the pointer can be shown not to get to the right because of `cx` on the right of the pointer or (2) no further backward TM steps are possible or (3) the end of the known symbols on the string is reached or (4) a stationary cycle is reached. The numbers after * indicate continuations.

$$
\mathtt{1Td\underline{c}abada}\alpha \leftarrow \begin{cases} \mathtt{1T\underline{c}cabada}\alpha \\ \mathtt{3Tdc\underline{d}bada}\alpha \leftarrow \begin{cases} \mathtt{3Td\underline{c}dbada}\alpha \leftarrow \begin{cases} \mathtt{2T\underline{a}cdbada}\alpha \\ \mathtt{1Tdc\underline{b}bada}\alpha \end{cases} \\ \mathtt{1Tdcd\underline{b}ada}\alpha \leftarrow \begin{cases} \mathtt{1Tdc\underline{c}bada}\alpha \\ \mathtt{3Tdcdb\underline{d}da}\alpha * 1 \end{cases} \end{cases} \end{cases}
$$

$$
* 1 \leftarrow \begin{cases} \mathtt{2Tdcd\underline{a}dda}\alpha \leftarrow \mathtt{1Tdc\underline{a}adda}\alpha \leftarrow \mathtt{3Tdca\underline{d}dda}\alpha \leftarrow \mathtt{1Tdcad\underline{b}da}\alpha * 2 \\ \mathtt{1Tdcdbd\underline{b}a}\alpha * 5 \end{cases}
$$

$$
* 2 \leftarrow \begin{cases} \mathtt{1Tdca\underline{c}bda}\alpha \\ \mathtt{3Tdcadb\underline{d}a}\alpha \leftarrow \mathtt{2Tdcad\underline{a}da}\alpha \leftarrow \mathtt{1Tdca\underline{a}ada}\alpha \leftarrow \mathtt{3Tdcaa\underline{d}da}\alpha * 3 \end{cases}
$$

$$
* 3 \leftarrow \mathtt{1Tdcaad\underline{b}a}\alpha \leftarrow \begin{cases} \mathtt{1Tdcaa\underline{c}ba}\alpha \\ \mathtt{3Tdcaadb\underline{d}}\alpha \leftarrow \begin{cases} \mathtt{2dca^2d\underline{a}d}\alpha \leftarrow \mathtt{1Tdcaa\underline{a}ad}\alpha * 4 \\ \mathtt{1Tdca^2dbd\underline{b}} \end{cases} \end{cases}
$$

$$
* 4 \leftarrow \mathtt{3Tdca^3\underline{d}d}\alpha \leftarrow \mathtt{1Tdca^3d\underline{b}}\alpha \leftarrow \begin{cases} \mathtt{1Tdca^3\underline{c}b}\alpha \\ \mathtt{3Tdca^3db\underline{d}} \\ \mathtt{2Tdca^3db\underline{c}} \end{cases}
$$

$$
* 5 \leftarrow \begin{cases} \mathtt{1Tdcdb\underline{c}ba}\alpha \leftarrow \mathtt{1Tdcd\underline{c}cba}\alpha \\ \mathtt{3Tdcdbdb\underline{d}}\alpha \leftarrow \begin{cases} \mathtt{2Tdcdbd\underline{a}d}\alpha \leftarrow \mathtt{1Tdcdb\underline{a}ad}\alpha \leftarrow \begin{cases} \mathtt{1Tdcd\underline{c}aad}\alpha * 6 \\ \mathtt{3Tdcdba\underline{d}d}\alpha * 8 \end{cases} \\ \mathtt{1Tdcdbdbd\underline{b}} \end{cases} \end{cases}
$$

$$
* 6 \leftarrow \begin{cases} \mathtt{1Tdc\underline{c}caad}\alpha \\ \mathtt{3Tdcdc\underline{d}ad}\alpha \leftarrow \mathtt{3Tdcd\underline{c}dad}\alpha \leftarrow \begin{cases} \mathtt{2Tdc\underline{a}cdad}\alpha \\ \mathtt{1Tdcdc\underline{b}ad}\alpha \leftarrow \mathtt{3Tdcdcb\underline{d}d}\alpha * 7 \end{cases} \end{cases}
$$

$$
* 7 \leftarrow \begin{cases} \mathtt{2Tdcdc\underline{a}dd}\alpha \leftarrow \mathtt{2Tdcd\underline{b}add}\alpha \leftarrow \mathtt{1Tdc\underline{a}badd}\alpha \\ \mathtt{1Tdcdcbd\underline{b}}\alpha \leftarrow \begin{cases} \mathtt{1Tdcdcb\underline{c}b}\alpha \leftarrow \mathtt{1Tdcdc\underline{c}cb}\alpha \\ \mathtt{3Tdcdcbdb\underline{d}} \\ \mathtt{2Tdcdcbdb\underline{c}} \end{cases} \end{cases}
$$

$$
* 8 \leftarrow \mathtt{1Tdcdbad\underline{b}}\alpha \leftarrow \begin{cases} \mathtt{1Tdcdba\underline{c}b}\alpha \\ \mathtt{3Tdcdbadb\underline{d}} \\ \mathtt{2Tdcdbadb\underline{c}} \end{cases}
$$

(52)

□

Lemma 7.8. *If in a row of Table 3, some LIGR's are produced then in another row with the RHS of "its affect" differing only by the symbol next to the string* $\mathtt{T}$*, the same set of LIGR's is produced.*

For the same pair of rows of the table, the two sets of RCS's are related to each other because truncating these computations by the the last step must give the same reults. esuch that a single forward computation step steps if possible.

Proof. Suppose a backward search gives

$$\mathtt{ST}\beta\underline{\mathtt{T_1}}\alpha \leftarrow \mathtt{A} \tag{53}$$

in the Table 3 where the pointer is at the right had end of $\mathtt{T}_1$ where $\mathtt{A}$ is a set of RCS's and LIGR's. Then this will be based on

$$\mathtt{ST}\underline{\mathtt{T_1}}\alpha \leftarrow \mathtt{S_1T}\underline{\mathtt{T_2}}\alpha \tag{54}$$

(an RCS) that may be also in the same table where the pointer is at the left hand end of $\mathtt{T}_2$, and α and β are arbitrary symbols (with α and $\mathtt{T}$ having their usual roles) and $\mathtt{T}_1$ and $\mathtt{T}_2$ (as is $\mathtt{T}$) are arbitrary strings of symbols and $\mathtt{S}$ and $\mathtt{S}_1$ are arbitrary states. This is because truncating the string to the right of $\mathtt{T}$ by one symbol on the left will convert any LIGR's arising only because of that last symbol to RCS's. This will be done several times if necessary to get the required line of the grand search. Therefore (53) can be written as

$$\mathtt{ST}\beta\underline{\mathtt{T_1}}\alpha \leftarrow \mathtt{S_1T}\beta\underline{\mathtt{T_2}}\alpha \leftarrow \mathtt{A}. \tag{55}$$

If the pointer does not reach β in this derivation, it follows from this that β can be replaced by any other symbol say γ

$$\mathtt{ST}\gamma\underline{\mathtt{T_1}}\alpha \leftarrow \mathtt{S_1T}\gamma\underline{\mathtt{T_2}}\alpha \leftarrow \left\{\mathtt{S_2T}\underline{\delta}\mathtt{T_2}\alpha, \mathtt{A}\right\}. \tag{56}$$

where the first member of the set on the right (an RCS) is there if and only if in the TM table $\mathtt{S_2}\underline{\delta} \to \mathtt{S_1}\gamma_{\text{-}}$, and γ and δ are also arbitrary symbols. If (53) does not lead to any RCS's then the derivation cannot have the pointer reaching β then the derivation is followed as in the proof of (55) except that β is replaced by γ and the pointer never reaches γ leading to the same LIGR's after the unused symbol γ has been removed and no RCS's. If β is reached in (55) then follow the reverse steps that lead to the pointer reaching β giving some RCS's that could be different from those in (55).

As long as β is not reached by the pointer, the symbols to the right of β are independent of β. Therefore if the backward search from (53) is completed, the corresponding results with a different value of β are obtained by (1) assessing whether or not the single step to β is possible from the start or from any point where the pointer reaches one space to the right of β and if so including the RCS obtained, and (2) taking the results that don't take the pointer to β and replacing β by the new symbol. This leaves the LIGR's unchanged after the symbol in place of β that plays no part in the calculations is removed. □

Lemma 7.9. *Any RCS of the form* $\mathtt{2T\underline{b}b^{n+1}add}\alpha$ *does not lead to any LIGR's for* $\mathtt{n} \geq \mathtt{0}$.

Proof. For convenience let the string $\mathtt{b^{n+1}add}\alpha$ be deonoted by $\mathtt{S}$ because it remains the same throughout the proof and note that the leftmost symbol of $\mathtt{S}$ is $\mathtt{b}$ if $\mathtt{n} \geq \mathtt{0}$. Add an arbitrary symbol β on the left according to the procedure described in section 6.3 and continue the backward search from there gives $\mathtt{2T}\beta\mathtt{\underline{b}S} \leftarrow \begin{cases}\mathtt{1T\underline{a}bS}\\ \mathtt{2T\underline{b}bS}\end{cases}$. The second case is as above with $\mathtt{n}$ increased by 1. Repeat this for the first case giving $\mathtt{1T}\beta\mathtt{\underline{a}bS} \leftarrow \mathtt{1T\underline{c}abS}$. Repeat this argument again gives

$$\mathtt{1T}\beta\mathtt{\underline{c}abS} \leftarrow \begin{cases}\mathtt{1T\underline{c}cabS}\ * \\ \\ \mathtt{3T}\beta\mathtt{c\underline{d}bS} \leftarrow \begin{cases} \mathtt{3T}\beta\mathtt{\underline{c}dbS} \leftarrow \begin{cases}\mathtt{1T}\beta\mathtt{c\underline{b}bS}\\ \mathtt{2T\underline{a}cdbS}\ * \\ \mathtt{3T\underline{c}cdbS}\ * \end{cases} \\ \mathtt{1T}\beta\mathtt{cd\underline{b}S} \leftarrow \mathtt{1T}\beta\mathtt{c\underline{c}bS} \end{cases} \\ \end{cases} \tag{57}$$

In this search tree, $*$ indicates that by Lemma 7.5 the pointer can never reach α i.e. no new LIGR's can result from further additions of symbols. Because this search tree is complete it follows that the backward search from $\mathtt{2T\underline{b}b^{n+1}add}\alpha$ cannot lead to an LIGR unless the backward search from $\mathtt{2T\underline{b}b^{n+2}add}\alpha$ also leads to an LIGR. This gives an infinite regress showing that no RCS of this form can lead to an LIGR for $\mathtt{n} \geq \mathtt{0}$. □

7.1 The main argument

Because of the fact that LIGR's result from backward searches where the pointer starts and ends up at α, any such result can be usefully classified by the parameter j_1 in (22), the maximum number of reverse TM steps away from α during the computation. The resulting LIGR will have length (on the right) equal to $j_1 + 2$ because the symbol where the computation starts and α have to be included. Obviously if a substring involved in such a computation is the same between two strings (obtained by truncation from opposite α), the results obtained such that the pointer never leaves that substring will also be the same, therefore the set of all LIGR's and their lengths obtained from a given string allows the set of LIGR's and their lengths to be obtained from any substring of the string by truncating it from the end opposite α. This is just the subset given by $j_1 \leq$ the length of the substring -2. The values of j_1 are given by the program [4] to facilitate obtaining the LIGR's.

Due to the problem with presenting this and the forward references mentioned above, in the following the numbers of LIGR's refer to the LIGR's listed in (95) which is the most up to date list of LIGR's obtained by Algorithm 7.1 applied to TM (23).

Rather than repeating the phrase "Applying $\mathtt{F}$ to the sequence of LIGR's $\mathtt{X}$ gives ..." on many occasions it will be shortened to $\mathtt{X} \overset{\mathtt{F}}{\Rightarrow} \ldots$.

7.1.1 Sequences ending with LIGR $\mathtt{1}$

Applying $\mathtt{F}$ to $\mathtt{1}$ gives just $\mathtt{1}\alpha\underline{\mathtt{a}}\mathtt{T} \xleftarrow{\mathtt{b}} \mathtt{1}\underline{\mathtt{c}}\mathtt{aT}$ i.e. LIGR $\mathtt{3}$ i.e. $\mathtt{1} \overset{\mathtt{F}}{\Rightarrow} \mathtt{3}$. Clearly applying $\mathtt{F}$ to an LIGR of length 1 as is done here could possibly lead to more results if $\mathtt{T}$ is specialised by giving a symbol at one end of the string (here the left end) therefore preceding LIGR's must be considered. LIGR 1 can be preceded by LIGR $\mathtt{4}$ giving $\mathtt{4}\cdot\mathtt{1}$ having the combined effect $\mathtt{3}\underline{\mathtt{T}} \xleftarrow{\mathtt{b}} \mathtt{2}\underline{\mathtt{a}}\mathtt{T} \xleftarrow{\mathtt{b}} \mathtt{1}\underline{\mathtt{a}}\mathtt{aT}$ i.e. $\mathtt{3}\underline{\mathtt{T}} \xleftarrow{\mathtt{bb}} \mathtt{1}\underline{\mathtt{a}}\mathtt{aT}$ and $\mathtt{4}\cdot\mathtt{1} \overset{\mathtt{F}}{\Rightarrow} \mathtt{3}$ because $\mathtt{F}$ gives the calculation

$$\mathtt{1}\alpha\underline{\mathtt{a}}\mathtt{aT} \begin{cases} \xleftarrow{\mathtt{b}} \mathtt{1}\underline{\mathtt{c}}\mathtt{aaT} \\ \leftarrow \begin{cases} \mathtt{3}\alpha\mathtt{a}\underline{\mathtt{a}}\mathtt{T} \\ \mathtt{3}\alpha\mathtt{a}\underline{\mathtt{b}}\mathtt{T} \end{cases} \end{cases} \text{ i.e. } \mathtt{1}\alpha\underline{\mathtt{a}}\mathtt{aT} \begin{cases} \xleftarrow{\mathtt{b}} \mathtt{1}\underline{\mathtt{c}}\mathtt{aaT} \\ \leftarrow \mathtt{3}\alpha\mathtt{a}\underline{\mathtt{d}}\mathtt{T} \end{cases} \tag{58}$$

The first part of this is LIGR $\mathtt{3}$, and the second part is a pair of RCS's. By specialising this by giving the first symbol(s) of $\mathtt{T}$, the first result will not generate any new LIGR's after reducing the result to its shortest form, the result will merely be replicated, but for the second part it is possible that the reverse computation could take the pointer back to α and so generate more new LIGR's so the search has to continue back, so we have thus far

$$\begin{array}{r} \ldots\mathtt{1} \overset{\mathtt{F}}{\Rightarrow} \{\mathtt{3}\} \\ \ldots\cdot\mathtt{4}\cdot\mathtt{1} \overset{\mathtt{F}}{\Rightarrow} \{\mathtt{3}\} \end{array}. \tag{59}$$

The second member of this is related to its first member because any results of $\mathtt{F}$ from the first part must be included in the results of the second part as happens here but the RCS's are not the primary result of $\mathtt{F}$ and are not included in (59).2. There are RCS's so any LIGR's that can precede $\mathtt{4}\cdot\mathtt{1}$ must be considered and $\mathtt{F}$ must be applied to all these. The LIGR's that can precede $\mathtt{4}$ are just $\mathtt{8}$,$\mathtt{10}$ and $\mathtt{21}$. The sequence $\mathtt{8}\cdot\mathtt{4}\cdot\mathtt{1}$ gives $\mathtt{3}\underline{\mathtt{T}} \xleftarrow{\mathtt{c}} \mathtt{3}\underline{\mathtt{c}}\mathtt{T} \leftarrow \mathtt{1}\underline{\mathtt{a}}\mathtt{acT}$. Applying $\mathtt{F}$ to this starts from (from (58) with the substitiution given by LIGR 8) $\mathtt{1}\alpha\underline{\mathtt{a}}\mathtt{acT} \leftarrow \mathtt{3}\alpha\mathtt{a}\underline{\mathtt{d}}\mathtt{cT}$ in addition to $\mathtt{3}$ as above and $\Delta\mathtt{F}_1$ is just the result of this backward search from $\mathtt{3}\alpha\mathtt{a}\underline{\mathtt{d}}\mathtt{cT}$ and because the computation cannot go back from there $\Delta\mathtt{F}_1(\mathtt{8},\mathtt{4}\cdot\mathtt{1}) = \emptyset$ and $\mathtt{F}_2(\mathtt{8}\cdot\mathtt{4}\cdot\mathtt{1}) = \emptyset$. Therefore there are no results of $\mathtt{F}$ and it is now it is clear that no preceding LIGR's specialising this $\mathtt{T}$ can give any results of $\mathtt{F}$, so the search for new results of $\mathtt{F}$ stops in this branch of the grand search tree.

The sequences $\mathtt{10}\cdot\mathtt{4}\cdot\mathtt{1}$ have the effect $\mathtt{1}\underline{\mathtt{c}}\mathtt{aT} \leftarrow \mathtt{1}\underline{\mathtt{a}}\mathtt{accdT}$ and applying $\mathtt{F}$ gives $\mathtt{1}\alpha\underline{\mathtt{a}}\mathtt{accdT} \leftarrow \mathtt{3}\alpha\mathtt{a}\underline{\mathtt{d}}\mathtt{ccdT}$ from which there are no further backward steps so there are no new LIGR's or RCS's and these branches of the grand search end i.e. $\Delta\mathtt{F}_1(\mathtt{10}\cdot\mathtt{4}\cdot\mathtt{1}) = \mathtt{F}_2(\mathtt{10}\cdot\mathtt{4}\cdot\mathtt{1}) = \emptyset$. $\mathtt{21}\cdot\mathtt{4}\cdot\mathtt{1}$ has the effect $\mathtt{1}\underline{\mathtt{c}}\mathtt{abcT} \leftarrow \mathtt{1}\underline{\mathtt{a}}\mathtt{ac}^3\mathtt{dcT}$ and $\mathtt{F}$ applied to this by (58) gives $\mathtt{1}\alpha\underline{\mathtt{a}}\mathtt{ac}^3\mathtt{dcT} \leftarrow \begin{cases} \mathtt{3}\alpha\mathtt{a}\underline{\mathtt{a}}\mathtt{c}^3\mathtt{dcT} \\ \mathtt{3}\alpha\mathtt{a}\underline{\mathtt{b}}\mathtt{c}^3\mathtt{dcT} \end{cases}$ which by Lemma 7.5 cannot lead to an LIGR. This completes the analysis for all sequences that can precede $\mathtt{4}\cdot\mathtt{1}$ therefore the next sequence of LIGR's to be considered is $\mathtt{5}\cdot\mathtt{1}$ i.e. $\mathtt{2\underline{T}} \leftarrow \mathtt{2\underline{b}T} \leftarrow \mathtt{1\underline{a}bT}$. The computation of F starts from $\mathtt{1}\alpha\mathtt{\underline{a}bT}$ and gives no result other than LIGR 3, so $\Delta\mathtt{F}_1(5,1) = \mathtt{F}_2(5\cdot 1) = \emptyset$. Next $\mathtt{9}\cdot\mathtt{1}$ must be considered which is $\mathtt{1\underline{c}aT} \leftarrow \mathtt{2\underline{a}cdT} \leftarrow \mathtt{1\underline{a}acdT}$.

$$\mathtt{9}\cdot\mathtt{1} \overset{\mathtt{F}}{\Rightarrow} \mathtt{1}\alpha\mathtt{\underline{a}acdT} \leftarrow \begin{cases}\mathtt{1\underline{c}aacdT}\\ \mathtt{3}\alpha\mathtt{a\underline{d}cdT}\end{cases}. \tag{60}$$

which is also a special case of (58). The first of these is just LIGR 3 and the second cannot continue, so there are no new LIGR's from preceding 1 by 9 i.e. $\Delta\mathtt{F}_1(9,1) = \mathtt{F}_2(9\cdot 1) = \emptyset$. It is not too difficult to show that similar results hold for all the other LIGR's that could precede LIGR 1 and all the results starting with F applied to $\mathtt{5}\cdot\mathtt{1}$ can be summarised as

$$\Delta\mathtt{F}_1(\{5,9,12,14,20\},1) = \mathtt{F}_2(\{5,9,12,14,20\}\cdot 1) = \emptyset. \tag{61}$$

This exhausts all search trees in the grand search starting from LIGR 1.

7.1.2 Sequences ending with LIGR 2

Next consider applying F to sequences ending with 2 which is $\mathtt{3\underline{T}} \overset{\mathtt{b}}{\leftarrow} \mathtt{1T\underline{b}}$. F applied to this gives

$$\mathtt{1T\underline{b}}\alpha \begin{cases} \overset{\mathtt{a}}{\leftarrow} \mathtt{3Tb\underline{d}}\\ \overset{\mathtt{c}}{\leftarrow} \mathtt{2Tb\underline{c}}\end{cases} \tag{62}$$

which are LIGR's 6 and 7 so

$$\mathtt{2} \overset{\mathtt{F}}{\Rightarrow} \{6,7\}. \tag{63}$$

The last symbol of T in (62) could affect this result so LIGR's preceding 2 must be considered which are just 7,16,18,22,24 and26. The sequence $\mathtt{7}\cdot\mathtt{2}$ is $\mathtt{1\underline{T}} \overset{\mathtt{a}}{\leftarrow} \mathtt{3T\underline{d}} \leftarrow \mathtt{1Td\underline{b}}$ and applying F gives the same set i.e. LIGR's $6,7$ and no RCS's i.e. $\Delta\mathtt{F}_1(7,2) = \mathtt{F}_2(7\cdot 2) = \emptyset$. This is because the pointer cannot move left in the reverse computation regardless of any other specialisations of T resulting from preceding LIGR's. Consider $\mathtt{7}\cdot\mathtt{2}$ which is $\mathtt{1\underline{T}} \overset{\mathtt{a}}{\leftarrow} \mathtt{3T\underline{d}} \leftarrow \mathtt{1Td\underline{b}}$. F gives $\mathtt{1Td\underline{b}}\alpha \leftarrow \mathtt{1T\underline{c}b}\alpha$ in addition to 6 and 7 and so can be potentially specialised further i.e. $\Delta\mathtt{F}_1(7,2) = \emptyset$ and $\mathtt{F}_2(7\cdot 2) = \mathtt{1T\underline{c}b}\alpha$. LIGR 7 can only be preceded by 2 and 15 so the next sequence to be considered in the grand search is $\mathtt{2}\cdot\mathtt{7}\cdot\mathtt{2}$ which is $\mathtt{3\underline{T}} \overset{\mathtt{b}}{\leftarrow} \mathtt{1T\underline{b}} \leftarrow \mathtt{1Tbd\underline{b}}$. F applied to this gives

$$\mathtt{1Tbd\underline{b}}\alpha \leftarrow \mathtt{1Tb\underline{c}b}\alpha \leftarrow \mathtt{1T\underline{c}cb}\alpha \tag{64}$$

i.e. $\Delta\mathtt{F}_1(2,7\cdot 2) = \emptyset$ and $\mathtt{F}_2(2\cdot 7\cdot 2) = \mathtt{1T\underline{c}cb}\alpha$. This RCS by Lemma (7.5) cannot lead to any new LIGR's because the pointer can never reach α by the reverse TM computation however many preceding LIGR's are added to the sequence. Therefore there is no point in continuing grand search along this branch. This is a short cut that was not anticipated in the general algorithm

7.1. Next consider $\mathtt{15} \cdot \mathtt{7} \cdot \mathtt{2}$ having the effect $\mathtt{3Tb}^5\underline{\mathtt{a}} \leftarrow \mathtt{1Tc}\begin{Bmatrix}\mathtt{db}\\ \mathtt{aa}\end{Bmatrix}\mathtt{dbdbd}\underline{\mathtt{b}}$. Applying $\mathtt{F}$ to this gives a result of the same form as in (64) therefore the same conclusion follows and the grand search continues from $\mathtt{16} \cdot \mathtt{2}$ which has the effect $\mathtt{3Tb}^5\underline{\mathtt{a}} \leftarrow \mathtt{1Tca}^3\mathtt{dbd}\underline{\mathtt{b}}$. Applying $\mathtt{F}$ to this gives some results of the form (64) not leading to any new results as above or with the symbol $\mathtt{a}$ in the rightmost but one position. This is of the form $\mathtt{1Tba}\underline{\mathtt{b}}$ having no preceding TM step to the left in $\mathtt{F}$ hence $\mathtt{F}$ applied to this gives $\Delta\mathtt{F}_1 = \mathtt{F}_2 = \emptyset$.

The sequence $\mathtt{18}{\cdot}\mathtt{2}$ has the effect $\mathtt{3Tb}^5\underline{\mathtt{a}} \leftarrow \mathtt{3Tcd}\begin{Bmatrix}\mathtt{ba}\\ \mathtt{cb}\end{Bmatrix}\mathtt{db}\underline{\mathtt{d}} \leftarrow \mathtt{1Tcd}\begin{Bmatrix}\mathtt{ba}\\ \mathtt{cb}\end{Bmatrix}\mathtt{dbd}\underline{\mathtt{b}}$. Applying $\mathtt{F}$ to this gives
$\mathtt{1Tcd}\begin{Bmatrix}\mathtt{ba}\\ \mathtt{cb}\end{Bmatrix}\mathtt{dbd}\underline{\mathtt{b}}\alpha \overset{\mathtt{b}}{\leftarrow} \mathtt{1Tcd}\begin{Bmatrix}\mathtt{ba}\\ \mathtt{cb}\end{Bmatrix}\mathtt{db}\underline{\mathtt{c}}\mathtt{b}\alpha \leftarrow \mathtt{1Tcd}\begin{Bmatrix}\mathtt{ba}\\ \mathtt{cb}\end{Bmatrix}\mathtt{d}\underline{\mathtt{c}}\mathtt{cb}\alpha$. There is no point going any further with the algorithm $\mathtt{F}$ because from this CS, by Lemma (7.5) it is not possible for the pointer to get to α regardless of any preceding LIGR's i.e. no new LIGR's can result from this, another unanticipated short cut to Algorithm 7.1. Similar results all hold for $\{22, 24, 26\} \cdot \mathtt{2}$ which lead to applying the backward search from a CS of the form $\mathtt{1Tdbd}\underline{\mathtt{b}}\alpha$.

7.1.3 Sequences ending with LIGR 3

Consider sequences of LIGR's ending with $\mathtt{3}$ which is $\mathtt{1}\underline{\mathtt{T}} \overset{\mathtt{b}}{\leftarrow} \mathtt{1}\underline{\mathtt{c}}\mathtt{T}$. Applying $\mathtt{F}$ starts from $\mathtt{1}\alpha\underline{\mathtt{c}}\mathtt{T} \overset{\mathtt{b}}{\leftarrow} \mathtt{1}\underline{\mathtt{c}}\mathtt{cT}$ showing that $\mathtt{3} \overset{\mathtt{F}}{\Rightarrow} \{\mathtt{3}\}$. $\mathtt{3}$ can be preceded by $\mathtt{1}$, $\mathtt{3}$, $\mathtt{11}$ and $\mathtt{13}$. The sequence $\mathtt{1} \cdot \mathtt{3}$ is $\mathtt{2}\underline{\mathtt{T}} \leftarrow \mathtt{1}\underline{\mathtt{a}}\mathtt{T} \leftarrow \mathtt{1}\underline{\mathtt{c}}\mathtt{aT}$. Applying $\mathtt{F}$ starts from $\mathtt{1}\alpha\underline{\mathtt{c}}\mathtt{aT}\begin{cases}\overset{\mathtt{b}}{\leftarrow} \mathtt{1}\underline{\mathtt{c}}\mathtt{caT}\\ \leftarrow \mathtt{3}\alpha\mathtt{c}\underline{\mathtt{d}}\mathtt{T}\end{cases}$ showing that two new RCS's are the only extra results of preceding $\mathtt{3}$ with $\mathtt{1}$. The sequence $\mathtt{4} \cdot \mathtt{1} \cdot \mathtt{3}$ is $\mathtt{3}\underline{\mathtt{T}} \leftarrow \mathtt{2}\underline{\mathtt{a}}\mathtt{T} \leftarrow \mathtt{1}\underline{\mathtt{c}}\mathtt{aaT}$. Applying $\mathtt{F}$ gives

$$\mathtt{1}\alpha\underline{\mathtt{c}}\mathtt{aaT} \leftarrow \mathtt{3}\alpha\mathtt{c}\underline{\mathtt{d}}\mathtt{aT} \leftarrow \ldots \begin{cases}\leftarrow \mathtt{3}\alpha\mathtt{cb}\underline{\mathtt{d}}\mathtt{T}\\ \overset{\mathtt{b}}{\leftarrow} \mathtt{2}\underline{\mathtt{a}}\mathtt{cdaT}\\ \overset{\mathtt{c}}{\leftarrow} \mathtt{3}\underline{\mathtt{c}}\mathtt{cdaT}\end{cases}. \tag{65}$$

where the pointer does not reach the $\mathtt{a}$ adjacent to $\mathtt{T}$ during this computation of the last two parts therefore this shortens to

$$\mathtt{1}\underline{\mathtt{c}}\mathtt{aT}\begin{cases}\overset{\mathtt{b}}{\leftarrow} \mathtt{2}\underline{\mathtt{a}}\mathtt{cdT}\\ \overset{\mathtt{c}}{\leftarrow} \mathtt{3}\underline{\mathtt{c}}\mathtt{cdT}\end{cases} \tag{66}$$

which are new LIGR's and will be numbered $\mathtt{9}$ and $\mathtt{10}$ respectively, and the RCS's which require further backward searching. The sequence $\mathtt{8} \cdot \mathtt{4} \cdot \mathtt{1} \cdot \mathtt{3}$ is $\mathtt{3}\underline{\mathtt{T}} \leftarrow \mathtt{3}\underline{\mathtt{c}}\mathtt{T} \leftarrow \mathtt{1}\underline{\mathtt{c}}\mathtt{aacT}$. Applying $\mathtt{F}$ gives

$$\mathtt{1}\alpha\underline{\mathtt{c}}\mathtt{aacT} \leftarrow \mathtt{3}\alpha\mathtt{cb}\underline{\mathtt{d}}\mathtt{cT}\begin{cases}\overset{\mathtt{b}}{\leftarrow} \mathtt{1}\underline{\mathtt{a}}\mathtt{badcT}\\ \overset{\mathtt{c}}{\leftarrow} \mathtt{2}\underline{\mathtt{b}}\mathtt{badcT}\end{cases} \tag{67}$$

after a few steps. These when abbreviated are

$$\texttt{1\underline{c}aaT}\begin{cases}\xleftarrow{\texttt{b}}\texttt{1\underline{a}badT}\\ \xleftarrow{\texttt{c}}\texttt{2\underline{b}badT}\end{cases}\tag{68}$$

which will be numbered LIGR's 11 and 12 respectively. Also there are now no RCS's, so this branch of the grand search ends.

The sequence $\texttt{10}\cdot\texttt{4}\cdot\texttt{1}\cdot\texttt{3}$ has the effect $\texttt{1\underline{c}aT}\leftarrow\texttt{1\underline{c}aaccdT}$ and F gives, using (65),

$$\texttt{1}\alpha\texttt{\underline{c}aaccdT}\leftarrow\texttt{3}\alpha\texttt{cb\underline{d}ccdT}\begin{cases}\xleftarrow{\texttt{b}}\texttt{1\underline{a}badccdT}\\ \xleftarrow{\texttt{c}}\texttt{2\underline{b}badccdT}\end{cases}\tag{69}$$

produces only 11 and 12 again and no new RCS's.

The analysis for $\texttt{21}\cdot\texttt{4}\cdot\texttt{1}\cdot\texttt{3}$ is similar with the same result. Next consider $\texttt{5}\cdot\texttt{1}\cdot\texttt{3}$ with the effect $\texttt{2\underline{T}}\leftarrow\texttt{1\underline{c}abT}$. Then applying F gives

$$\texttt{1}\alpha\texttt{\underline{c}abT}\leftarrow\texttt{3}\alpha\texttt{c\underline{d}bT}\leftarrow\begin{cases}\texttt{3}\alpha\texttt{\underline{c}dbT}\begin{cases}\xleftarrow{\texttt{b}}\texttt{2\underline{a}cdbT}\\ \xleftarrow{\texttt{c}}\texttt{3\underline{c}cdbT}\end{cases}\\ \texttt{1}\alpha\texttt{cd\underline{b}T}\end{cases}\tag{70}$$

apart from CS's for which there is no preceding CS. This gives two results which reduce to LIGR's 9 and 10 and an RCS. $\texttt{4}\cdot\texttt{5}\cdot\texttt{1}\cdot\texttt{3}$ is $\texttt{3\underline{T}}\leftarrow\texttt{1\underline{c}abaT}$ and F gives

$$\texttt{1}\alpha\texttt{\underline{c}abaT}\leftarrow\texttt{1}\alpha\texttt{cd\underline{b}aT}\leftarrow\texttt{3}\alpha\texttt{cdb\underline{d}T}\tag{71}$$

from (70). This RCS requires going back in the grand search.

The next sequence $\texttt{7}\cdot\texttt{4}\cdot\texttt{5}\cdot\texttt{1}\cdot\texttt{3}$ is $\texttt{3\underline{T}}\leftarrow\texttt{1\underline{c}abacT}$. F gives

$$\texttt{1}\alpha\texttt{\underline{c}abacT}\leftarrow\texttt{3}\alpha\texttt{cdb\underline{d}cT}\xleftarrow{\texttt{5}}\texttt{2}\alpha\texttt{cadb\underline{c}T}.\tag{72}$$

By Lemma 7.4 because of the a on the left of the pointer, no new LIGR's can result if this branch of the grand search is continued. For $\texttt{10}\cdot\texttt{4}\cdot\texttt{5}\cdot\texttt{1}\cdot\texttt{3}$ which has the effect $\texttt{1\underline{c}aT}\leftarrow\texttt{1\underline{c}abaccdT}$, F can start (by (72)) from $\texttt{2}\alpha\texttt{cadb\underline{c}cdT}$ and again by Lemma 7.4 no LIGR's can result from this branch. The same holds for $\texttt{21}\cdot\texttt{4}\cdot\texttt{5}\cdot\texttt{1}\cdot\texttt{3}$. Consider $\texttt{5}\cdot\texttt{5}\cdot\texttt{1}\cdot\texttt{3}$, F can start from $\texttt{1}\alpha\texttt{\underline{c}abbT}\leftarrow\texttt{1}\alpha\texttt{cd\underline{b}bT}\leftarrow\texttt{1}\alpha\texttt{c\underline{c}bbT}$ from which there are no RCS's or LIGR's. Applying F to $\texttt{9}\cdot\texttt{5}\cdot\texttt{1}\cdot\texttt{3}$ can start from $\texttt{1}\alpha\texttt{\underline{c}abacaT}\leftarrow\texttt{1}\alpha\texttt{cd\underline{b}acaT}\leftarrow\begin{cases}\texttt{1\underline{c}cabacdT}\\ \texttt{2\underline{a}cdbacdT}\\ \texttt{3\underline{c}cdbacdT}\end{cases}$ where other branches terminate due to Lemmas 7.4 and 7.5 giving LIGR's 3, 9, 10. $\texttt{12}\cdot\texttt{5}\cdot\texttt{1}\cdot\texttt{3}$ is $\texttt{1\underline{c}aaT}\leftarrow\texttt{2\underline{b}badT}\leftarrow\texttt{1\underline{c}ab}^{3}\texttt{cadT}$

F gives $\texttt{1}\alpha\texttt{\underline{c}abcaaT}\leftarrow\texttt{1}\alpha\texttt{cd\underline{b}caaT}\leftarrow\begin{cases}\texttt{2\underline{a}ccdcaaT}\\ \texttt{3\underline{c}ccdcaaT}\\ \texttt{2}\alpha\texttt{cca\underline{c}aaT}\end{cases}$ where the last result cannot give any LIGR's. These can be shortened to the new LIGR's 20 and

21 respectively. $12 \cdot 5 \cdot 1 \cdot 3$ is $\mathtt{1\underline{c}aaT} \leftarrow \mathtt{2\underline{b}badT} \leftarrow \mathtt{1\underline{c}abbbadT}$. F gives $\mathtt{1}\alpha\mathtt{\underline{c}abbbabT} \leftarrow \begin{cases}\mathtt{2\underline{a}cdb^3adT}\\ \mathtt{3\underline{c}cdb^3adT}\end{cases}$ shortening to LIGR's 9 and 10.

$14 \cdot 5 \cdot 1 \cdot 3$ is $\mathtt{1\underline{c}cT} \leftarrow \mathtt{2\underline{b}bcT} \leftarrow \mathtt{1\underline{c}ab^3cT}$. F gives $\mathtt{1}\alpha\mathtt{\underline{c}ab^3cT} \leftarrow \mathtt{1}\alpha\mathtt{cd\underline{b}bbcT} \leftarrow \mathtt{1}\alpha\mathtt{c\underline{c}b^3cT}$ no new results. $9 \cdot 1 \cdot 3$ is $\mathtt{1\underline{c}aT} \leftarrow \mathtt{2\underline{a}cdT} \leftarrow \mathtt{1\underline{c}aacdT}$ F gives $\mathtt{1}\alpha\mathtt{\underline{c}aacdT} \leftarrow \mathtt{3}\alpha\mathtt{c\underline{d}acdT} \leftarrow \begin{cases}\mathtt{2\underline{a}cdacdT}\\ \mathtt{3\underline{c}cdacdT}\\ \mathtt{1\underline{a}badcdT}\\ \mathtt{2\underline{b}badcdT}\end{cases}$. These shorten to the LIGR's $9 - 12$ already found and no RCS's.

$12 \cdot 1 \cdot 3$ is $\mathtt{1\underline{c}aaT} \leftarrow \mathtt{2\underline{b}badT} \leftarrow \mathtt{1\underline{c}abbadT}$ F gives $\mathtt{1}\alpha\mathtt{\underline{c}abbaaT} \leftarrow \mathtt{3}\alpha\mathtt{c\underline{d}bbaaT} \leftarrow \begin{cases}\mathtt{2\underline{a}cdbbaaT}\\ \mathtt{3\underline{c}cdbbaaT}\end{cases}$ which shorten to LIGR's $9, 10$ with no RCS's. $14 \cdot 1 \cdot 3$ is $\mathtt{1\underline{c}cT} \leftarrow \mathtt{2\underline{b}bcT} \leftarrow \mathtt{1\underline{c}abbcT}$. F gives results that shorten to LIGR's $9, 10$ again.

The sequence $3 \cdot 3$ has the effect $\mathtt{1\underline{T}} \overset{\mathtt{b}}{\leftarrow} \mathtt{1\underline{c}T} \leftarrow \mathtt{1\underline{c}cT}$. F gives $\mathtt{1}\alpha\mathtt{\underline{c}cT} \leftarrow \mathtt{2}\alpha\mathtt{c\underline{c}T}$ an RCS so continue. $1 \cdot 3 \cdot 3$ has the effect $\mathtt{2\underline{T}} \overset{\mathtt{b}}{\leftarrow} \mathtt{1\underline{a}T} \leftarrow \mathtt{1\underline{c}caT}$

$$\mathtt{1}\alpha\mathtt{\underline{c}caT} \leftarrow \mathtt{2}\alpha\mathtt{c\underline{c}aT} \leftarrow \mathtt{2}\alpha\mathtt{\underline{b}caT} \begin{cases} \overset{\mathtt{b}}{\leftarrow} \mathtt{1\underline{a}bcaT}\\ \overset{\mathtt{c}}{\leftarrow} \mathtt{2\underline{b}bcaT}\end{cases} \tag{73}$$

This shortens to

$$\mathtt{1\underline{c}cT} \begin{cases} \leftarrow \mathtt{1\underline{a}bcT}\\ \leftarrow \mathtt{2\underline{b}bcT}\end{cases} \tag{74}$$

without any RCS's. Equation (74) will be called LIGR's 13 and 14 respectively. Consider the sequence $3 \cdot 3 \cdot 3$ with effect $\mathtt{1\underline{T}} \overset{\mathtt{b}}{\leftarrow} \mathtt{1\underline{c}T} \leftarrow \mathtt{1\underline{c}ccT}$

$$\mathtt{1}\alpha\mathtt{\underline{c}ccT} \leftarrow \mathtt{2}\alpha\mathtt{c\underline{c}cT} \leftarrow \mathtt{2}\alpha\mathtt{\underline{b}ccT} \begin{cases} \overset{\mathtt{b}}{\leftarrow} \mathtt{1\underline{a}bccT}\\ \overset{\mathtt{c}}{\leftarrow} \mathtt{2\underline{b}bccT}\end{cases} \tag{75}$$

which shortens to 13 and 14 above, without any RCS's. Next consider $11 \cdot 3$ which is $\mathtt{1\underline{c}aaT} \leftarrow \mathtt{1\underline{a}badT} \leftarrow \mathtt{1\underline{c}abadT}$. Applying F gives

$$\mathtt{1}\alpha\mathtt{\underline{c}abadT} \leftarrow \begin{cases} \overset{\mathtt{b}}{\leftarrow} \mathtt{2\underline{a}cdbadT}\\ \overset{\mathtt{c}}{\leftarrow} \mathtt{3\underline{c}cdbadT}\\ \leftarrow \mathtt{1}\alpha\mathtt{cdbd\underline{b}T}\end{cases} \tag{76}$$

The minimal forms of the first two results are the LIGR's 9 and 10. The sequence $13 \cdot 3$ is $\mathtt{1\underline{c}cT} \leftarrow \mathtt{1\underline{a}bcT} \leftarrow \mathtt{1\underline{c}abcT}$. F gives

$$\mathtt{1}\alpha\mathtt{\underline{c}abcT} \leftarrow \begin{cases} \mathtt{2\underline{a}cdbcT}\\ \mathtt{3\underline{c}cdbcT}\\ \mathtt{2}\alpha\mathtt{cdb\underline{c}T}\end{cases} \tag{77}$$

which when shortened give LIGR's $\mathtt{9}$ and $\mathtt{10}$ and an RCS. Only $\mathtt{3}$ can precede $\mathtt{13}$ and $\mathtt{3{\cdot}13{\cdot}3}$ is $\mathtt{1\underline{c}T} \leftarrow \mathtt{1\underline{c}cT} \leftarrow \mathtt{1\underline{c}abcT}$. Note that here an extra symbol $\mathtt{c}$ was needed in order that the RHS of $\mathtt{3}$ matched the LHS of $\mathtt{13\cdot 3}$. $\mathtt{F}$ gives the same result as above. Of the LIGR's that can precede $\mathtt{3}$ only $\mathtt{3}$ is compatible because of the symbol $\mathtt{c}$ on left and this gives $\mathtt{3\cdot 3\cdot 13\cdot 3}$ which is $\mathtt{1\underline{T}} \leftarrow \mathtt{1\underline{c}T} \leftarrow \mathtt{1\underline{c}abcT}$ which again gives the same result of $\mathtt{F}$. $\mathtt{1\cdot 3\cdot 3\cdot 13\cdot 3}$ is $\mathtt{2\underline{T}} \leftarrow \mathtt{1\underline{a}T} \leftarrow \mathtt{1\underline{c}abcaT}$. $\mathtt{F}$ gives no RCS's and LIGR's $\mathtt{20}$ and $\mathtt{21}$. $\mathtt{3{\cdot}3{\cdot}3{\cdot}13{\cdot}3}$ is $\mathtt{1\underline{T}} \leftarrow \mathtt{1\underline{c}T} \leftarrow \mathtt{1\underline{c}abccT}$. $\mathtt{F}$ gives LIGR's $\mathtt{20}$ and $\mathtt{21}$ and no RCS's, and likewise for $\mathtt{11\cdot 3\cdot 3\cdot 13\cdot 3}$. It is now obvious that the same result will come from $\mathtt{13\cdot 3\cdot 3\cdot 13\cdot 3}$ because it starts from a CS that shares in common with the previous case the string beyond which the computation to get these results does not go. The sequences $\mathtt{11\cdot 3\cdot 13\cdot 3}$ are $\mathtt{13\cdot 3\cdot 13\cdot 3}$ are not possible because of other compatibility restrictions based non-adjacent LIGR's.

7.1.4 Sequences ending with LIGR $\mathtt{4}$

LIGR $\mathtt{4}$ is $\mathtt{3\underline{T}} \overset{\mathtt{b}}{\leftarrow} \mathtt{2\underline{a}T}$. $\mathtt{F}$ gives

$$\mathtt{2}\alpha\mathtt{\underline{a}T}\begin{cases} \overset{\mathtt{b}}{\leftarrow} \mathtt{1\underline{a}aT} \\ \overset{\mathtt{c}}{\leftarrow} \mathtt{2\underline{b}aT} \end{cases} \tag{78}$$

which shortens to $\mathtt{1}$ and $\mathtt{5}$, and this result could depend on the leftmost symbol of $\mathtt{T}$ because right-moving reverse steps could occur. Therefore LIGR's preceding $\mathtt{4}$ must be considered. $\mathtt{8\cdot 4}$ is $\mathtt{3\underline{T}} \leftarrow \mathtt{3\underline{c}T} \leftarrow \mathtt{2\underline{a}cT}$, and applying $\mathtt{F}$ gives no LIGR's or RCS's. $\mathtt{10\cdot 4}$ is $\mathtt{1\underline{c}aT} \leftarrow \mathtt{3\underline{c}cdT} \leftarrow \mathtt{2\underline{a}ccdT}$ and $\mathtt{F}$ gives no LIGR's or RCS's. Therefore these results show that $\{\mathtt{8},\mathtt{10}\}\cdot\mathtt{4} \overset{\mathtt{F}}{\Rightarrow} \{\mathtt{1},\mathtt{5}\}$ only.

7.1.5 Sequences ending with LIGR $\mathtt{5}$

LIGR $\mathtt{5}$ is $\mathtt{2\underline{T}} \leftarrow \mathtt{2\underline{b}T}$ and applying $\mathtt{F}$ gives $\mathtt{2}\alpha\mathtt{\underline{b}T}\begin{cases} \overset{\mathtt{b}}{\leftarrow} \mathtt{1\underline{a}bT} \\ \overset{\mathtt{c}}{\leftarrow} \mathtt{2\underline{b}bT} \end{cases}$ which shortens to $\mathtt{1}$ and $\mathtt{5}$ so $\mathtt{5} \overset{\mathtt{F}}{\Rightarrow} \{\mathtt{1},\mathtt{5}\}$. The sequence $\mathtt{4\cdot 5}$ is $\mathtt{3\underline{T}} \leftarrow \mathtt{2\underline{a}T} \leftarrow \mathtt{2\underline{b}aT}$. Applying $\mathtt{F}$ starts from $\mathtt{2}\alpha\mathtt{\underline{b}aT}$ and gives no new LIGR's and no RCS's. This is likewise true for $\mathtt{5}$,$\mathtt{9}$ and $\mathtt{12}$ $\mathtt{14}$ and $\mathtt{20}$ preceding $\mathtt{5}$ and all follow from the fact that 2_a and 2_b cannot be arrived at from a step of the TM, so all sequences ending with $\{\mathtt{4},\mathtt{5},\mathtt{9},\mathtt{12},\mathtt{14},\mathtt{20}\}\cdot\mathtt{5} \overset{\mathtt{F}}{\Rightarrow} \{\mathtt{1},\mathtt{5}\}$.

7.1.6 Sequences ending with LIGR $\mathtt{6}$

LIGR $\mathtt{6}$ is $\mathtt{1\underline{T}} \leftarrow \mathtt{2T\underline{c}}$. Applying $\mathtt{F}$ gives $\mathtt{2T\underline{c}}\alpha \leftarrow \emptyset$. A left-moving reverse step could occur depending on the rightmost symbol of $\mathtt{T}$ so this needs to be specialised by considering all possible previous LIGR's i.e $\mathtt{2}$ and $\mathtt{15}$. The sequence $\mathtt{2\cdot 6}$ is $\mathtt{3\underline{T}} \leftarrow \mathtt{1T\underline{b}} \leftarrow \mathtt{2Tb\underline{c}}$ and applying $\mathtt{F}$ gives $\mathtt{2Tb\underline{c}}\alpha \leftarrow \mathtt{1T\underline{a}c}\alpha$. By

Lemma 7.7 this cannot lead to new LIGR's however the string is extended by preceding LIGR's in the sequence.

7.1.7 Sequences ending with LIGR 7

LIGR 7 is $\texttt{1}\underline{\texttt{T}} \leftarrow \texttt{3T}\underline{\texttt{d}}$, and $\texttt{3T}\underline{\texttt{d}}\alpha \overset{\texttt{b}}{\leftarrow} \texttt{1Td}\underline{\texttt{b}}$ which shortens to $\texttt{3}\underline{\texttt{T}} \leftarrow \texttt{1T}\underline{\texttt{b}}$ i.e. 2, so $\texttt{7} \overset{\texttt{F}}{\Rightarrow} \{\texttt{2}\}$. Because of this, from all subsequent specialisations of this resulting from LIGR's preceding 7, all lead under F to LIGR 2. Only 2 and 15 can precede 7 and $\texttt{2}\cdot\texttt{7}$ is $\texttt{3}\underline{\texttt{T}} \leftarrow \texttt{1T}\underline{\texttt{b}} \leftarrow \texttt{3Tb}\underline{\texttt{d}}$ and the backward search is just $\texttt{3Tb}\underline{\texttt{d}}\alpha \leftarrow \texttt{2T}\underline{\texttt{a}}\texttt{d}\alpha$ giving an RCS so the grand search continues back.

LIGR 2 can be preceded by 7 and 16, 18, 22, 24, 26. The sequence $\texttt{7}\cdot\texttt{2}\cdot\texttt{7}$ is $\texttt{1}\underline{\texttt{T}} \leftarrow \texttt{3T}\underline{\texttt{d}} \leftarrow \texttt{3Tdb}\underline{\texttt{d}}$ and $\texttt{3Tdb}\underline{\texttt{d}}\alpha \leftarrow \begin{cases}\texttt{1T}\underline{\texttt{a}}\texttt{ad}\alpha\\ \texttt{1Tdbd}\underline{\texttt{b}}\end{cases}$ giving the LIGR 2 again and an RCS. Continuing back gives the sequence $\texttt{2}\cdot\texttt{7}\cdot\texttt{2}\cdot\texttt{7}$ which is $\texttt{3}\underline{\texttt{T}} \leftarrow \texttt{3Tbdb}\underline{\texttt{d}}$. The backward search gives $\texttt{3Tbdb}\underline{\texttt{d}}\alpha \leftarrow \begin{cases}\texttt{1T}\underline{\texttt{c}}\texttt{aad}\alpha\\ \texttt{3Tbadb}\underline{\texttt{d}}\\ \texttt{2Tbadb}\underline{\texttt{c}}\end{cases}$ for which it is easy to check that actually both the d's must be b's (if either of them are a a proper subset of the results are obtained and if both are b all these results are obtained) so this gives the two LIGR's 22 and 23 and an RCS so the search continues back. Therefore $\texttt{7}\cdot\texttt{2}\cdot\texttt{7}\cdot\texttt{2}\cdot\texttt{7}$ is $\texttt{1}\underline{\texttt{T}} \leftarrow \texttt{3Tdbdb}\underline{\texttt{d}}$ and the backward search gives

$$\texttt{3Tdbdb}\underline{\texttt{d}}\alpha \leftarrow \texttt{1Td}\underline{\texttt{c}}\texttt{aad}\alpha \leftarrow \begin{cases}\texttt{1T}\underline{\texttt{c}}\texttt{caad}\alpha\\ \texttt{2T}\underline{\texttt{a}}\texttt{cdad}\alpha\\ \texttt{1T}\underline{\texttt{a}}\texttt{badd}\alpha\ .\\ \texttt{3Tdcbdb}\underline{\texttt{d}}\\ \texttt{2Tdcbdb}\underline{\texttt{c}}\end{cases} \tag{79}$$

together with LIGR's 2, 22, 23 that are ignored as a result of the first substitution in (79).

Every possible combination of the d's in the starting CS $\texttt{3Tdbdbd}\underline{\texttt{d}}$ being a or b gives a result that is included here and all the results here are obtained if all these d's are b. Two of the RCS's cannot lead to LIGR's because of Lemma 7.5 so the remaining results are the RCS $\texttt{1T}\underline{\texttt{a}}\texttt{badd}\alpha$ and the other two results which can be shortened to LIGR's 24 and 25. In order to obtain these the rightmost two of the d's in the starting CS must be b, thus the final result of the backward search from $\texttt{3Tdbdb}\underline{\texttt{d}}\alpha$ is $\texttt{1T}\underline{\texttt{a}}\texttt{badd}\alpha$ and LIGR's $2, 22-25$.

Continuing the grand backward search gives $(2{\cdot}7)^3$ which is $\texttt{3}\underline{\texttt{T}} \leftarrow \texttt{3Tbdbdb}\underline{\texttt{d}}$. Applying F to this gives, using the previous RCS, $\texttt{3Tbdbdb}\underline{\texttt{d}}\alpha \leftarrow \texttt{1Tb}\underline{\texttt{a}}\texttt{badd}\alpha \leftarrow \texttt{1T}\underline{\texttt{c}}\texttt{abadd}\alpha$ and no extra LIGR's. Continuing the grand backward search gives $7\cdot(2\cdot 7)^3$ which is $\texttt{1}\underline{\texttt{T}} \leftarrow \texttt{3Tdbdbdb}\underline{\texttt{d}}$. Applying F starts with $\texttt{3Tdbdbdb}\underline{\texttt{d}}\alpha \leftarrow \texttt{1Td}\underline{\texttt{c}}\texttt{abadd}\alpha \leftarrow$. As before the effect of this substitution is to ignore all the LIGR's obtained hitherto that must be included later. As above getting all these results requires the two middle d's to be b and is independent of whether the first d is a or b because it cannot be altered and will be removed in obtaining the LIGR's at this stage, so these three d's can be put equal to b so there are just 2 cases, the last d is a or b. For case a, there are many RCS's all of which are ruled out by Lemma 7.5 and 43 CS's that give the LIGR's $2, 15-19$. For case b there are fewer RCS's all of which are again ruled out by Lemma 7.5 and CS's that give LIGR's $2, 22-27$. These calculations are rather tedious but straightforward and easily done with the program [4]. The next sequences in the grand search $\{16, 18, 22, 24, 26\}\cdot(2\cdot 7)^3$, $15\cdot(7\cdot 2)^2\cdot 7$, $\{16, 18, 22, 24, 26\}\cdot(2\cdot 7)^2$, $15\cdot 7\cdot 2\cdot 7$ and $26\cdot 2\cdot 7$ all have in common, entries in column 2 of Table 3 origins of the form $\ldots \leftarrow \texttt{3T}\ldots(\texttt{db})^3\underline{\texttt{d}}$. As shown above these are all special cases of CS's to which when F is applied result in LIGR's $2, 15-19, 22-27$ and no usable RCS's. Therefore no LIGR's other than these can result from applying F to any of these sequences. The sequence $16\cdot 2\cdot 7$ is $\texttt{3Tb}^5\underline{\texttt{a}} \leftarrow \texttt{3Tca}^3\texttt{dbdb}\underline{\texttt{d}}$ and applying F gives (using (79)) $\texttt{3Tca}^3\texttt{dbdb}\underline{\texttt{d}}\alpha \leftarrow \texttt{1Tca}^3\underline{\texttt{a}}\texttt{badd}\alpha$ which cannot be continued back, and the LIGR's $2, 22-25$. The sequence $18\cdot 2\cdot 7$ gives $\texttt{3Tb}^5\underline{\texttt{a}} \leftarrow \texttt{3Tcd}\begin{Bmatrix}\texttt{ba}\\ \texttt{cb}\end{Bmatrix}\texttt{dbdb}\underline{\texttt{d}}$. Applying the same substitution to start gives $\texttt{1Tcd}\begin{Bmatrix}\texttt{ba}\\ \texttt{cb}\end{Bmatrix}\underline{\texttt{a}}\texttt{badd}\alpha$ from which the top part cannot be continued back, together with LIGR's $2, 22-25$. For the bottom part the starting point $\texttt{3Tcdcbdbdb}\underline{\texttt{d}}$ can be systematically done as follows.

$$\texttt{3T}^*\texttt{ab}\underline{\texttt{d}}\alpha \leftarrow \begin{cases}\texttt{1T}^*\texttt{abd}\underline{\texttt{b}}\\ \texttt{2T}^*\texttt{a}\underline{\texttt{a}}\texttt{da}\end{cases} \tag{80}$$

where the first result just gives LIGR 2 and the second cannot be continued back. Also the following is easily derived:

$$\texttt{3T}^*\texttt{abbb}\underline{\texttt{d}}\alpha \leftarrow \begin{cases} \texttt{3T}^*\texttt{acbdb}\underline{\texttt{d}} \\ \texttt{2T}^*\texttt{acbdb}\underline{\texttt{c}} \\ \texttt{2T}^*\texttt{abadb}\underline{\texttt{c}} \\ \texttt{3T}^*\texttt{abadb}\underline{\texttt{d}} \end{cases} \tag{81}$$

and the LIGR $\texttt{2}$ i.e. LIGR's $\texttt{2}, \texttt{22} - \texttt{25}$. The remaining cases $\texttt{3T}^*\texttt{cdcb}^5\underline{\texttt{d}}\alpha$ give results that can be deduced from the result for $\texttt{3T(db)}^3\underline{\texttt{d}}$ above which always gives results in $\texttt{2}, \texttt{15} - \texttt{19}, \texttt{22} - \texttt{27}$. This is because there are no longer and RCS's so the leftmost `d` is never reached so can be repalced by any symbols giving the same results. Pulling all this together shows that $\texttt{18} \cdot \texttt{2} \cdot \texttt{7}$ always gives results in this same set $\texttt{2}, \texttt{15} - \texttt{19}, \texttt{22} - \texttt{27}$.

$\texttt{22} \cdot \texttt{2} \cdot \texttt{7}$ is $\texttt{3Tbb}\underline{\texttt{b}} \leftarrow \texttt{3Tadbdb}\underline{\texttt{d}}$. Taking into account (80) and (81) the remaining results of the backward search are from $\texttt{3Tab}^4\underline{\texttt{d}}$. From (79), (80) and (81) as well, this gives altogether $\texttt{1Ta}\underline{\texttt{a}}\texttt{badd}\alpha$ which cannot be continued back and LIGR's $\texttt{2}, \texttt{22} - \texttt{25}$.

$\texttt{24} \cdot \texttt{2} \cdot \texttt{7}$ gives rise to the RCS's $\texttt{3Tcbdbdb}\underline{\texttt{d}}$. Taking out the subcases given by (80) and (81) the remaining cases are from $\texttt{3Tcb}^5\underline{\texttt{d}}$ which gives LIGR's $\texttt{2}, \texttt{15} - \texttt{19}, \texttt{22} - \texttt{27}$ because Lemma 7.8 by replacing the symbol `d` next to `T` by `c`, so the final result is LIGR's $\texttt{2}, \texttt{15} - \texttt{19}, \texttt{22} - \texttt{27}$. $\texttt{26} \cdot \texttt{2} \cdot \texttt{7}$ can have the backward search `F` done in the same way giving the same result. For $\texttt{15} \cdot \texttt{7}$ the same approach starts with taking out the cases with (80) and (81) leaving $\texttt{3Tc}\begin{Bmatrix}\texttt{db}\\ \texttt{aa}\end{Bmatrix}\texttt{b}^4\underline{\texttt{d}}$. The top part can be treated as above with the same result and the bottom part gives LIGR's $\texttt{2}, \texttt{22} - \texttt{25}$ as in the analysis of $\texttt{22} \cdot \texttt{2} \cdot \texttt{7}$. Combining all these results just gives LIGR's $\texttt{2}, \texttt{15} - \texttt{19}, \texttt{22} - \texttt{27}$.

The bottom part $\leftarrow \texttt{1Tcdc}\underline{\texttt{c}}\texttt{abadd}\alpha$ which is a special case of $\texttt{1T}\underline{\texttt{c}}\texttt{abadd}\alpha$ mentioned above just prior to the analysis of $\texttt{7} \cdot (\texttt{2} \cdot \texttt{7})^3$. In this analysis only LIGR's and RCS's are produced that are ruled out by Lemma 7.5 for generating LIGR's? See Lemma 7.8? $\texttt{22} \cdot \texttt{2} \cdot \texttt{7}$ gives $\texttt{3Tadbdb}\underline{\texttt{d}}$. The above analysis for $\texttt{3Tbdbdb}\underline{\texttt{d}}$ gives LIGR's $\texttt{2}, \texttt{22} - \texttt{25}$ therefore by Lemma 7.8 the same set of LIGR's is produced from the analysis of `F` applied to $\texttt{3Tadbdb}\underline{\texttt{d}}$. The RCS's: $\texttt{1T}\underline{\texttt{c}}\texttt{abadd}\alpha$ forward 1 step gives $\texttt{1Tb}\underline{\texttt{a}}\texttt{badd}\alpha$. Replace the first `a` by `b` and continue forward gives $\texttt{1T}\underline{\texttt{a}}\texttt{badd}\alpha \rightarrow \texttt{1}\underline{\texttt{T}}\texttt{ababa}\alpha$ so I suggest to get the new set of RCS's is to take the original ones, go forward 1 step, replace the symbol next to `T` then go forward and backward to find other RCS's with the pointer in this same position one from the end and go back over replaced symbol if possible by a reverse step to get the new RCS's.

************* which cannot be continued further back, because there are no RCS's and no new LIGR's so this branch of the grand search ends. $\texttt{7(b)} \cdot \texttt{2} \cdot \texttt{7(a)}$ which has the effect $\texttt{1}\underline{\texttt{T}} \leftarrow \texttt{3Tbb}\underline{\texttt{a}}$. The backward search is $\texttt{3Tbb}\underline{\texttt{a}}\alpha \leftarrow \texttt{2Tb}\underline{\texttt{a}}\texttt{a}\alpha \leftarrow \texttt{1T}\underline{\texttt{a}}\texttt{aa}\alpha$ i.e. there are no new LIGR's and one RCS. Continuing back gives $\texttt{2} \cdot \texttt{7(b)} \cdot \texttt{2} \cdot \texttt{7}$ with the effect $\texttt{3}\underline{\texttt{T}} \leftarrow \texttt{3Tbbb}\underline{\texttt{a}}$ with the backward search giving only the following result

$$\texttt{3Tbbb}\underline{\texttt{a}}\alpha \leftarrow \texttt{1T}\underline{\texttt{c}}\texttt{aaa}\alpha. \tag{82}$$

This again gives one RCS and no new LIGR's. Continuing, the sequence $\texttt{7}\cdot\texttt{2}\cdot\texttt{7(b)}\cdot\texttt{2}\cdot\texttt{7}$ has the effect $\texttt{1}\underline{\texttt{T}} \leftarrow \texttt{3Tabbb}\underline{\texttt{a}}$ with backward search that gives no RCS and no new LIGR after 3 steps and using (82).

By now it seems that a clear pattern has emerged with LIGR's 2 and 7(b) alternating, and with LIGR 7 terminating the sequences, however it is not yet clear how an induction proof can be completed showing this generally. Attempts to do so initially failed with the wrong inductive hypothesis because insufficient symbols were used. These attempts forced a consideration of further iterations of the basic procedure as follows.

The sequence $\texttt{7(b)}\cdot\texttt{2}\cdot\texttt{7(b)}\cdot\texttt{2}\cdot\texttt{7}$ has the effect $\texttt{1}\underline{\texttt{T}} \leftarrow \texttt{3Tbbbb}\underline{\texttt{a}}$ with backward search results using (82) for the first step giving

$$\texttt{3Tb}^4\underline{\texttt{a}}\alpha \leftarrow \texttt{1Tb}\underline{\texttt{c}}\texttt{aaa}\alpha \leftarrow \begin{cases} \texttt{1T}\underline{\texttt{c}}\texttt{caaa}\alpha \\ \texttt{2T}\underline{\texttt{a}}\texttt{cdaa}\alpha \\ \texttt{1T}\underline{\texttt{a}}\texttt{bada}\alpha \end{cases}. \tag{83}$$

Here there are three RCS's and no new LIGR's but the first two of these can be discontinued because Lemma 7.5 implies that these RCS's cannot lead to new LIGR's.

Continuing gives the sequence $\texttt{2}\cdot\texttt{7(b)}\cdot\texttt{2}\cdot\texttt{7(b)}\cdot\texttt{2}\cdot\texttt{7}$ with the effect $\texttt{3}\underline{\texttt{T}} \leftarrow \texttt{3Tb}^5\underline{\texttt{a}}$ and the backward search started using (83) gives

$$\texttt{3Tb}^5\underline{\texttt{a}}\alpha \leftarrow \texttt{1Tb}\underline{\texttt{a}}\texttt{bada}\alpha \leftarrow \texttt{1T}\underline{\texttt{c}}\texttt{abada}\alpha. \tag{84}$$

Continuing gives the sequence $\texttt{7(b)}\cdot\texttt{2}\cdot\texttt{7(b)}\cdot\texttt{2}\cdot\texttt{7(b)}\cdot\texttt{2}\cdot\texttt{7}$ with effect $\texttt{1}\underline{\texttt{T}} \leftarrow \texttt{3Tb}^6\underline{\texttt{a}}$. The results of the backward search $\Delta\texttt{F}_1$ and $\texttt{F}_2$ for $\texttt{7(b)}\cdot\texttt{2}\cdot\texttt{7(b)}\cdot\texttt{2}\cdot\texttt{7(b)}\cdot\texttt{2}\cdot\texttt{7}$ are given by

$$\texttt{3Tb}^6\underline{\texttt{a}}\alpha \leftarrow \texttt{1Tb}\underline{\texttt{c}}\texttt{abada}\alpha \leftarrow \begin{cases} \texttt{1Tbc}\begin{Bmatrix}\texttt{db}\\ \texttt{aa}\end{Bmatrix}\texttt{dbd}\underline{\texttt{b}} \\[2ex] \texttt{Tbca}^3\texttt{db}\begin{Bmatrix}\texttt{2}\underline{\texttt{c}}\\ \texttt{3}\underline{\texttt{d}}\end{Bmatrix} \\[2ex] \texttt{Tbcd}\begin{Bmatrix}\texttt{ba}\\ \texttt{cb}\end{Bmatrix}\texttt{db}\begin{Bmatrix}\texttt{3}\underline{\texttt{d}}\\ \texttt{2}\underline{\texttt{c}}\end{Bmatrix} \end{cases} \tag{85}$$

These results are clearly very important and to make them quite clear they will be expressed as

Lemma 7.10. *Every reverse computation path represented in the second part of* (85) *leads either to one of the RHS's which when expressed with the least number of symbols gives one of the set of LIGR's* 15 − 19 *or ends at a point* *where no further reverse computation steps are possible or (by Lemma 7.5) to a CS where no reverse computation path can lead to the symbol α (giving a LIGR) regardless of how many more symbols are added on the left i.e. regardless of how many LIGR's are prepended to the the sequence being considered. This includes the single reverse TM step to the left. Thus no useful RCS's are produced.*

Note that whenever there are multiple arrays with alternatives in the same expression in (85), the alternatives can all be chosen independently of each other. LIGR's $\mathtt{15}-\mathtt{19}$ seem to be much deeper results not obtainable from the simpler technique based on (1) unless much longer sequences are considered that will require a lot more resources (time and memory space) to find that earlier method.

Next consider the sequence $\mathtt{15}\cdot\mathtt{7(b)}\cdot\mathtt{2}\cdot\mathtt{7(b)}\cdot\mathtt{2}\cdot\mathtt{7(a)}$ with effect $\mathtt{3Tb^5\underline{a}} \leftarrow \mathtt{1Tc}\left\{\begin{matrix}\mathtt{db}\\\mathtt{aa}\end{matrix}\right\}\mathtt{dbd\underline{b}} \leftarrow \mathtt{3Tc}\left\{\begin{matrix}\mathtt{db}\\\mathtt{aa}\end{matrix}\right\}\mathtt{dbdbb^4\underline{a}}$. Applying $\mathtt{F}$ using (84) gives $\mathtt{3Tc}\left\{\begin{matrix}\mathtt{db}\\\mathtt{aa}\end{matrix}\right\}\mathtt{dbdb^5\underline{a}}\alpha \leftarrow \mathtt{1Tc}\left\{\begin{matrix}\mathtt{db}\\\mathtt{aa}\end{matrix}\right\}\mathtt{dbd\underline{c}abada}\alpha \leftarrow \ldots$. If the next reverse TM step is left then by Lemma 7.10 no more LIGR's can result from it, so this branch can be discontinued. This can only happen if $\mathtt{d}=\mathtt{b}$. If $\mathtt{d}=\mathtt{a}$ the same result holds because of Lemma 7.10. Consider $\mathtt{16}\cdot\mathtt{2}\cdot\mathtt{7(b)}\cdot\mathtt{2}\cdot\mathtt{7(a)}$ giving $\mathtt{3Tb^5\underline{a}} \leftarrow \mathtt{3Tca^3dbd\underline{d}} \leftarrow \mathtt{3Tca^3dbdb^3\underline{a}}$. $\mathtt{F}$ gives $\mathtt{3Tca^3dbdb^3\underline{a}}\alpha \leftarrow \mathtt{1Tca^3d\underline{c}abada}\alpha$ using (84) if the rightmost $\mathtt{d}=\mathtt{b}$. This by Lemma 7.10 gives no new results. If that $\mathtt{d}=\mathtt{a}$ the backward search gives starting from $\mathtt{3Tca^3dbab^3\underline{a}}\alpha \leftarrow \mathtt{1Tca^3dba\underline{c}a^3}\alpha$ (by (82)) gives just $\mathtt{2Tca^3dba\underline{b}ada}\alpha$ after 6 steps and stops so there are no new RCS's or LIGR's produced. Starting from $\mathtt{23}\cdot\mathtt{2}\cdot\mathtt{7(b)}\cdot\mathtt{2}\cdot\mathtt{7(a)}$ which is $\mathtt{3Tb^5\underline{a}} \leftarrow \mathtt{3Tcd}\left\{\begin{matrix}\mathtt{ba}\\\mathtt{cb}\end{matrix}\right\}\mathtt{db\underline{d}} \leftarrow \mathtt{3Tcd}\left\{\begin{matrix}\mathtt{ba}\\\mathtt{cb}\end{matrix}\right\}\mathtt{dbdb^3\underline{a}}$ the same result is clearly produced because the last 7 symbols are the same as the case above and the pointer does not leave this set during these computations. Consider $\mathtt{20}\cdot\mathtt{7(b)}\cdot\mathtt{2}\cdot\mathtt{7}$ giving $\mathtt{3Tb^5\underline{a}} \leftarrow \mathtt{3Tc}\left\{\begin{matrix}\mathtt{db}\\\mathtt{aa}\end{matrix}\right\}\mathtt{dbdbbb\underline{a}}$ and $\mathtt{F}$ gives the same result. $\mathtt{21}\cdot\mathtt{2}\cdot\mathtt{7}$ gives $\mathtt{3Tb^5\underline{a}} \leftarrow \mathtt{3Tca^3db\underline{d}} \leftarrow \mathtt{3Tca^3dbdb\underline{a}}$. All the results terminate giving no new results. $\mathtt{23}\cdot\mathtt{2}\cdot\mathtt{7}$ is $\mathtt{3Tb^5\underline{a}} \leftarrow \mathtt{3Tcd}\left\{\begin{matrix}\mathtt{db}\\\mathtt{aa}\end{matrix}\right\}\mathtt{db\underline{d}} \leftarrow \mathtt{3Tcd}\left\{\begin{matrix}\mathtt{db}\\\mathtt{aa}\end{matrix}\right\}\mathtt{dbdb\underline{a}}$. Again $\mathtt{F}$ gives no results. The same applies to $\mathtt{15}\cdot\mathtt{7(a)}$ i.e. $\mathtt{3Tb^5\underline{a}} \leftarrow \mathtt{1Tc}\left\{\begin{matrix}\mathtt{db}\\\mathtt{aa}\end{matrix}\right\}\mathtt{dbd\underline{b}} \leftarrow \mathtt{3Tc}\left\{\begin{matrix}\mathtt{db}\\\mathtt{aa}\end{matrix}\right\}\mathtt{dbdb\underline{a}}$.

$$\mathtt{3Tdbdb\underline{a}}\alpha \leftarrow \mathtt{1T\underline{a}bada}\alpha. \tag{86}$$

Only if both $\mathtt{d}$'s are $\mathtt{b}$ does this work so it is actually $\mathtt{3Tb^4\underline{a}} \leftarrow \mathtt{1T\underline{a}bada}$. From this it follows that $\mathtt{F}$ applied to $\mathtt{17}\cdot\mathtt{3}\cdot\mathtt{7}$ leads to no results.

7.1.8 Sequences ending with LIGR 7(b)

LIGR 7(b) is $\mathtt{1T} \leftarrow \mathtt{3T\underline{b}}$ and $\mathtt{3T\underline{b}}\alpha \overset{\mathtt{b}}{\leftarrow} \mathtt{1Tb\underline{b}}$ which shortens to LIGR 2 i.e. $7(\mathtt{b}) \overset{\mathtt{F}}{\Rightarrow} 2$. LIGR 7(b) can only be preceded by LIGR 2 and 15. The sequence $2 \cdot 7(\mathtt{b})$ which has the effect $\mathtt{3\underline{T}} \leftarrow \mathtt{3Tb\underline{b}}$ and $\mathtt{3Tb\underline{b}}\alpha \leftarrow \mathtt{2T\underline{a}b}\alpha$ an RCS, in addition to 2 as above, so the preceding LIGR needs to be considered i.e. 7, 7(b), 16, 18/19, 22, 24 and 26. The sequence $7(\mathtt{a}) \cdot 2 \cdot 7(\mathtt{b})$ has the effect $\mathtt{1\underline{T}} \leftarrow \mathtt{3T\underline{a}} \leftarrow \mathtt{3Tab\underline{b}}$ and

$$\mathtt{3Tab\underline{b}}\alpha \leftarrow \mathtt{2Ta\underline{a}b}\alpha \tag{87}$$

from which no reverse TM step can be made so $\ldots 7(\mathtt{a}) \cdot 2 \cdot 7(\mathtt{b}) \overset{\mathtt{F}}{\rightarrow} 2$ and no further extensions to the sequence are necessary. The sequence $7(\mathtt{b}) \cdot 2 \cdot 7(\mathtt{b})$ has the effect $\mathtt{1\underline{T}} \leftarrow \mathtt{3Tbb\underline{b}}$ and the backward search gives $\mathtt{3Tbb\underline{b}}\alpha \leftarrow \mathtt{1T\underline{a}ab}\alpha$ an RCS only, so the next sequence to be considered is $2 \cdot 7(\mathtt{b}) \cdot 2 \cdot 7(\mathtt{b})$ which has the effect $\mathtt{3\underline{T}} \leftarrow \mathtt{3Tbbb\underline{b}}$. The backward search gives

$$\mathtt{3Tbbb\underline{b}}\alpha \leftarrow \begin{cases} \mathtt{1T\underline{c}aab}\alpha \\ \mathtt{3Tbadb\underline{d}} \\ \mathtt{2Tbadb\underline{c}} \end{cases} \tag{88}$$

giving 1 RCS and 6 new LIGR's but only two using the abbreviation d, which are $\mathtt{3Tbb\underline{b}} \leftarrow \begin{cases} \mathtt{3Tadb\underline{d}} \\ \mathtt{2Tadb\underline{c}} \end{cases}$ which are 22 and 23 respectively. $7 \cdot 2 \cdot 7(\mathtt{b}) \cdot 2 \cdot 7(\mathtt{b})$ is $\mathtt{1\underline{T}} \leftarrow \mathtt{3T\underline{a}} \leftarrow \mathtt{3Tab^3\underline{b}}$. F gives ($\Delta\mathtt{F_1}$)

$$\mathtt{3Tab^3\underline{b}}\alpha \leftarrow \begin{cases} \mathtt{3Tacbdb\underline{d}} \\ \mathtt{2Tacbdb\underline{c}} \end{cases} \tag{89}$$

which can be shortened to $\mathtt{3Tb^3\underline{b}}\alpha \leftarrow \begin{cases} \mathtt{3Tcbdb\underline{d}} \\ \mathtt{2Tcbdb\underline{c}} \end{cases}$ which are LIGR's 24 and 25 respectively. $7(\mathtt{b}) \cdot 2 \cdot 7(\mathtt{b}) \cdot 2 \cdot 7(\mathtt{b})$ is $\mathtt{1\underline{T}} \leftarrow \mathtt{3T\underline{b}} \leftarrow \mathtt{3Tb^4\underline{b}}$. F gives $\mathtt{3Tb^4\underline{b}}\alpha \leftarrow \mathtt{1Tb\underline{c}aab}\alpha \leftarrow \begin{cases} \mathtt{1T\underline{a}badb}\alpha \\ \mathtt{2Tcbdb\underline{c}} \\ \mathtt{3Tcbdb\underline{d}} \end{cases}$ giving one RCS and two LIGR's which are 24 and 25 again. $(2 \cdot 7(\mathtt{b}))^3$ is $\mathtt{3\underline{T}} \leftarrow \mathtt{1T\underline{b}} \leftarrow \mathtt{3Tb^5\underline{b}}$. F gives

$$\mathtt{3Tb^5\underline{b}}\alpha \leftarrow \mathtt{1Tb\underline{a}badb}\alpha \leftarrow \mathtt{1T\underline{c}abadb}\alpha \tag{90}$$

i.e. just one RCS. $7(\mathtt{a}) \cdot (2 \cdot 7(\mathtt{b}))^3$ is $\mathtt{1\underline{T}} \leftarrow \mathtt{3T\underline{a}} \leftarrow \mathtt{3Tab^5\underline{b}}$. Applying F gives

$$\mathtt{3Tab^5\underline{b}}\alpha \leftarrow \mathtt{1Ta\underline{c}abadb}\alpha \leftarrow \begin{cases} \mathtt{3Tacadbdb\underline{d}} \\ \mathtt{2Tacadbdb\underline{c}} \end{cases} \tag{91}$$

which can be shortened by taking out the first a giving the LIGR's 30 and 31. At this point it will be useful to note that whatever symbol is put in place of the first a (absorbed into T below), if the pointer reaches that position in one step, the ca by Lemma 7.5 will prevent the pointer ever reaching α so no new LIGR's can result regardless of extra preceding LIGR's considered at the start. If the reversed TM could start by going right and then get to the position of the first a then this would have been indicated as an RCS in (91) which did not occur therefore

Lemma 7.11. *starting the backward search from* $\mathtt{3Tb^5\underline{b}}\alpha$ *gives no additional LIGR's i.e.* $\mathtt{\Delta F_1} = \emptyset$.

Consider $\mathtt{7(b)\cdot(2\cdot7(b))^3}$ which is $\mathtt{1\underline{T}} \leftarrow \mathtt{3T\underline{b}}$. Applying F gives a result very similar to the one above because just the first a is replaced by b. The search tree above has just two places where the pointer gets to the second symbol and could bring the first b into play. In both these case the second symbol is c. This ensures that a left reverse TM step would give a CS with cx to the right of the pointer and from there by Lemma 7.5 no continuation of the backward search with extra prepended LIGR's could give a new LIGR, so these branches should be discontinued. If the pointer goes right in these cases the results will be the same as above but with the first a replaced by b giving the same new LIGR's again and no RCS's. Again the same results are obtained if $\mathtt{(2\cdot7(b))^3}$ is preceded by 18/19, 22, 24 and 26. Next consider $\mathtt{15\cdot7(b)\cdot2\cdot7(b)\cdot2\cdot7(b)}$ which has the effect $\mathtt{3Tb^5\underline{a}} \leftarrow \mathtt{1Tc}\begin{Bmatrix}\mathtt{db}\\\mathtt{aa}\end{Bmatrix}\mathtt{dbd\underline{b}} \leftarrow \mathtt{3Tc}\begin{Bmatrix}\mathtt{db}\\\mathtt{aa}\end{Bmatrix}\mathtt{dbdbb^4\underline{b}}$. Applying F gives

$$\mathtt{3Tc}\begin{Bmatrix}\mathtt{db}\\\mathtt{aa}\end{Bmatrix}\mathtt{dbdbb^4\underline{b}}\alpha \leftarrow \mathtt{1Tc}\begin{Bmatrix}\mathtt{db}\\\mathtt{aa}\end{Bmatrix}\mathtt{dbdb\underline{a}badb}\alpha \leftarrow \mathtt{1Tc}\begin{Bmatrix}\mathtt{db}\\\mathtt{aa}\end{Bmatrix}\mathtt{dbd\underline{c}abadb}\alpha \tag{92}$$

which by (91) similarly to how (85) was used, cannot lead to any new LIGR's just 30 and 31 again and no RCS's other than those that do not lead to any more LIGR's. Consider the sequences $\{\mathtt{16, 18/19, 22, 24, 26}\}\cdot\mathtt{2\cdot7(b)\cdot2\cdot7(b)}$. It is easy to show that all these when F is applied, lead to special cases of $\mathtt{3Tdbdb^3\underline{b}}$. Referring the beginning of Section 7.1, if the rightmost d is a, j1 is 2 for 6 results and 3 for 6 results giving results which must be the same as 26 and 27 (for j1 = 2) and 24 and 25 for j1 = 3. If both d's are b the results of the backward search can be written compactly as $\mathtt{3Tbcadbdb\underline{d}}$ and $\mathtt{2Tbcadbdb\underline{c}}$, which because j1 = 5 can be shortened to length 7 giving the LIGR's which must be the same as 26 and 27.

Consider the sequence $\mathtt{15\cdot7(b)\cdot2\cdot7(b)}$ which is $\mathtt{3Tb^5\underline{a}} \leftarrow \mathtt{1Tc}\begin{Bmatrix}\mathtt{db}\\\mathtt{aa}\end{Bmatrix}\mathtt{dbd\underline{b}} \leftarrow \mathtt{3Tc}\begin{Bmatrix}\mathtt{db}\\\mathtt{aa}\end{Bmatrix}\mathtt{dbdbbb\underline{b}}$. For applying F, parts of this have already been done. Using (89) if the last d is a, F gives first LIGR's 24 and 25 with substrings of this obtained by truncating from the left having already been taken into account. Otherwise if the second to last d is a (91) gives LIGR's 26 and 27.

Also if it is $\mathtt{b}$ again the same LIGR's are obtained by applying the paragraph in the analysis of $\mathtt{7(b)}\cdot(\mathtt{2}\cdot\mathtt{7(b)})^3$. There are no new RCS's or LIGR's. Consider applying $\mathtt{F}$ to

$$\{\mathtt{16},\mathtt{18/19},\mathtt{22},\mathtt{24},\mathtt{26}\}\cdot\mathtt{2}\cdot\mathtt{7(b)}. \tag{93}$$

The effect of these all have in common the following symbols in the CS in the rhs's: $\mathtt{3dbdb\underline{b}}$ so try the reverse search starting from $\mathtt{3dbdb\underline{b}}\alpha$. If the second $\mathtt{d}$ is $\mathtt{a}$ it stops after one reverse step as in (87). Otherwise if the first $\mathtt{d}$ is $\mathtt{a}$ it leads to LIGR's $\mathtt{26}$ and $\mathtt{27}$ and in addition LIGR's $\mathtt{24}$ and $\mathtt{25}$ by (89). Some of the effects of (93) have in common $\mathtt{3ab^4\underline{b}}$ and $\mathtt{3ab^4\underline{b}}\alpha \leftarrow \mathtt{1a\underline{a}badb}\alpha$ where the reversed TM stops. $\mathtt{Tb^5\underline{b}}\alpha$ by Lemma 7.11 leads to no new LIGR's or RCS's. These results account for all possibilities in (93) so none of these gives rise to new LIGR's or RCS's. Consider $\mathtt{20}\cdot\mathtt{7(b)}$ having effect $\mathtt{3Tb^a\underline{a}} \leftarrow$ $\mathtt{1Tc}\begin{Bmatrix}\mathtt{db}\\ \mathtt{aa}\end{Bmatrix}\mathtt{dbd\underline{b}} \leftarrow \mathtt{3Tc}\begin{Bmatrix}\mathtt{db}\\ \mathtt{aa}\end{Bmatrix}\mathtt{dbdb\underline{b}}$. These rhs's are all of the form $\mathtt{3dbdb\underline{b}}$ so above argument shows that no new results can emerge.

7.1.9 Sequences ending with LIGR $\geq \mathtt{8}$

In Table 3, if the there are $\cdots$ in column 1, anything in column 3 will exclude RCS's that have been followed up and are also included in that row, so for example if column 3 had $\emptyset$ then no follow up results are needed in other rows just as if column 1 had no $\cdots$.

LIGR $\mathtt{8}$ is $\mathtt{3\underline{T}} \leftarrow \mathtt{3\underline{c}T}$ and $\mathtt{3}\alpha\mathtt{\underline{c}T}\begin{cases}\overset{\mathtt{b}}{\leftarrow}\mathtt{2\underline{a}cT}\\ \overset{\mathtt{c}}{\leftarrow}\mathtt{3\underline{c}cT}\end{cases}$ shortens to $\mathtt{4}$ and $\mathtt{8}$ so $\ldots\mathtt{8}\xrightarrow{\mathtt{F}}$ $\{\mathtt{4},\mathtt{8}\}$ but because the left end symbol of $\mathtt{T}$ is not specified, specialising it by including previous LIGR's could generate more results. $\mathtt{8}\cdot\mathtt{8}$ is $\mathtt{3\underline{T}} \leftarrow \mathtt{3\underline{c}T} \leftarrow$ $\mathtt{3\underline{c}cT}$ and $\mathtt{3}\alpha\mathtt{\underline{c}cT}$ gives nothing new so $\ldots\mathtt{8}\cdot\mathtt{8}\xrightarrow{\mathtt{F}}\{\mathtt{4},\mathtt{8}\}$. Similarly it follows that $\mathtt{10}\cdot\mathtt{8}$ under $\mathtt{F}$ produce no new results, so $\ldots\{\mathtt{8},\mathtt{10}\}\cdot\mathtt{8}\xrightarrow{\mathtt{F}}\{\mathtt{4},\mathtt{8}\}$. In a similar manner also the following can be easily established $\ldots\{\mathtt{9}\}\xrightarrow{\mathtt{F}}\{\mathtt{1},\mathtt{5}\}$
$\ldots\{\mathtt{10}\}\xrightarrow{\mathtt{F}}\{\mathtt{4},\mathtt{8}\}$
$\ldots\{\mathtt{11}\}\xrightarrow{\mathtt{F}}\{\mathtt{3}\}$
$\ldots\{\mathtt{12}\}\xrightarrow{\mathtt{F}}\{\mathtt{1},\mathtt{5}\}$
$\ldots\{\mathtt{13}\}\xrightarrow{\mathtt{F}}\{\mathtt{3}\}$
$\ldots\{\mathtt{14}\}\xrightarrow{\mathtt{F}}\{\mathtt{1},\mathtt{5}\}$
In all these cases, $\mathtt{F}$ produces results that cannot that when specialised to the cases where $\mathtt{T}$ has particular forms generate any new LIGR's from them. This results from the pointer not reaching next to the arbitrary string $\mathtt{T}$ at any point in these derivations. These are examples of the absence of RCS's in the result of $\mathtt{F}$ not giving any new LIGR's by specialising the original LIGR sequence by preceding it with another LIGR.

LIGR $\mathtt{24}$ can be preceded by $\mathtt{7(b)},\mathtt{18/19},\mathtt{24},\mathtt{26}$. $\mathtt{7(b)}$ preceding $\mathtt{24}$ makes

no difference to the result because $\mathtt{7(b)}\cdot\mathtt{24}$ has the effect $\mathtt{1\underline{T}} \leftarrow \mathtt{3T\underline{b}} \leftarrow \mathtt{3T^{*}cbdb\underline{d}}$ only if $\mathtt{T}$ ends in $\mathtt{bbb}$ i.e. $\mathtt{T} = \mathtt{T^{*}bbb}$. Because $\mathtt{18/19}$ in $\mathtt{18/19}\cdot\mathtt{24}$ must have the origin in state 3, its effect is $\mathtt{3Tb^{5}\underline{a}} \leftarrow \mathtt{3Tcd}\begin{Bmatrix}\mathtt{ba}\\ \mathtt{cb}\end{Bmatrix}\mathtt{db\underline{d}} \leftarrow \mathtt{3T^{*}cbdb\underline{d}}$ with $\mathtt{T^{*}} = \mathtt{Tcdc}$ i.e. $\mathtt{3Tcdccbdb\underline{d}}$ and $\mathtt{24}\cdot\mathtt{24}$ is $\mathtt{3Tb^{3}\underline{b}} \leftarrow \mathtt{3Tcbdb\underline{d}} \leftarrow \mathtt{3T^{*}cbdb\underline{d}}$ where $\mathtt{T^{*}} = \mathtt{Tc}$ i.e. $\mathtt{3Tccbdb\underline{d}}$. Likewise $\mathtt{26}\cdot\mathtt{24}$ is $\mathtt{3Tb^{5}\underline{b}} \leftarrow \mathtt{3Tcadcbdb\underline{d}}$. Also in each of these the second $\mathtt{d}$ from the right must actually be $\mathtt{b}$ otherwise no RCS's are obtained by Lemma 7.4. Thus these results are all special cases of $\mathtt{3Tccbbb\underline{d}}$ and $\mathtt{3Tcadcbbb\underline{d}}$ which need $\mathtt{F}$ to be applied to them in Table 3. To make this easier a computer program [4] was written based on a slightly modified version of the function “origins” appearing in [5] to obtain the results of all possible backward searches with the TM starting from any given input CS. This showed that the following CS's gave or not RCS's as follows starting from $\mathtt{3Tccbbb\underline{d}}$. The case when $\mathtt{d} = \mathtt{b}$ gives
$\mathtt{3Tccbbb\underline{b}}$ yes
$\mathtt{3Taccbbb\underline{b}}$ no
$\mathtt{3Tbccb^{3}\underline{b}}$ yes
$\mathtt{3Tabccb^{3}\underline{b}}$ no
$\mathtt{3Tbbccb^{3}\underline{b}}$ yes
$\mathtt{3Tabbccb^{3}\underline{b}}$ no
$\mathtt{3Tbbbccb^{3}\underline{b}}$ no
$\mathtt{3Tcbbccb^{3}\underline{b}}$ no
$\mathtt{3Tcbccb^{3}\underline{b}}$ no
$\mathtt{3Tcccb^{3}\underline{b}}$ yes
$\mathtt{3Tac^{3}b^{3}\underline{b}}$ no
$\mathtt{3Tbc^{3}b^{3}\underline{b}}$ yes
$\mathtt{3Tabc^{3}b^{3}\underline{b}}$ no
$\mathtt{3Tbbc^{3}b^{3}\underline{b}}$ yes
$\mathtt{3Tabbc^{3}b^{3}\underline{b}}$ no
$\mathtt{3Tbbbc^{3}b^{3}\underline{b}}$ no
$\mathtt{3Tcbbc^{3}b^{3}\underline{b}}$ no
$\mathtt{3Tcbc^{3}b^{3}\underline{b}}$ no
$\mathtt{3Tc^{4}b^{3}\underline{b}}$ yes etc.
This sequence of results was obtained by systematically searching in a similar manner to the main argument in section 5.5. By this point it looks as if a cycle could have been obtained. The reason for this was not difficult to find, it is because $\mathtt{3Tccbbb\underline{b}}\alpha \leftarrow \mathtt{2T\underline{b}bbadb}\alpha$ (only this RCS because of Lemma 7.5) together with LIGR's $\mathtt{2}, \mathtt{22} - \mathtt{25}$ identified by using the $\mathtt{j1}$ values, and it is possible to substitute $\mathtt{T} = \mathtt{T^{*}c^{n}}$ and derive

$$\mathtt{3T^{*}c^{n+2}b^{3}\underline{b}}\alpha \leftarrow \begin{cases}\mathtt{2T^{*}c^{n}\underline{b}bbadb}\alpha \leftarrow \mathtt{2T^{*}\underline{b}b^{n+2}adb}\alpha \\ \mathtt{1T^{*}c^{n+2}b^{4}\underline{b}}\text{ etc.}\end{cases} \quad (1\text{ RCS and LIGR's } \mathtt{2}, \mathtt{22}\text{–}\mathtt{25}) \tag{94}$$

for all $\mathtt{n} \geq \mathtt{0}$.

Also going back gives $\mathtt{3T^*ac^{n+2}b^3\underline{b}}\alpha \leftarrow$ no RCS's,
$\mathtt{3T^*bc^{n+2}b^3\underline{b}}\alpha \leftarrow \mathtt{1T^*\underline{a}b^{n+3}adb}\alpha$ (1 RCS), and going back gives
$\mathtt{3T^*abc^{n+2}b^3\underline{b}}\alpha \leftarrow$ no RCS's by Lemma 7.4,
$\mathtt{3T^*bbc^{n+2}b^3\underline{b}}\alpha \leftarrow \mathtt{1T^*\underline{c}ab^{n+3}adb}\alpha$ (1 RCS), and going back gives
$\mathtt{3T^*abbc^{n+2}b^3\underline{b}}\alpha \leftarrow$ no RCS's by Lemma 7.4,

$$\mathtt{3T^*b^3c^{n+2}b^3\underline{b}}\alpha \leftarrow \begin{cases} \mathtt{1T\underline{c}cab^{n+3}adb}\alpha \\ \mathtt{1Tbc\underline{b}b^{n+3}adb}\alpha \\ \mathtt{2T\underline{a}cdb^{n+3}adb}\alpha \\ \mathtt{1Tbc\underline{c}b^{n+3}adb}\alpha \end{cases} \text{(no usable RCS's)}$$

$$\mathtt{3T^*cbbc^{n+2}b^3\underline{b}}\alpha \leftarrow \begin{cases} \mathtt{1T^*cc\underline{c}b^{n+3}adb}\alpha \\ \mathtt{3T^*\underline{c}cdb^{n+3}adb}\alpha \\ \mathtt{1T^*cc\underline{b}b^{n+3}adb}\alpha \end{cases} \text{(no usable RCS's)}$$

$\mathtt{3T^*cbc^{n+2}b^3\underline{b}}\alpha \leftarrow \mathtt{1T^*c\underline{a}b^{n+3}adb}\alpha$ (no RCS's).
$\mathtt{3T^*cc^{n+2}b^3\underline{b}}\alpha$ i.e. (94) with $\mathtt{n}$ increased by 1.
In each of these short derivations only a few reverse steps of the TM are needed where as always Lemma 7.4 and Lemma 7.5 rule out continuing on many branches (each branch is represented by a final CS to help verify the results). All possible RCS's are indicated at each stage and no more LIGR's were found.

To find the RCS's associated with $\mathtt{3T^*cc^{n+2}b^3\underline{b}}\alpha$, the last CS to be examined, this is the same as the CS on the LHS of (94) with $\mathtt{n}$ increased by 1. This results in an infinite regress showing that no LIGR's result from $\mathtt{3T^*c^{n+2}b^3\underline{b}}\alpha$ other than those found at the start i.e. $2, 22 - 25$.

The same type of argument can be applied to the case where the last $\mathtt{d}$ is $\mathtt{a}$. This time the negative results for RCS's for the case where the starting CS contains any $\mathtt{a}$'s (see Lemma 7.4) will not be mentioned.

$$\mathtt{3Tc^{n+2}b^3\underline{a}}\alpha \leftarrow \begin{cases} \mathtt{1Tc^{n+2}b^3a\underline{b}} \\ \mathtt{3Tc^{n+2}ba\underline{d}a}\alpha \\ \mathtt{3Tc^{n+1}\underline{c}cda^2}\alpha \\ \mathtt{2T\underline{b}b^{n+2}ada}\alpha \end{cases} \text{i.e. 1 LIGR (2) and 1 RCS}$$

Going back gives $\mathtt{3Tbc^{n+2}b^3\underline{a}}\alpha \leftarrow \mathtt{1T\underline{a}b^{n+3}ada}\alpha$ (1 RCS)
Going back gives $\mathtt{3Tbbc^{n+2}b^3\underline{a}}\alpha \leftarrow \mathtt{1T\underline{c}ab^{n+3}ada}\alpha$ (1 RCS)

$$\text{Going back gives } \mathtt{3Tb^3c^{n+2}b^3\underline{a}}\alpha \leftarrow \begin{cases} \mathtt{1T\underline{c}cab^{n+3}ada}\alpha \\ \mathtt{2T\underline{a}cdb^{n+3}ada}\alpha \\ \mathtt{1Tbc\underline{b}b^{n+3}ada}\alpha \\ \mathtt{1Tbc\underline{c}b^{n+3}ada}\alpha \end{cases} \text{(no usable RCS's)}$$

$$\mathtt{3Tcb^2c^{n+2}b^3\underline{a}}\alpha \leftarrow \begin{cases} \mathtt{3T\underline{c}cdn^{n+3}ada}\alpha \\ \mathtt{1Tcc\underline{b}b^{n+3}ada}\alpha \\ \mathtt{1Tcc\underline{c}b^{n+3}ada}\alpha \end{cases} \text{(no usable RCS's)}$$

$\mathtt{3Tcbc^{n+2}b^3\underline{a}}\alpha \leftarrow \mathtt{1Tc\underline{a}b^{n+3}ada}\alpha$ (no RCS's)
$\mathtt{3Tc^{n+3}b^3\underline{a}}\alpha$ is as above with $\mathtt{n} \to \mathtt{n} + 1$ showing in all that only 1 LIGR (2) results from $\mathtt{3Tc^{n+2}b^3\underline{a}}\alpha$ for an arbitrary string $\mathtt{T}$. Applying a similar argument to $\mathtt{3Tcadcbbb\underline{d}}$ does not yield any RCS's regardless of whether either $\mathtt{d}$ is $\mathtt{a}$ or

`b`. In all of these results the LIGR's generated are the same `2`,`22` − `25` because these results have `j1` = `0`, `2` or `3` only and these LIGR's must be obtained because the starting `CS` is of the form `3T`*`bbbb`. This shows that any sequence of LIGR's ending with `24` under `F` only gives LIGR's `2`, `22` − `25`. The remaining results in Table 3 were obtained with the program [4] but could be obtained similarly to the above.

This version of the table is cut down to the minimum. The last but one column gives just all RCS's.

Table 3: The RCS's and LIGR's resulting from `F` applied to sequences of LIGR's

Possible sequences of LIGR's	It's effect	The RCS's excluding those because of Lemmas 7.4 and 7.5	LIGR's produced by `F`
$\mathtt{1}$	$\mathtt{2\underline{T} \xleftarrow{b} 1\underline{a}T}$	$\mathtt{1\alpha\underline{c}aT}$	$\mathtt{3}$
$\mathtt{4 \cdot 1}$	$\mathtt{3\underline{T} \xleftarrow{b} 2\underline{a}T \xleftarrow{b} 1\underline{a}aT}$	$\mathtt{3\alpha a\underline{d}T}$	$\mathtt{3}$
$\mathtt{8 \cdot 4 \cdot 1}$	$\mathtt{3\underline{T} \xleftarrow{c} 3\underline{c}T \leftarrow 1\underline{a}acT}$	$\emptyset$	$\mathtt{3}$
$\mathtt{10 \cdot 4 \cdot 1}$	$\mathtt{1\underline{c}aT \leftarrow 3\underline{c}cdT \leftarrow 1\underline{a}accdT}$	$\emptyset$	$\mathtt{3}$
$\mathtt{21 \cdot 4 \cdot 1}$	$\mathtt{1\underline{c}abcT \leftarrow 3\underline{c}ccdcT \leftarrow 1\underline{a}ac^3dcT}$	$\emptyset$	$\mathtt{3}$
$\mathtt{5 \cdot 1}$	$\mathtt{2\underline{T} \leftarrow 2\underline{b}T \leftarrow 1\underline{a}bT}$	$\emptyset$	$\mathtt{3}$
$\mathtt{9 \cdot 1}$	$\mathtt{1\underline{c}aT \leftarrow 2\underline{a}cdT \leftarrow 1\underline{a}acdT}$	$\emptyset$	$\mathtt{3}$
$\mathtt{12 \cdot 1}$	$\mathtt{1\underline{c}aaT \leftarrow 2\underline{b}badT \leftarrow 1\underline{a}bbadT}$	$\emptyset$	$\mathtt{3}$
$\mathtt{14 \cdot 1}$	$\mathtt{1\underline{c}cT \leftarrow 2\underline{b}bcT \leftarrow 1\underline{a}bbcT}$	$\emptyset$	$\mathtt{3}$
$\mathtt{20 \cdot 1}$	$\mathtt{1\underline{c}abcT \leftarrow 2\underline{a}ccdcT \leftarrow 1\underline{a}accdcT}$	$\emptyset$	$\mathtt{3}$
$\mathtt{2}$	$\mathtt{3\underline{T} \xleftarrow{b} 1T\underline{b}}$	$\begin{cases}\mathtt{3Tb\underline{d}\alpha}\\ \mathtt{2Tb\underline{c}\alpha}\end{cases}$	$\{\mathtt{6},\mathtt{7}\}$
$\mathtt{7 \cdot 2}$	$\mathtt{1\underline{T} \xleftarrow{a} 3T\underline{d} \leftarrow 1Td\underline{b}}$	$\mathtt{1T\underline{c}b\alpha}$	$\{\mathtt{6},\mathtt{7}\}$
$\mathtt{2 \cdot 7 \cdot 2}$	$\mathtt{3\underline{T} \xleftarrow{b} 1T\underline{b} \leftarrow 1Tbb\underline{b}}$	$\mathtt{1T\underline{c}cb\alpha}$	$\{\mathtt{6},\mathtt{7}\}$
$\mathtt{15 \cdot 7 \cdot 2}$	$\mathtt{3Tb^5\underline{a} \leftarrow 1Tc\begin{Bmatrix}db\\ aa\end{Bmatrix}dbd\underline{b}}$ $\mathtt{\leftarrow 1Tc\begin{Bmatrix}db\\ aa\end{Bmatrix}dbdbb\underline{b}}$	$\emptyset$	$\{\mathtt{6},\mathtt{7}\}$
$\mathtt{16 \cdot 2}$	$\mathtt{3Tb^5\underline{a} \leftarrow 3Tc\begin{Bmatrix}db\\ aa\end{Bmatrix}dbdb\underline{b}}$ $\mathtt{\leftarrow 1Tc\begin{Bmatrix}db\\ aa\end{Bmatrix}dbdbb\underline{b}}$	$\emptyset$	$\{\mathtt{6},\mathtt{7}\}$
$\mathtt{18 \cdot 2}$	$\mathtt{3Tb^5\underline{a} \leftarrow 3Tcd\begin{Bmatrix}ba\\ cb\end{Bmatrix}db\underline{d} \leftarrow}$ $\mathtt{1Tcd\begin{Bmatrix}ba\\ cb\end{Bmatrix}dbd\underline{b}}$	$\emptyset$	$\{\mathtt{6},\mathtt{7}\}$
$\{\mathtt{22},\mathtt{24},\mathtt{26}\} \cdot \mathtt{2}$	$\mathtt{\leftarrow 1T^*dbd\underline{b}}$		$\{\mathtt{6},\mathtt{7}\}$
$\mathtt{3}$	$\mathtt{1\underline{T} \xleftarrow{b} 1\underline{c}T}$	$\mathtt{1\alpha\underline{c}cT}$	$\mathtt{3}$
$\mathtt{1 \cdot 3}$	$\mathtt{2\underline{T} \leftarrow 1\underline{a}T \leftarrow 1\underline{c}aT}$	$\mathtt{3\alpha c\underline{d}T}$	$\mathtt{3}$
$\mathtt{4 \cdot 1 \cdot 3}$	$\mathtt{3\underline{T} \leftarrow 2\underline{a}T \leftarrow 1\underline{c}aaT}$	$\mathtt{3\alpha cb\underline{d}T}$	$\{\mathtt{3},\mathtt{9},\mathtt{10}\}$
$\mathtt{8 \cdot 4 \cdot 1 \cdot 3}$	$\mathtt{3\underline{T} \leftarrow 3\underline{c}T \leftarrow 1\underline{c}aacT}$	$\emptyset$	$\{\mathtt{3},\mathtt{9},\mathtt{10},\mathtt{11},\mathtt{12}\}$
$\mathtt{10 \cdot 4 \cdot 1 \cdot 3}$	$\mathtt{1\underline{c}aT \leftarrow 3\underline{c}cdT}$ $\mathtt{\leftarrow 1\underline{c}aaccdT}$	$\emptyset$	$\{\mathtt{3},\mathtt{9},\mathtt{10},\mathtt{11},\mathtt{12}\}$
$\mathtt{21 \cdot 4 \cdot 1 \cdot 3}$	$\mathtt{1\underline{c}abcT \leftarrow 3\underline{c}ccdcT}$ $\mathtt{\leftarrow 1\underline{c}aac^4dcT}$	$\emptyset$	$\{\mathtt{3},\mathtt{9},\mathtt{10},\mathtt{11},\mathtt{12}\}$

$5\cdot1\cdot3$	$2\underline{T}\leftarrow 2\underline{b}T\leftarrow 1\underline{c}abT$	$1\alpha cd\underline{b}T$	$\{3,9,10\}$
$4\cdot5\cdot1\cdot3$	$3\underline{T}\leftarrow 2\underline{a}T\leftarrow 1\underline{c}abaT$	$3\alpha cdb\underline{d}T$	$\{3,9,10\}$
$8\cdot4\cdot5\cdot1\cdot3$	$3\underline{T}\leftarrow 3\underline{c}T\leftarrow 1\underline{c}abacT$	$2\alpha cadb\underline{c}T$	$\{3,9,10\}$
$10\cdot4\cdot5\cdot1\cdot3$	$1\underline{c}aT\leftarrow 3\underline{c}cdT\leftarrow 1\underline{c}abaccdT$	$\emptyset$	$\{3,9,10\}$
$21\cdot4\cdot5\cdot1\cdot3$	$1\underline{c}abcT\leftarrow 3\underline{c}c^2dcT$ $\leftarrow 1\underline{c}abac^3dcT$	$\emptyset$	$\{3,9,10\}$
$5\cdot5\cdot1\cdot3$	$2\underline{T}\leftarrow 2\underline{b}T\leftarrow 1\underline{c}abbT$	$\emptyset$	$\{3,9,10\}$
$9\cdot5\cdot1\cdot3$	$1\underline{c}aT\leftarrow 2\underline{a}cdT\leftarrow 1\underline{c}abacdT$	$\emptyset$	$\{3,9,10\}$
$12\cdot5\cdot1\cdot3$	$1\underline{c}aaT\leftarrow 2\underline{b}badT\leftarrow 1\underline{c}ab^3adT$	$\emptyset$	$\{3,9,10\}$
$14\cdot5\cdot1\cdot3$	$1\underline{c}cT\leftarrow 2\underline{b}bcT\leftarrow 1\underline{c}ab^3cT$	$\emptyset$	$\{3,9,10\}$
$20\cdot5\cdot1\cdot3$	$1\underline{c}abcT\leftarrow 2\underline{a}ccdcT$ $\leftarrow 1\underline{c}abaccdcT$	$\emptyset$	$\{3,9,10\}$
$9\cdot1\cdot3$	$1\underline{c}aT\leftarrow 2\underline{a}cdT\leftarrow 1\underline{c}aacdT$	$\emptyset$	$\{3,9,10,11,12\}$
$12\cdot1\cdot3$	$1\underline{c}aaT\leftarrow 2\underline{b}badT\leftarrow 1\underline{c}abbadT$	$\emptyset$	$\{3,9,10\}$
$20\cdot1\cdot3$	$1\underline{c}abcT\leftarrow 2\underline{a}ccdcT$ $\leftarrow 1\underline{c}abaccdcT$	$\emptyset$	$\{3,9,10\}$
$3\cdot3$	$1\underline{T}\overset{b}{\leftarrow} 1\underline{c}T\leftarrow 1\underline{c}cT$	$2\alpha c\underline{c}T$	$\{3\}$
$1\cdot3\cdot3$	$2\underline{T}\overset{b}{\leftarrow} 1\underline{a}T\leftarrow 1\underline{c}caT$	$\emptyset$	$\{3,13,14\}$
$3\cdot3\cdot3$	$1\underline{T}\overset{b}{\leftarrow} 1\underline{c}T\leftarrow 1\underline{c}ccT$	$\emptyset$	$\{3,13,14\}$
$11\cdot3$	$1\underline{c}aaT\leftarrow 1\underline{a}badT\leftarrow 1\underline{c}abadT$	$1\alpha cdbd\underline{b}T$	$\{9,10\}$
$13\cdot3$	$1\underline{c}cT\leftarrow 1\underline{a}bcT\leftarrow 1\underline{c}abcT$	$2\alpha cdb\underline{c}T$	$\{9,10\}$
$3\cdot13\cdot3$	$1\underline{c}T\leftarrow 1\underline{c}cT\leftarrow 1\underline{c}abcT$	$2\alpha cdb\underline{c}T$	$\{9,10\}$
$3\cdot3\cdot13\cdot3$	$1\underline{T}\leftarrow 1\underline{c}T\leftarrow 1\underline{c}abcT$	$2\alpha cdb\underline{c}T$	$\{9,10\}$
$1\cdot3\cdot3\cdot13\cdot3$	$2\underline{T}\leftarrow 1\underline{a}T\leftarrow 1\underline{c}abcaT$	$\emptyset$	$\{9,10,20,21\}$
$\{3,11,13\}$ $\cdot3\cdot3\cdot13\cdot3$	$1\underline{T}\leftarrow 1\underline{c}T\leftarrow 1\underline{c}abccT$	$\emptyset$	$\{9,10,20,21\}$
4	$3\underline{T}\leftarrow 2\underline{a}T$	$\begin{cases}1\alpha\underline{a}T\\2\alpha\underline{b}T\end{cases}$	$\{1,5\}$
$8\cdot4$	$3\underline{T}\leftarrow 3\underline{c}T\leftarrow 2\underline{a}cT$	$\emptyset$	$\{1,5\}$
$10\cdot4$	$1\underline{c}aT\leftarrow 3\underline{c}cdT\leftarrow 2\underline{a}ccdT$	$\emptyset$	$\{1,5\}$
$21\cdot4$	$1\underline{c}abcT\leftarrow 3\underline{c}ccdcT\leftarrow 2\underline{a}c^3dcT$	$\emptyset$	$\{1,5\}$
5	$2\underline{T}\leftarrow 2\underline{b}T$	$\begin{cases}1\alpha\underline{a}T\\2\alpha\underline{b}T\end{cases}$	$\{1,5\}$
$\{4,5,9,12,14,20\}\cdot5$	$3\underline{T}\leftarrow 2\underline{a}T\leftarrow 2\underline{b}\ldots T$	$\emptyset$	$\{1,5\}$
6	$1\underline{T}\leftarrow 2T\underline{c}$	$\emptyset$	$\emptyset$
$\{2,15\}\cdot6$	$3\underline{T}\leftarrow 1T\underline{b}\leftarrow 2T\ldots b\underline{c}$	$\emptyset$	$\emptyset$
7	$1\underline{T}\leftarrow 3T\underline{d}$	$1Td\underline{b}$	$\{2\}$
$2\cdot7$	$3\underline{T}\leftarrow 1T\underline{b}\leftarrow 3Tb\underline{d}$	$2T\underline{a}d\alpha$	$\{2\}$
$7\cdot2\cdot7$	$1\underline{T}\leftarrow 3T\underline{a}\leftarrow 3Tdb\underline{d}$	$1T\underline{a}ad\alpha$	$\{2\}$
$2\cdot7\cdot2\cdot7$	$3\underline{T}\leftarrow 3Tbdb\underline{d}$	$1T\underline{c}aad\alpha$	$\{2,22,23\}$
$7\cdot2\cdot7\cdot2\cdot7$	$1\underline{T}\leftarrow 3Tdbdb\underline{d}$	$1T\underline{a}badd\alpha$	$\{2,22-25\}$
$2\cdot7\cdot2\cdot7\cdot2\cdot7$	$3\underline{T}\leftarrow 3Tbdbdb\underline{d}$	$1T\underline{c}abadd\alpha$	$\{2,22-25\}$
$7\cdot2\cdot7\cdot2\cdot7\cdot2\cdot7$	$1\underline{T}\leftarrow 3Tdbdbdb\underline{d}$	$\emptyset$	$\{2,15-19,22-27\}$
$\{16,18,22,24,$ $26\}\cdot(2\cdot7)^3$	$\leftarrow 3T(bd)^3\underline{d}$	$\emptyset$	$\{2,15-19,22-27\}$
	$1\underline{T}\leftarrow 3Tdbdbdb\underline{a}$	$\emptyset$	$\{2,15-19\}$
	$1\underline{T}\leftarrow 3Tdbdbdb\underline{b}$	$\emptyset$	$\{2,22-27\}$
$15\cdot7\cdot2\cdot7\cdot2\cdot7$	$3Tb^5\underline{a}\leftarrow 3Tc\begin{Bmatrix}db\\aa\end{Bmatrix}dbdbdbdb\underline{d}$	$\emptyset$	$\{2,15-19,22-27\}$
$16\cdot2\cdot7\cdot2\cdot7$	$3Tb^5\underline{a}\leftarrow 3Tca^3dbdbdb\underline{d}$	$\emptyset$	$\{2,15-19,22-27\}$

$\{18,22,24,26\} \cdot 2\cdot 7\cdot 2\cdot 7$	$3Tb^5\underline{d} \leftarrow 3Tcd\ldots dbdbdb\underline{d}$	$\emptyset$	$\{2,15-19,22-27\}$
$15\cdot 7\cdot 2\cdot 7$	$3Tb^5\underline{a} \leftarrow 3Tc\left\{\begin{matrix}db\\aa\end{matrix}\right\}dbdbdb\underline{d}$	$\emptyset$	$\{2,15-19,22-27\}$
$16\cdot 2\cdot 7$	$3Tb^5\underline{a} \leftarrow 3Tca^3dbdb\underline{d}$	$\emptyset$	$\{2,15-19,22-25\}$
$18\cdot 2\cdot 7$	$3Tb^5\underline{a} \leftarrow 3Tcd\left\{\begin{matrix}ba\\cb\end{matrix}\right\}dbdb\underline{d}$	$\emptyset$	$\{2,15-19,22-27\}$
$22\cdot 2\cdot 7$	$3Tbb\underline{b} \leftarrow 3Tadbdb\underline{d}$	$\emptyset$	$\{2,22-27\}$
$24\cdot 2\cdot 7$	$3Tb^3\underline{b} \leftarrow 3Tcbdbdb\underline{d}$	$\emptyset$	$\{2,15-19,22-27\}$
$26\cdot 2\cdot 7$	$3Tb^5\underline{b} \leftarrow 3Tcadbdbdb\underline{d}$		
$15\cdot 7$	$3Tb^5\underline{a} \leftarrow 3Tc\left\{\begin{matrix}db\\aa\end{matrix}\right\}dbdb\underline{d}$	$\emptyset$	$\{2,15-19,22-27\}$
	$\leftarrow 3Tab^4\underline{b}$	$\emptyset$	$\{2,22-25\}$
	$\leftarrow 3Tcb^5\underline{b}$	$\emptyset$	$\{2,22-27\}$
$\{16,18,22,24,26\}\cdot(2\cdot 7)^3$	$\leftarrow 3T\cdots db^5\underline{b}$	$\emptyset$	$\{2,22-27\}$
$\{16,18,22,24,26\}\cdot 2\cdot 7$	$\leftarrow 3\ldots\gamma ddbdb\underline{b}$	$\emptyset$	$\{2,22-27\}$
8	$3\underline{T} \leftarrow 3\underline{c}T$	$\begin{cases}2\alpha\underline{a}cT\\3\alpha\underline{c}cT\end{cases}$	$\{4,8\}$
$\{8,10,21\}\cdot 8$	$\leftarrow 3\underline{c}cT$	$\emptyset$	$\{4,8\}$
9	$1\underline{c}aT \leftarrow 2\underline{a}cdT$	$\emptyset$	$\{1,5\}$
10	$1\underline{c}aT \leftarrow 3\underline{c}cdT$	$\emptyset$	$\{4,8\}$
11	$1\underline{c}aaT \leftarrow 1\underline{a}badT$	$\emptyset$	$\{3\}$
12	$1\underline{c}aaT \leftarrow 2\underline{b}badT$	$\emptyset$	$\{1,5\}$
13	$1\underline{c}cT \leftarrow 1\underline{a}bcT$	$\emptyset$	$\{3\}$
14	$1\underline{c}cT \leftarrow 2\underline{b}bcT$	$\emptyset$	$\{1,5\}$
15	$3Tb^5\underline{a} \leftarrow 1Tc\left\{\begin{matrix}db\\aa\end{matrix}\right\}dbd\underline{b}$	$\emptyset$	$\{6,7\}$
16	$3Tb^5\underline{a} \leftarrow 3Tca^3db\underline{d}$	$\emptyset$	$\{2,22,23\}$
17	$3Tb^5\underline{a} \leftarrow 2Tca^3db\underline{c}$	$\emptyset$	$\emptyset$
18	$3Tb^5\underline{a} \leftarrow 3Tcd\left\{\begin{matrix}ba\\cb\end{matrix}\right\}db\underline{d}$	$\emptyset$	$\{2,22-25\}$
19	$3Tb^5\underline{a} \leftarrow 2Tcd\left\{\begin{matrix}ba\\cb\end{matrix}\right\}db\underline{c}$	$\emptyset$	$\{2,22-25\}$
20	$1\underline{c}abcT \leftarrow 2\underline{a}ccdcT$	$\emptyset$	$\{1,5\}$
21	$1\underline{c}abcT \leftarrow 3\underline{c}ccdcT$	$\emptyset$	$\{4,8\}$
22	$3Tbb\underline{b} \leftarrow 3Tadb\underline{d}$	$\emptyset$	$\{2,22,23\}$
23	$3Tbb\underline{b} \leftarrow 2Tadb\underline{c}$	$\emptyset$	$\emptyset$
$\cdots 28$	$3Tb^3\underline{b} \leftarrow 3Tcbdb\underline{d}$	$\emptyset$	$\{2,22-25\}$
25	$3Tb^3\underline{b} \leftarrow 2Tcbdb\underline{c}$	$\emptyset$	$\emptyset$
$\cdots 26$	$3Tb^5\underline{b} \leftarrow 3Tcadbdb\underline{d}$	$\emptyset$	$\{2,22-25\}$
27	$3Tb^5\underline{b} \leftarrow 2Tcadbdb\underline{c}$	$\emptyset$	$\emptyset$

The set of LIGR's is

$$
\begin{array}{ll}
1 & \mathtt{2\underline{T} \overset{b}{\leftarrow} 1\underline{a}T} \\
2 & \mathtt{3\underline{T} \overset{b}{\leftarrow} 1T\underline{b}} \\
3 & \mathtt{1\underline{T} \overset{b}{\leftarrow} 1\underline{c}T} \\
4 & \mathtt{3\underline{T} \overset{b}{\leftarrow} 2\underline{a}T} \\
5 & \mathtt{2\underline{T} \overset{c}{\leftarrow} 2\underline{b}T} \\
6 & \mathtt{1\underline{T} \overset{c}{\leftarrow} 2T\underline{c}} \\
7 & \mathtt{1\underline{T} \overset{a}{\leftarrow} 3T\underline{d}} \\
8 & \mathtt{3\underline{T} \overset{c}{\leftarrow} 3\underline{c}T} \\
9 & \mathtt{1\underline{c}aT \overset{b}{\leftarrow} 2\underline{a}cdT} \\
10 & \mathtt{1\underline{c}aT \overset{c}{\leftarrow} 3\underline{c}cdT} \\
11 & \mathtt{1\underline{c}aaT \overset{b}{\leftarrow} 1\underline{a}badT} \\
12 & \mathtt{1\underline{c}aaT \overset{c}{\leftarrow} 2\underline{b}badT} \\
13 & \mathtt{1\underline{c}cT \overset{b}{\leftarrow} 1\underline{a}bcT} \\
14 & \mathtt{1\underline{c}cT \overset{c}{\leftarrow} 2\underline{b}bcT} \\
15 & \mathtt{3Tb^5\underline{a} \overset{b}{\leftarrow} 1Tc \begin{Bmatrix} db \\ aa \end{Bmatrix} dbd\underline{b}} \\
16 & \mathtt{3Tb^5\underline{a} \overset{a}{\leftarrow} 3Tca^3dbd\underline{d}} \\
17 & \mathtt{3Tb^5\underline{a} \overset{c}{\leftarrow} 2Tca^3db\underline{c}} \\
18 & \mathtt{3Tb^5\underline{a} \overset{a}{\leftarrow} 3Tcd \begin{Bmatrix} ba \\ cb \end{Bmatrix} dbd\underline{d}} \\
19 & \mathtt{3Tb^5\underline{a} \overset{c}{\leftarrow} 2Tcd \begin{Bmatrix} ba \\ cb \end{Bmatrix} db\underline{c}} \\
20 & \mathtt{1\underline{c}abcT \overset{b}{\leftarrow} 2\underline{a}ccdcT} \\
21 & \mathtt{1\underline{c}abcT \overset{c}{\leftarrow} 3\underline{c}ccdcT} \\
22 & \mathtt{3Tbb\underline{b} \overset{a}{\leftarrow} 3Tadbd\underline{d}} \\
23 & \mathtt{3Tbb\underline{b} \overset{c}{\leftarrow} 2Tadb\underline{c}} \\
24 & \mathtt{3Tb^3\underline{b} \overset{a}{\leftarrow} 3Tcbdbd\underline{d}} \\
25 & \mathtt{3Tb^3\underline{b} \overset{c}{\leftarrow} 2Tcbdb\underline{c}} \\
26 & \mathtt{3Tb^5\underline{b} \overset{a}{\leftarrow} 3Tcadbdbd\underline{d}} \\
27 & \mathtt{3Tb^5\underline{b} \overset{c}{\leftarrow} 2Tcadbdb\underline{c}} \\
28 & \mathtt{1\underline{c}ababT \overset{b}{\leftarrow} 1\underline{a}bbccbT} \\
29 & \mathtt{1\underline{c}ababT \overset{c}{\leftarrow} 2\underline{b}bbccbT} \\
30 & \mathtt{1\underline{c}ababcT \overset{b}{\leftarrow} 1\underline{a}bbccacT} \\
31 & \mathtt{1\underline{c}ababcT \overset{c}{\leftarrow} 2\underline{b}bbccacT}
\end{array}
\tag{95}
$$

Because new LIGR's have been found, the "can possibly be preceded by" relation needs to be updated as follows

$\mathtt{1, 5}$	$\mathtt{4, 5, 9, 12, 14, 20, 29, 31}$	
$\mathtt{2}$	$\mathtt{7, 16, 18, 22, 24, 26}$	
$\mathtt{3}$	$\mathtt{1, 3, 11, 13}$	
$\mathtt{4, 8}$	$\mathtt{8, 10, 21}$	
$\mathtt{6, 7}$	$\mathtt{2, 15, 28, 30}$	
$\mathtt{9} - \mathtt{14, 20, 21}$	$\mathtt{3}$	(96)
$\mathtt{15, 16, 17, 18, 19}$	$\mathtt{7}$	
$\mathtt{20, 21}$	$\mathtt{3}$	
$\mathtt{22, 23}$	$\mathtt{7, 16, 22, 24, 26}$	
$\mathtt{24, 25}$	$\mathtt{7, 18, 24, 26}$	
$\mathtt{26, 27}$	$\mathtt{7}$	

The word "possibly" is included because there may be other symbols in either of the strings $\mathtt{T}$ that prevent a match of the sequences. Note that this refers to the order of the LIGR's which is reverse order of the TM computation.

The results summarised in Table 3 show that the set of LIGR's $\mathtt{1} - \mathtt{31}$ is closed under $\mathtt{F}$ and therefore by Theorem 7.3 these are sufficient to derive all the IRR's from the IRR(2).

8 Return to the IGR's and their significance

The IGR's in Table 1 can be expressed in a more organised way in the Table 5. Importantly the LIGR's involved on the left hand side and obtained from the computer calculation, agree with the very long calculation in the preceding section summarised in Table 3.

The following is the table of the distinct RIGR's in Table 5 with the α values put back in and the redundant symbols $\mathtt{T_2}$ representing arbitrary strings removed.

Table 4: The set of RIGR's

$\mathtt{1\underline{a}} \rightarrow \mathtt{2b\_}$	$\mathtt{1\underline{b}} \rightarrow \mathtt{3\_b}$	$\mathtt{1\underline{c}} \rightarrow \mathtt{1b\_}$	$\mathtt{1\underline{c}a} \rightarrow \mathtt{2bb\_}$	$\mathtt{1\underline{c}abab} \rightarrow \mathtt{3b^5\_}$
$\mathtt{1\underline{c}ababa} \rightarrow \mathtt{3\_bababa}$	$\mathtt{1\underline{c}ababc} \rightarrow \mathtt{3bbbbbc\_}$	$\mathtt{1\underline{c}abc} \rightarrow \mathtt{1bbbb\_}$	$\mathtt{1\underline{c}c} \rightarrow \mathtt{1bb\_}$	$\mathtt{2\underline{a}} \rightarrow \mathtt{3b\_}$
$\mathtt{2\underline{b}} \rightarrow \mathtt{2c\_}$	$\mathtt{2b\underline{c}} \rightarrow \mathtt{3\_bc}$	$\mathtt{2bb\underline{c}} \rightarrow \mathtt{1\_abc}$	$\mathtt{2c\underline{c}} \rightarrow \mathtt{1bb\_}$	$\mathtt{3b^5\underline{a}} \rightarrow \mathtt{3\_bababa}$
$\mathtt{3cb\underline{a}} \rightarrow \mathtt{3bbb\_}$	$\mathtt{3\underline{b}} \rightarrow \mathtt{1\_a}$	$\mathtt{3bb\underline{b}} \rightarrow \mathtt{1\_aba}$	$\mathtt{3bbb\underline{b}} \rightarrow \mathtt{3\_baba}$	$\mathtt{3bbbbb\underline{b}} \rightarrow \mathtt{3\_bababa}$
$\mathtt{3cb\underline{b}} \rightarrow \mathtt{3bbb\_}$	$\mathtt{3c\underline{b}} \rightarrow \mathtt{2bb\_}$	$\mathtt{3\underline{c}} \rightarrow \mathtt{3c\_}$	$\mathtt{3\underline{c}ba} \rightarrow \mathtt{3bbb\_}$	$\mathtt{3\underline{c}bab} \rightarrow \mathtt{3\_baba}$

Table 5: The table of IGR's (Table 1) reexpressed in terms of LIGR's and RIGR's

α	LIGR	RIGR
	$1\underline{T_1} \leftarrow 1\underline{c}T_1$	
	$2\underline{T_1} \leftarrow 1\underline{a}T_1$	
	$3\underline{T_1} \leftarrow 2\underline{a}T_1$	
	$1\underline{c}aT_1 \leftarrow 2\underline{a}cdT_1$	$1\_T_2 \rightarrow 3\_bT_2$
b	$1\underline{c}cT_1 \leftarrow 1\underline{a}bcT_1$	$3\_T_2 \rightarrow 1\_aT_2$
	$1\underline{c}aaT_1 \leftarrow 1\underline{a}badT_1$	
	$1\underline{c}abcT_1 \leftarrow 2\underline{a}ccdcT_1$	
	$1\underline{c}ababT_1 \leftarrow 1\underline{a}bbccbT_1$	
	$1\underline{c}ababcT_1 \leftarrow 1\underline{a}bbccacT_1$	
c	$3\underline{T_1} \leftarrow 3\underline{c}T_1$	$1\_aT_2 \rightarrow 2bb\underline{T_2}$
		$3\_babT_2 \rightarrow 3\_babaT_2$
c	$2\underline{T_1} \leftarrow 2\underline{b}T_1$	$1\_cT_2 \rightarrow 1bb\underline{T_2}$
		$3\_baT_2 \rightarrow 3bbb\underline{T_2}$
		$3\_babT_2 \rightarrow 3\_babaT_2$
		$1\_ababaT_2 \rightarrow 3\_bababaT_2$
a	$1\underline{T_1} \leftarrow 3T_1\underline{d}$	$1T_2\_ \rightarrow 2T_2b\_$
		$2T_2\_ \rightarrow 3T_2b\_$
		$3T_2b^5\_ \rightarrow 3\underline{T_2}bababa$
c	$1\underline{T_1} \leftarrow 2T_1\underline{c}$	$1T_2\_ \rightarrow 1T_2b\_$
		$3T_2\_ \rightarrow 3T_2c\_$
		$2T_2b\_ \rightarrow 3\underline{T_2}bc$
		$2T_2c\_ \rightarrow 1T_2bb\_$
		$2T_2bb\_ \rightarrow 1\underline{T_2}abc$
b	$3\underline{T_1} \leftarrow 1T_1\underline{b}$	$2T_2\_ \rightarrow 2T_2c\_$
		$3T_2c\_ \rightarrow 2T_2bb\_$
		$3T_2bb\_ \rightarrow 1\underline{T_2}aba$
		$3T_2cb\_ \rightarrow 3T_2bbb\_$
		$3T_2bbb\_ \rightarrow 3\underline{T_2}baba$
		$3T_2bbbbb\_ \rightarrow 3\underline{T_2}bababa$
c	$1\underline{c}cT_1 \leftarrow 2\underline{b}bcT_1$	$3\_babT_2 \rightarrow 3\_babaT_2$
		$1\_ababT_2 \rightarrow 3b^5\underline{T_2}$
		$1\_ababaT_2 \rightarrow 3\_bababaT_2$
		$1\_ababcT_2 \rightarrow 3bbbbbc\underline{T_2}$
c	$1\underline{c}aT_1 \leftarrow 3\underline{c}cdT_1$	$1\_abcT_2 \rightarrow 1bbbb\underline{T_2}$
		$3\_babT_2 \rightarrow 3\_babaT_2$
		$1\_ababaT_2 \rightarrow 3\_bababaT_2$
	$1\underline{c}aaT_1 \leftarrow 2\underline{b}badT_1$	
c	$1\underline{c}abcT_1 \leftarrow 3\underline{c}ccdcT_1$	$3\_babT_2 \rightarrow 3\_babaT_2$
	$1\underline{c}ababT_1 \leftarrow 2\underline{b}bbccbT_1$	$1\_ababaT_2 \rightarrow 3\_bababaT_2$
	$1\underline{c}ababcT_1 \leftarrow 2\underline{b}bbccacT_1$	
c	$3T_1bb\underline{b} \leftarrow 2T_1adb\underline{c}$	$3T_2\_ \rightarrow 3T_2c\_$
a	$3T_1bb\underline{b} \leftarrow 3T_1adb\underline{d}$	$3T_2cb\_ \rightarrow 3T_2bbb\_$
		$3T_2b^5\_ \rightarrow 3\underline{T_2}bababa$

Notice that the complete Table 5 consisting of 11 blocks has the property that within each block every LIGR combines with every RIGR to form an IGR where the LIGR is $\texttt{CS1} \leftarrow \texttt{CS3}$ and the RIGR is $\texttt{CS2} \rightarrow \texttt{CS4}$.i.e. the IGR is $\texttt{CS1} \rightarrow\rightarrow \texttt{CS2} \Rightarrow \texttt{CS3} \rightarrow\rightarrow \texttt{CS4}$ provided $\texttt{CS1} \rightarrow\rightarrow \texttt{CS2}$ has the type LRL or RLR. The last condition is automatically satisfied because within each block in Table 5 because CS1 and CS2 both have the pointer on the same side (and strings $\texttt{T}_1, \texttt{T}_2$). This property of Table 5 needs to be proved generally.

The result with IGR's 4,5,14,15 omitted (corresponding to 8.1 and 9.1 in block 1, and 3.2 and 4.2 in block 9 of Table 5) occurred when the program was run with $\texttt{n} = \texttt{9}$. In this case the incomplete Table 5 does not have the above property.

Can the IGR's be obtained directly based on the LIGR's without first obtaining the IRR's as was done in the computer program?

8.1 Obtaining information about a TM from its IGR's

Table 6 shows the frequencies of the different types of IRR's obtained from the computer program [5] applied to TM1.

Table 6: The frequencies of the different types of IRR's

length	RLL	RLR	LRL	LRR	total
3	3	3	4	2	12
4	1	4	5	1	11
5	0	4	9	1	14
6	4	3	16	1	24
7	0	0	30	0	30
8	0	0	56	0	56
9	0	0	106	0	106
10	0	0	201	0	201

[Can this be used to obtain a description of the IRR's analogous to paper 1 equation 26? which involves an operator that gives many strings of length 1 more than the input string?]

From this it appears that there are no IRR's of length $\geq \texttt{7}$ of the type RLR, then all such IRR's have type RLL, LRL, or LRR. If this is generally true, and if the TM reaches position $\texttt{6}$ followed by position $\texttt{1}$ it cannot subsequently reach position $\geq \texttt{7}$ because this would require a subsequence of CS's of the form $\texttt{n} \rightarrow \texttt{1} \rightarrow \texttt{n}+\texttt{1}$ with $\texttt{n} = \texttt{6}$ which would be an IRR by lemma 2.1 of type RLR of length $\texttt{n}+\texttt{1}$ contradicting the assumption. The pointer is then constrained to positions $\leq$6, and if it reaches position 0 then because it has reached position 5 previously, the same argument can be applied showing that it cannot then reach position $\geq \texttt{6}$ etc.. This implies that the pointer is contrained to being in a moving window of length 6 that moves left by one space when the pointer moves just to its left. Because of this, if a snapshot is taken of its behaviour whenever the TM reaches just beyond the left hand end of the window, whatever symbol is finds there the result will be at the next snapshot that the symbols of the window have changed depending on the previous symbols there and the new symbol. This is an effective finite state machine with internal state corresponding to the set of symbols in the window and its actual machine state, and it continues indefinitely unless a stationary cycle occurs. That would halt the effective finite state machine.

Each IGR in Table 1 is a logical implication from the existence of one IRR to another, each containing the arbitrary strings $\mathtt{T}_1$ and $\mathtt{T}_2$. For example `IGR55` on the right gives an IRR of type `RLL` if $|\mathtt{T}_2| = \mathtt{0}$. Its length is $|\mathtt{T}_1| + \mathtt{4} = |\mathtt{T}_2| + \mathtt{6}$ but needs completion (i.e. further computation as far as possible) if $|\mathtt{T}_2| > \mathtt{0}$. IGR51 gives an IRR of type RLR and requires $|\mathtt{T}_1| + \mathtt{4} = |\mathtt{T}_2| + \mathtt{3}$ which is its length.

The set of IGR's in Table 1 is divided into two parts because the left hand members either represent an IRR of type LRL or RLR (extendable IRR's). The corresponding right hand members represent IRR's of four types such that each IGR is of one of four types as follows: `LRL` $\Rightarrow$ `LRL`, `LRL` $\Rightarrow$ `LRR`, `RLR` $\Rightarrow$ `RLR`, `RLR` $\Rightarrow$ `RLL`. The relation "could be followed by" is true where the right hand member of the first IGR matches the left hand member of the second one for suitable strings $\mathtt{T}_1$ and $\mathtt{T}_2$. Thus in many cases the matching will be conditional and this relation will include all possble matches between the IGR's so that the absence of such a relation definitely rules out any possible match. The relation "could be followed by" implies one IGR can follow the other logically to generate a new IRR's from a given one. For example IGR 41 can be followed by IGR 50 provided the last symbol of $\mathtt{T}_2$ is `c`. Because of this any such relation needs to be verified for the particular case in question.

In the case that the number of IGR's is finite as in this example TM, the absence of IRR's of type RLR of length `n` for all sufficiently large `n` would follow from the absence of a closed cycle of IGR's of type `RLR` $\Rightarrow$ `RLR` that can be repeated indefinitely. The search for such closed cycles can be done following from the relation on IGR's defined by "could be followed by" that is easily obtained either from Table 1 by matching the state and the symbols in the neighbourhood of the pointer and has been programmed.

This relation can be represented by a directed graph (digraph) where the nodes are IGR's and is also separated into two parts because of the two types of extendable IRR's. The existence of any circuits should be established for each part of the digraph separately.

[The existence of such a circuit implies that (provided it can be iterated indefinitely) the number of IRR's of this type generated from the TM is infinite. This implies the existence of IRR's of the type of arbitrary large length (n). The other case of the absence of a circuit implies their number is finite.]

For the example TM 23, there were no circuits for the digraph of IRR's of type RLR but there were circuits for IRR's of type LRL showing that arbitrarily long IRR's of type LRL exist but not for IRR's of type RLR. The two circuits of length 1 involve respectively IGR's 38 and 39. IGR 38 iterated twice gives $\mathtt{2}\underline{\mathtt{T_1}} \rightarrow\rightarrow \mathtt{3\_babT_2} \overset{\mathtt{c}}{\Rightarrow} \mathtt{2\underline{b}T_1} \rightarrow\rightarrow \mathtt{3\_babaT_2} \overset{\mathtt{c}}{\Rightarrow} \mathtt{2\underline{b}bT_1} \rightarrow\rightarrow \mathtt{3\_babaaT_2}$ and it can be clearly iterated n times giving on the right giving $\mathtt{2\underline{b}b}^{n-1}\mathtt{T_1} \rightarrow\rightarrow \mathtt{3\_baba}^{n}\mathtt{T_2}$. In order to determine $\mathtt{T_1}$ and $\mathtt{T_2}$, the table of IRR's generated from the TM shows that an example of IGR 38 has context pair $(\mathtt{bbcb}, \mathtt{a})$ showing that $\mathtt{T_1} = \mathtt{bbcb}$ and $\mathtt{T_2} = \mathtt{a}$ corresponding to the IRR $\mathtt{2\underline{b}bcb} \rightarrow\rightarrow \mathtt{3\_baba}$. Therefore a result of the iteration of $\mathtt{38}$ is $\mathtt{2\underline{b}b}^{n-1}\mathtt{bbcb} \rightarrow\rightarrow \mathtt{3\_baba}^{n+1}$ for $n \geq \mathtt{0}$. Another example of this is $\mathtt{2\underline{b}bcac} \rightarrow\rightarrow \mathtt{3\_babac}$ giving the result $\mathtt{2\underline{b}b}^{n-1}\mathtt{bbcccb} \rightarrow\rightarrow \mathtt{3\_baba}^{n+2}\mathtt{b}$ for $n \geq \mathtt{0}$. Similarly IGR 39 iterated n times gives $\mathtt{3\underline{c}c}^{n-1}\mathtt{T_1} \rightarrow\rightarrow \mathtt{3\_baba}^{n}\mathtt{T_2}$. An example of a circuit of length 2 is $\mathtt{22} \cdot \mathtt{7}$. IGR 22 is $\mathtt{1}\underline{\mathtt{T_1}} \rightarrow\rightarrow \mathtt{3\_T_2} \overset{\mathtt{22b}}{\Rightarrow} \mathtt{1\underline{c}T_1} \rightarrow\rightarrow \mathtt{1\_aT_2}$. If $\mathtt{T_1}$ begins with $\mathtt{c}$ then $\mathtt{22} \cdot \mathtt{7}$ can be written as $\mathtt{1\underline{c}T_1} \rightarrow\rightarrow \mathtt{3\_T_2} \overset{\mathtt{22b}}{\Rightarrow} \mathtt{1\underline{c}cT_1} \rightarrow\rightarrow \mathtt{1\_aT_2} \overset{\mathtt{7b}}{\Rightarrow} \mathtt{1\underline{a}bcT_1} \rightarrow\rightarrow \mathtt{3\_baT_2}$. However this clearly does not iterate because $\mathtt{1\underline{a}bc}$ does not match $\mathtt{1\underline{c}}$. This also happens with $\mathtt{1} \cdot \mathtt{26}$. An example of a circuit of length 2 that does iterate is $\mathtt{1} \cdot \mathtt{22}$ which gives $\mathtt{1}\underline{\mathtt{T_1}} \rightarrow\rightarrow \mathtt{1\_T_2} \overset{\mathtt{1b}}{\Rightarrow} \mathtt{1\underline{c}T_1} \rightarrow\rightarrow \mathtt{3\_bT_2} \overset{\mathtt{22}}{\Rightarrow} \mathtt{1\underline{c}cT_1} \rightarrow\rightarrow \mathtt{1\_abT_2}$. This iterates to get $\mathtt{1\underline{c}}^{2n}\mathtt{T_1} \rightarrow\rightarrow \mathtt{1\_(ab)}^{n}\mathtt{T_2}$. An example of the first IRR is $\mathtt{1\underline{c}b} \rightarrow\rightarrow \mathtt{1\_ab}$. In this case $\mathtt{T_1} = \mathtt{cb}$ and $\mathtt{T_2} = \mathtt{ab}$ therefore $\mathtt{1\underline{c}}^{2n}\mathtt{cb} \rightarrow\rightarrow \mathtt{1\_(ab)}^{n+1}$.

Because there are no circuits in the digraph of RLR type IGR's, all the IRR's of type RLR are obtained by starting with the RLR IRR's of length 2 and applying all the IGR's of type $\mathtt{RLR} \Rightarrow \mathtt{RLR}$ until this can be done no more. These IGR's are just those numbered $\mathtt{40}-\mathtt{44}, \mathtt{46}, \mathtt{47}, \mathtt{50}, \mathtt{51}$. Of these IGR's $\mathtt{43}, \mathtt{44}, \mathtt{46}$ do not match any others. The IGR's of type RLR generating the IRR(2) are just $\mathtt{40}$ with context $(\mathtt{c}, \mathtt{b})$, $\mathtt{41}$ with context $(\mathtt{a}, \mathtt{b})$ and $\mathtt{47}$ with context $(\mathtt{c},)$. Carrying this out gives the results in the diagram following where the counts for the numbers of distinct IRR's for each length are given in the top two rows. The provenence of the IRR's is indicated below. The numbers in square brackets are the count for the node, the other numbers in the nodes are the latest IGR used followed by the α value as given above $\Rightarrow$ in the IGR's and consecutive numbering of the new origins in the IGR's in the order they are given in Table 1. Arrows indicate the order of application of the IGR's, x indicates the end of the sequence of IGR's, or a failure of the contexts to match. Also there are some conditions that must be satisfied written below the corresponding arrow. The resulting frequencies match the IRR results of the computer program (but the numbers differ because of the way IRR's are counted) and therefore demonstrate all the results for the TM including the moving window behaviour.

n	2	3	4	5	6
#IRR	6	4	6	10	6

$$\begin{tikzcd}
 & & {[2]\mathtt{41a1}} \arrow[r] & {[2]\mathtt{50}} \arrow[r] & {[2]\mathtt{43x}} \\
 & & {[2]\mathtt{41a2}} \arrow[r, "\mathtt{d=b}"'] \arrow[rd] \arrow[rdd, "\mathtt{d=b}"'] & {[2]\mathtt{44}\left\{\begin{array}{l}\mathtt{c1x}\\ \mathtt{c2x}\end{array}\right.} & \\
{[3](\mathtt{c},\mathtt{b})\mathtt{40}\left\{\begin{array}{l}\mathtt{a1},\mathtt{a2}\\ \mathtt{c1x}\end{array}\right.} \arrow[r] & {[2]\mathtt{42}} \arrow[ruu] \arrow[ru] \arrow[rd] & & {[2]\mathtt{50}} \arrow[r] & {[2]\mathtt{43x}} \\
 & & {[2]\mathtt{46x}} & {[4]\mathtt{51}\left\{\begin{array}{l}\mathtt{a1x}\\ \mathtt{a2x}\\ \mathtt{a3x}\\ \mathtt{a4}\end{array}\right.} \arrow[r] & {[2]\mathtt{44}}
\end{tikzcd}$$

$$\begin{tikzcd}
{[2](\mathtt{a},\mathtt{b})\mathtt{41}\left\{\begin{array}{l}\mathtt{a1}\\ \mathtt{a2}\end{array}\right.} \arrow[r, "\mathtt{x}"] \arrow[rd, "\mathtt{x}"] & {\mathtt{44}\left\{\begin{array}{l}\mathtt{c1}\\ \mathtt{c2}\end{array}\right.} & \\
 & \mathtt{50} \arrow[r] & \mathtt{43}
\end{tikzcd}$$

$$\begin{tikzcd}
 & {[1]\mathtt{41a1}} \arrow[r, "\mathtt{x}"] & \mathtt{50} \arrow[r] & \mathtt{43} \\
{[1](\mathtt{c},)\mathtt{47}} \arrow[ru] \arrow[r] & {[1]\mathtt{41a2}} \arrow[ru, "\mathtt{x}"] \arrow[r, "\mathtt{x}"] \arrow[rd, "\mathtt{x}"] & \mathtt{44} & \\
 & & \mathtt{51} &
\end{tikzcd}$$

$\mathtt{40} \to (\mathtt{b},)\mathtt{42} \to (\mathtt{bb},\mathtt{c})\mathtt{41} \to \mathtt{51}$ These specialisations remove the conditions that otherwise would apply to the connections between the IGR's. Can this be done systematically to all the IGR's?

$\mathtt{1}\underline{\mathtt{T}_\mathtt{1}}\mathtt{bb} \to\to \mathtt{2T}_\mathtt{2}\mathtt{c}\_,\mathtt{3T}_\mathtt{1}\underline{\mathtt{b}} \to\to \mathtt{2T}_{\mathtt{2}\_}$

Show this gives rise to finite state machine behaviour and relate it to $n_\mathtt{L}$ and $n_\mathtt{R}$ in my first TM paper.

Combining RLR iterations with LRL iterations for a TM with cycles in both cases. This would give rise to possibly repeating cycles that might be describable as a simulated TM.

References

[1] Methods for Understanding Turing Machine Computations

[2] John Nixon Reverse engineering Turing Machines and the Collatz Conjecture

[3] The previous version in D of the computer program for analysis of Turing Machines

[4] Program for just doing the backward search for a single CS

[5] The new program tie v3.0 for doing the computations in this paper