A STUDY OF A MODEL FOR PARALLEL COMPUTATION

by

Raymond Reiter

July, 1965

This research was supported by Air Force Contract AF 30(602)-3546.
Abstract

This paper deals with a generalization of a model for parallel computation as formulated by Karp and Miller. A parallel computation is viewed as a directed graph in which a node represents a sequence of operations to be performed upon data on the node input branches with the results of an operation being placed upon the output branches. An operation associated with a node \( n \) may initiate if and only if, for each input branch \( d_p \) to \( n \), the data queue length on \( d_p \) is greater than, or equal to, some threshold \( T_p \). It is shown that a unique computation is defined independently of the timing of the node operations. Methods are derived for determining whether a computation terminates and for finding the number of performances of each node computation step involved. In particular, an algorithm is given which yields the number of performances of each node computation step for the set of terminating nodes of a computation graph. Finally, necessary and sufficient conditions for data queue lengths to remain bounded are derived.
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>The Model</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>Determinacy of Computation Graphs</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>Termination Properties of Executions</td>
<td>17</td>
</tr>
<tr>
<td>5</td>
<td>Termination Properties of Proper Executions</td>
<td>18</td>
</tr>
<tr>
<td>6</td>
<td>Conditions for Self-Termination</td>
<td>22</td>
</tr>
<tr>
<td>7</td>
<td>Properties of Queue Lengths</td>
<td>31</td>
</tr>
<tr>
<td>8</td>
<td>Conclusion</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>References</td>
<td>34</td>
</tr>
</tbody>
</table>
1. Introduction

This paper generalizes a model for parallel computation formulated by Karp and Miller [1]. Briefly, this model represents a computation as a directed graph in which a node $n_i$ represents an operation to be performed upon data which lie upon the input branches to $n_i$. The results of the operation by node $n_i$ are placed upon the output branches of $n_i$. A branch, therefore, represents a data queue. With each branch $d_p$ directed from node $n_i$ to node $n_j$ is associated a quadruple $(A_p, U_p, W_p, T_p)$, the elements of which are interpreted as follows:

- $A_p$ - the initial number of data items on branch $d_p$
- $U_p$ - the number of data items placed on $d_p$ upon the termination of the operation associated with $n_i$.
- $W_p$ - the number of data items removed from $d_p$, under a first-in first-out queue discipline, upon the initiation of the operation associated with node $n_j$.
- $T_p$ - a threshold. In order to initiate the operation associated with node $n_j$, the queue length on $d_p$ must be greater than, or equal to $T_p$.

It is shown in [1] that such a computation graph represents a unique computation, determined independently of operation times. Methods of determining termination properties and the number of performances of each computation step are developed. Finally, necessary and sufficient conditions for data queue lengths to remain bounded are derived.

In the present paper, a model for parallel computation based upon that in [1] is given. This model is a generalization in that the operation associated with a given node need not be fixed, but may vary throughout the
computation, provided that it is uniquely specified as a function of the number of previous performances of the given node during the course of the computation. For this general model, determinacy is established in the sense that a unique computation is determined independently of operation times. The model is then specialized to be data dependent upon the lengths of data queues as follows: with each branch $d_p$, directed from node $n_i$ to node $n_j$, is associated a quadruple $(A_p, \{U_p(n)\}, \{W_p(n)\}, T_p)$ where $A_p$ and $T_p$ have the previous interpretation. The sequence $\{U_p(n)\}$ specifies the number of data words, placed on $d_p$ upon the termination of the $n$-th operation of node $n_i$. $\{W_p(n)\}$ specifies the number of data words, under a first-in first-out queue discipline, removed from $d_p$ upon the initiation of the $n$-th operation of node $n_j$.

With this model we address ourselves to finding methods of determining whether such a computation terminates and to finding the number of performances of each node computation step involved. In particular, an algorithm is given which yields the number of performances of each node computation step for the set of terminating nodes of a computation graph. It is found, as was the case in [1], that the maximal strongly connected subgraphs of a computation graph, and their loops, play a central role in the analysis. Finally, it is assumed that the sequences $\{U_p(n)\}$ and $\{W_p(n)\}$ are periodic. Under this assumption, methods of determining termination properties and the number of performances of each node computation step are given, and necessary and sufficient conditions for data queue lengths to remain bounded are derived.
In the development it will be assumed that the reader is familiar with the definitions and principal results of [1]. The proofs of all results which follow in an obvious fashion from the proofs of analogous results in [1] will be omitted.
2. The Model

Definition 1

A computation graph is a directed graph consisting of:

(i) nodes $n_1, n_2, \ldots, n_i$.

(ii) branches $d_1, d_2, \ldots, d_t$, where any given branch $d_p = (n_i, n_j)$ is directed from a specified node $n_i$ to a specified node $n_j$.

(iii) a sextuple $(A_p, H_p, \alpha_p, h_p, t_p, \pi_p)$ associated with each branch $d_p = (n_i, n_j)$ where:

(a) $A_p$ is a given set, the alphabet.

(b) $H_p \subseteq \pi_p$ where $\pi_p$ is the free semigroup on $A_p$, with identity $\emptyset$. If $\sigma \in H_p$, $\tau \in \pi_p$, then $\sigma \tau \in H_p$.

(c) $a_p \in \pi_p$.

(d) $h_p$ is a mapping: $\pi_p \rightarrow \pi_p$ such that for $\sigma \in H_p$, $\tau \in \pi_p$, $h_p(\sigma \tau) = h_p(\sigma)$ and for $\sigma \in H_p$, $h_p(\sigma) = \emptyset$.

(e) $t_p$ is a mapping: $\pi_p \rightarrow \pi_p$ such that for $\sigma \in H_p$, $\tau \in \pi_p$, $t_p(\sigma \tau) = t_p(\sigma) \tau$ and for $\sigma \in H_p$, $t_p(\sigma) = \sigma$.

(f) $\alpha_p$ is a mapping: $\pi_p \times \pi_p \times \pi_p \rightarrow \pi_p$ where $n_i$ has input branches $d_{p_1}, d_{p_2}, \ldots, d_{p_r}$, and for $\sigma(p_1) \in \pi_p$, $i=1, 2, \ldots, r$; $\alpha_p: (h_{p_1}(\sigma(p_1)), \ldots, h_{p_r}(\sigma(p_r))) \rightarrow \sigma(p) \in \pi_p$.

The sextuple associated with each branch may be interpreted as follows:

$A_p$ is the set of all data words upon which $O_j$, the operation associated with node $j$, can operate. $\pi_p$ represents the set of all sequences of data words which can appear as a queue on the branch $d_p$. $H_p$ is interpreted as the class of all sequences acceptable to $O_j$; i.e. $O_j$ cannot initiate unless the
data queue on branch \( d_p \) is an element of \( H_p \), so that \( H_p \) defines a threshold for the initiation of \( O_j \). The data string \( a_p \) represents the initial data queue on the branch \( d_p \). The function \( h_p \) defines, for a given data queue, the minimal acceptable head which permits the initiation of \( O_j \), and the function \( t_p \) defines the data queue remaining after \( O_j \) removes data from the head of the queue upon its initiation. Finally the function \( \alpha_p \) defines the data sequence which is placed on the tail of the branch \( d_p = (n_i, n_j) \) upon termination of the operation \( O_j \).

We shall require that the operation \( O_j \) associated with a given node \( n_j \) be eligible for initiation if and only if for each branch \( d_p \) directed into \( n_j \), the data sequence on \( d_p \) is acceptable, i.e. is an element of \( H_p \), and that no previously initiated performance of \( O_j \) is still in progress. After \( O_j \) becomes eligible for initiation, there may be some delay before initiation actually occurs. For each branch \( d_p \) directed into \( n_j \), \( O_j \), upon initiation, removes an initial sequence of data words on the queue associated with \( d_p \). There may then be a further delay before \( O_j \) terminates, at which time for each branch \( d_q \) directed out of \( n_j \), a sequence of data words determined by the function \( \alpha_q \) is placed on \( d_q \). The delays mentioned above are left unspecified by the model and, in fact, may differ for different initiations and performances of the same operation.

The above constraints on initiation lead to the following definitions of the possible sequences of initiations associated with a given computation graph \( G \).

Let \( \mathcal{S} \) be a sequence of nonempty sets \( S_1, S_2, \ldots, S_n, \ldots \), such that each \( S_n \subseteq \{1, 2, \ldots, \lambda, 1, 2, \ldots, \tilde{l}\} \) where \( \lambda \) is the number of nodes in \( G \). Let
\(x(j,0) = 0\) for \(j \in \{1,2,\ldots,l\}\) and for \(N > 0\) let \(x(j,N)\) denote the number of sets \(S_m\), \(1 \leq m \leq N\) of which \(j\) is an element. A similar definition holds for \(x(\bar{j},N)\).

Let \(d_p = (n_i,n_j)\) and let \(n_i\) have input branches \(d_{p_1}, d_{p_2}, \ldots, d_{p_r}\). We define a sequence \(\sigma(p,N) \in_p \) for \(N=0,1,\ldots,\) as follows: \(\sigma(p,0) = a_p\). For \(N \geq 1\), assume that for all \(p\), \(\sigma(p,0), \sigma(p,1), \ldots, \sigma(p,N-1)\) have been defined and let \(N' = \max \{T : T \leq N \text{ and } i \in S_T\}\). Then if \(i \in S_N\) and \(N'\) exists

\[\alpha(p,N-1) = \alpha_p\left[\sum_{i=1}^{p_1} \sigma(p_1,N'-1), \ldots, \sum_{i=p_1}^{p_r} \sigma(p_r,N'-1)\right].\]

Otherwise, \(\alpha(p,N-1) = \emptyset\). Then, if \(i \in S_N\),

\[\sigma(p,N) = \sum_{i=1}^{p_1} \sigma(p,N-1)\alpha(p,N-1)\]

\(j \notin S_N\), \(\sigma(p,N) = \sigma(p,N-1)\alpha(p,N-1)\).

**Definition 2**

The sequence \(\mathcal{E} = S_1, S_2, \ldots\) is an execution of \(G\) if and only if, for all \(N \geq 0\), the following conditions hold.

(i) \(x(j,N) - x(\bar{j},N) \in \{0,1\}\).

(ii) If \(j \in S_N\), there exists \(R \geq N\) such that \(\bar{j} \in S_R\).

(iii) If \(j \in S_{N+1}\) and \(G\) has a branch \(d_p\) from \(n_i\) to \(n_j\), then \(\sigma(p,N) \in H_p\).

(iv) If \(\mathcal{E}\) is finite and of length \(P\), then for each \(j\), there exists a node \(n_i\) and a branch \(d_p\) from \(n_i\) to \(n_j\) such that \(\sigma(p,P) \notin H_p\).

**Definition 3**

An execution \(\mathcal{E}\) of \(G\) is called a proper execution if the following implication holds.

(v) If \(\sigma(p,N) \in H_p\) for every branch \(d_p = (n_i,n_j)\) directed into \(n_j\), then \(j \in S_P\) for some \(P > N\).
The sequence $\mathcal{C}$ is interpreted as giving a temporal sequence of initiations of operations throughout a performance of the parallel computation specified by $G$. The occurrence of $S_N$ denotes the simultaneous initiation of $O_j$ for all $j \in S_N$ and the simultaneous termination of $O_j$ for all $\tilde{j} \in S_N$. The recursive definition of $\sigma(p,N)$ is given with this interpretation in mind so that $\sigma(p,N)$ is viewed as the sequence of data words occurring on branch $d_p$ immediately after all initiations and terminations associated with $S_N$ have been effected. Thus, for example, if $\tilde{i} \in S_N$ and $j \in S_N'$, the sequence $\alpha(p,N-1)$ which is the result of the previous initiation of $O_{\tilde{i}}$, is placed upon the queue $\sigma(p,N-1)$, yielding the data sequence $\sigma(p,N-1)\alpha(p,N-1)$. The result of initiating $O_j$ then leaves on $d_p$ the sequence $t_p(\sigma(p,N-1)\alpha(p,N-1))$ which is $\sigma(p,N)$. Conditions (i) and (ii) of Definition 2 insure that to each initiation of $O_j$ there corresponds a termination and that no termination may occur which is not preceded by an initiation. Condition (iii) states that a node of $G$ may initiate only if it is eligible for initiation. Condition (iv) gives the only situation under which an execution may terminate. Condition (v) of Definition 3 states that any node of $G$ which is eligible for initiation must eventually initiate. Finally, it should be noted that the simultaneous initiation and termination of any operation $O_j$ is permissible.
3. **Determinacy of Computation Graphs**

In this section we prove that, for proper executions of a computation graph with given initial data, the sequence of data words occurring on any branch is unique. Thus a computation graph represents a unique computation regardless of which proper execution occurs. We refer to this property as the determinacy of computation graphs. This result is embodied in Theorem 1, for the proof of which we require the following:

**Lemma**

Let $E = S_1, S_2, \ldots, S_N, \ldots$, be an execution of a computation graph and let $x(j,N) = n$. Then $\sigma(p,N) = t_p^{(n)}[a_p \alpha(p,0) \alpha(p,1), \ldots \alpha(p,N-1)]$.

**PROOF:** We shall use induction on $N$. For $N=1$ we must have $x(j,1) \in \{0,1\}$. If $x(j,1) = 0$, $j \not\in S_1$, so that $\sigma(p,1) = a_p \alpha(p,0) = t_p^{(0)}[a_p \alpha(p,0)]$. If $x(j,1) = 1$, $j \in S_1$ so that $\sigma(p,1) = t_p[a_p \alpha(p,0)]$. In either case, the lemma holds.

Assume the truth of the lemma for $N=P$. Then if $x(j,P) = n$,

$$\sigma(p,P) = t_p^{(n)}[a_p \alpha(p,0) \ldots \alpha(p,P-1)]$$

Suppose $j \not\in S_{P+1}$. Then

$$\sigma(p,P+1) = \sigma(p,P) \alpha(p,P), \text{ i.e. } t_p^{(n)}[a_p \alpha(p,0) \ldots \alpha(p,P-1)] \alpha(p,P).$$

If $n=0$, the lemma follows immediately. If $n \geq 1$, $t_p^{(n)}[a_p \alpha(p,0) \ldots \alpha(p,P-1)] = t_p[t_p^{(n-l)}(a_p \alpha(p,0) \ldots \alpha(p,P-1))]$ and by (iii) of Definition 2 we must have $t_p^{(n-l)}(a_p \alpha(p,0) \ldots \alpha(p,P-1)) \in H_p$. Hence

$$t_p^{(n-l)}(a_p \alpha(p,0) \ldots \alpha(p,P-1)) \alpha(p,P) = t_p^{(n-l)}(a_p \alpha(p,0) \ldots \alpha(p,P-1)) \alpha(p,P).$$

By iterating this argument we finally obtain

$$\sigma(p,P+1) = t_p^{(n)}[a_p \alpha(p,0) \ldots \alpha(p,P)]$$

so that for $j \not\in S_{P+1}$ the lemma holds.
If $j \in S_{p+1}$, \( \sigma(p,P+1) = t_p[\sigma(p,P)\alpha(p,P)] \)

\[
= t_p[t_p^{(n)}[a_p \alpha(p,0) \ldots \alpha(p,P-1)]\alpha(p,P)]
\]

\[
= t_p[t_p^{(n)}[a_p \alpha(p,0) \ldots \alpha(p,P)]]
\]

by the above argument, so that \( \sigma(p,P+1) = t_p^{(n+1)}[a_p \alpha(p,0) \ldots \alpha(p,P)] \)

and the induction is completed.

For the proof of Theorem 1 we shall require some notation. If \( \sigma \)
has the form \( \sigma = \theta \tau \) with \( \theta, \tau \in \mathbb{N}_p \), we write \( \theta \subseteq \sigma \). Let \( \mathcal{E} \) be an execution.

We write \( N(j,s) = \min \{ N \mid x(j,N) = s \} \).

In what follows, primed symbols will indicate that these are symbols
defined with respect to the proper execution \( \mathcal{E}' \) (e.g.
\( \sigma'(p,N'), x'(j,N') \)) while unprimed symbols either relate to the proper
execution \( \mathcal{E} \) (e.g. \( \sigma(p,N), x(j,N) \)) or are independent of the executions
considered, (e.g. \( h_p, t_p, a_p \)).

**Theorem 1**

Let \( \mathcal{E} = S_1, S_2, \ldots, S_N, \ldots \), and \( \mathcal{E}' = S_1', S_2', \ldots, S_N', \ldots \), be two
proper executions of a computation graph \( G \). If \( x(j,N) = s \) then
there exists \( N' \) such that \( x'(j,N') = s \). Moreover, if \( j \in S_N \),
then \( N' \) can be chosen such that \( x'(j,N'-1) = s-1 \) and

\[
h_p(\sigma(p,N-1)) = h_p(\sigma'(p,N'-1)).
\]

**PROOF:** We apply induction to \( N(j,s) \) with respect to the proper execution
\( \mathcal{E} \). Suppose \( N(j,s) = 1 \). Then \( s \in \{0,1\} \). If \( s=0 \), i.e. \( x(j,1) = 0 \), then,
since \( x'(j,0) = 0 \), we take \( N' = 0 \) and the theorem holds in this case. If
\( s=1 \), then \( j \in S_1 \) which implies, by (iii) of Definition 2, \( a_p \in h_p \). This implies,
by (v) of Definition 3, that there exists \( N' \) such that \( x'(j,N') = 1 \) with
\( j \in S'_{N} \). By the above lemma we have \( \sigma'(j, N'-1) = a_{P} \alpha'(p, o) \ldots \alpha'(p, N'-2) \)
and since \( a_{P} \in H_{P} \), we have \( h_{P}(\sigma(p, o)) = h_{P}(a_{P}) = h_{P}(\sigma'(j, N'-1)) \) and again
the theorem holds.

Now assume the theorem holds for \( N(j, s) \leq P \) and suppose \( N(j, s) = P+1 \).
Clearly, we must have \( s \geq 1 \). Then by the lemma, \( \sigma(p, P) = t_{P}^{(s-1)}[a_{P} \alpha(p, o) \ldots \alpha(p, P-1)] \) and since \( j \in S'_{P+1} \), by (iii) of Definition 2, we must have
\( \sigma(p, P) \in H_{P} \). Now \( N(j, s-1) \leq P \) so that, by the induction hypothesis, there
exists \( M' \) such that \( x'(j, M') = s-1 \) with \( j \in S'_{M'} \). We shall prove that there
exists \( N' \) such that \( x'(j, N') = s \). Suppose, on the contrary, that \( N' \) does
not exist. Then, for all \( R' \geq M' \), we must have \( \sigma'(p, R') = t_{P}^{(s-1)}[a_{P} \alpha(p, o) \ldots \alpha(p, R'-1)] \). Let \( x(i, P) = t \). Then \( N(i, t) \leq P \), so that by the induction
hypothesis and the definition of \( \alpha(p, \cdot) \), to each nonnull \( \alpha(p, T) \), \( T \leq P-1 \),
there corresponds an \( \alpha'(p, T') \), such that \( \alpha(p, T) = \alpha'(p, T') \). Finally, for
\( M_{1}, M_{2} \leq P-1 \), suppose that \( \alpha(p, M_{1}) \) and \( \alpha(p, M_{2}) \) are the results of two con-
secutive occurrences of \( i \in S_{T_{1}} \) and \( i \in S_{T_{2}} \), i.e. \( T_{1} < T_{2} \). Then by (i) of
Definition 2, we must have \( M_{1} < M_{2} \). By the induction hypothesis there exist
\( T_{1}' \) and \( T_{2}' \) with \( T_{1} < T_{2}' \) with \( i \in S_{T_{1}'} \) and \( i \in S_{T_{2}'} \). By (i) and (ii) of Definition
2 there exist \( M_{1}' \) and \( M_{2}' \) such that \( i \in S'_{M_{1}'+1} \) and \( i \in S'_{M_{2}'+1} \) and \( M_{1}' < M_{2}' \). Hence,
to each nonnull sequence \( \alpha(p, M) \) of \( \sigma(p, P) \) there corresponds a sequence
\( \alpha'(p, M) = \alpha(p, M) \) of \( \sigma'(p, R') \) and their ordering is preserved. It follows
that, for sufficiently large \( R' \), \( \sigma(p, P) \subseteq \sigma'(p, R') \). But \( \sigma(p, P) \in H_{P} \) so that,
by (v) of Definition 3, there exists \( N' \) such that \( x'(j, N') = s \) and \( j \in S'_{N'} \).
Clearly, \( x'(j, N'-1) = s-1 \). Finally, if \( x(i, P) \leq x'(i, N'-1) \), then
\( \sigma(p, P) \subseteq \sigma'(p, N'-1) \) while if \( x(i, P) > x'(i, N'-1) \), then \( \sigma'(p, N'-1) \subseteq \sigma(p, P) \).
In either case, since both \( \sigma(p, P) \) and \( \sigma'(p, N'-1) \) are in \( H_{P} \), it follows that
\( h_{P}(\sigma(p, P)) = h_{P}(\sigma'(p, N'-1)) \). This completes the induction.
Theorem 1 establishes that the number of performances of an operation \( O_j \) is the same for all proper executions of a given computation graph \( G \). Moreover, the \( n \)-th performance of \( O_j \) under two proper executions operates upon the same data in both. It follows that the results of a proper execution of \( G \) are determined solely by the initial data regardless of the timing of the operations, i.e. regardless of the proper execution considered. One more point should be noted. The proof of Theorem 1 and its preceding lemma requires only that the set \( H_p \) be the same at the \( n \)-th initiation of \( O_j \) for all proper executions, and that each of the functions \( t_p, \alpha_p \), and \( h_p \) be single valued and perform the same mapping for all proper executions at the \( n \)-th initiation of \( O_j \). Therefore consider modifying the definition of a computation graph by replacing the single "threshold set" \( H_p \) by a collection \( \{H_p(n)\} \) such that the threshold associated with \( d_p = (n_i, n_j) \) is \( H_p(n) \) upon the \( n \)-th initiation of \( O_j \). Similarly, replace the functions \( t_p(\cdot) \) and \( h_p(\cdot) \) by functions \( t_p(\cdot, n) \) and \( h_p(\cdot, n) \) respectively, and modify the recursive definition of \( \sigma(p, N) \) in the obvious way. Then by the above remark, Theorem 1 holds for this more general definition of a computation graph and its associated proper executions.

In the following development we shall investigate a particular realization of this more general form of a computation graph. More precisely, we shall deal with a model which is specified by the following choice of \( H_p(n) \) and \( t_p(\cdot, n) \): \( H_p(n) = H_p = \{\sigma_p \mid \sigma_p \in \sigma_p \text{ and } \iota(\sigma_p) \geq T_p\} \). Here \( \iota(\sigma_p) \) is the length of the sequence \( \sigma_p \) with the convention \( \iota(\emptyset) = 0 \), and \( T_p \) is a non-negative integer. Let \( \{W_p(n) \mid n=1,2,\ldots\} \) be a sequence of nonnegative integers with \( W_p(n) \leq T_p \). If \( \sigma \in H_p \), we have \( \sigma = a_1 a_2 \ldots a_{W_p(n)} \ldots a_{T_p} \ldots a_r \).
where \( s_i \in Q_p, i=1,2,\ldots,r \). Define \( t_p(\sigma_p,n) = a_{W_p(n)} \ldots \sigma_r \). For \( \sigma \in H_p \), \( t_p(\sigma_p,n) = \sigma_p \). Let \( \xi \) be an execution for the above form of a computation graph. A simple induction yields:

\[
i[\sigma(p,N)] = i(a_p) + \sum_{R=0}^{N-1} i[\alpha(p,R)] - \sum_{n=1}^{x(j,N)} W_p(n) - \sum_{n=1}^{x(j,N)} W_p(n).
\]

But \( \alpha(p,R) = \emptyset \) except possibly when \( \bar{i} \in S_{R+1} \). For \( \bar{i} \in S_{R+1} \) define

\[
U_p(x(\bar{i},R+1)) = i[\alpha(p,R)].
\]

Then

\[
i[\sigma(p,N)] = A_p + \sum_{n=1}^{x(\bar{i},N)} U_p(n) - \sum_{n=1}^{x(j,N)} W_p(n) \quad \text{where} \quad A_p = i(a_p).
\]

It follows that \( \sigma(p,N) \in H_p \) if and only if \( i[\sigma(p,N)] \geq T_p \). Thus for \( d_p = (n_i,n_j), T_p, A_p, U_p(n) \) and \( W_p(n) \) may be interpreted, respectively, as a threshold which prevents \( O_j \) from initiating whenever the threshold exceeds the queue length on \( d_p \), the initial data length on branch \( d_p \), the length of data placed on \( d_p \) upon the \( n \)-th termination of \( O_i \), and the length of data removed from \( d_p \) upon the \( n \)-th initiation of \( O_j \).

Finally, we note, by (i) of Definition 2, that \( x(i,N) - x(\bar{i},N) \in \{0,1\} \), so that \( \sigma(p,N) \in H_p \) implies that

\[
A_p + \sum_{n=1}^{x(i,N)} U_p(n) - \sum_{n=1}^{x(j,N)} W_p(n) \geq T_p
\]

while \( \xi \) finite implies \( x(i,N) = x(\bar{i},N) \) by (i) and (ii) of Definition 2.

We may then reformulate Definition 2 and 3 of an execution and proper execution to conform to the present model as follows:
Definition 2'

The sequence $E = S_1, S_2, \ldots$, is an execution of $G$ if and only if, for all $N \geq 0$ the following conditions hold.

(i) $x(j,N) - x(j,N) \in \{0,1\}$.

(ii) If $j \in S_{R}$, there exists $R \geq N$ such that $j \in S_{R}$.

(iii) If $j \in S_{N+1}$ and $G$ has a branch $d_p$ from $n_i$ to $n_j$, then

$$A_p + \sum_{l} U_p(n) - \sum_{l} W_p(n) \geq T_p.$$

(iv) If $E$ is finite and of length $P$, then for each $j$ there exists a node $n_i$ and a branch $d_p$ from $n_i$ to $n_j$, such that

$$A_p + \sum_{l} U_p(n) - \sum_{l} W_p(n) < T_p.$$

Definition 3'

An execution $E$ of $G$ is called a proper execution if the following implication holds.

(v) If

$$A_p + \sum_{l} U_p(n) - \sum_{l} W_p(n) \geq T_p$$

for every branch $d_p = (n_i,n_j)$ directed into $n_j$, then $j \in S_{P}$ for some $P > N$. 
An Exemple

Consider matrix multiplication \( C = AB \) where

\[
    c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}
\]

\( i=1, \ldots, m; \ j=1, \ldots, p. \)

Figure 1 shows a computation graph for this calculation where each branch \( d_p \) is labeled with its associated quadruple \((A_p, \{U_p(n)\}, \{W_p(n)\}, T_p)\).

Each of the indices \( i, j, \) and \( k \) is assumed to have zero as its initial value. Index modification is performed by the operations \( O_1, O_2, \ldots, O_7 \) as follows:

- \( O_1 \) replaces \( k \) by \((k+1) \mod n\)
- \( O_2 \) replaces \( j \) by \((j+1) \mod p\)
- \( O_3 \) replaces \( i \) by \( i+1 \).

The initial data on the branches \((n_1, n_1), (n_2, n_2)\) and \((n_3, n_3)\) consist of markers sequencing the indices through their appropriate values. Termination of the queue associated with branch \((n_3, n_3)\) is caused by the depletion of the queue associated with branch \((n_3, n_3)\). The values \( U = np \) and \( W=1 \) on the branch \((n_3, n_4)\) specify that any given value of the index \( i \) is used in \( np \) consecutive performances of \( O_4 \). Similarly, for branch \((n_2, n_5)\).

- \( O_4 \) fetches \( a_{ik} \) from storage
- \( O_5 \) fetches \( b_{kj} \) from storage
- \( O_6 \) forms the product \( a_{ik} b_{kj} \)
- \( O_7 \) adds \( a_{ik} b_{kj} \) to the partial sum in a sum bucket and puts out \( c_{ij} \) on \((n_7, n_7)\) only when it has performed \( n \) summations.
At the end of the calculation \((n_\gamma, n_\bar{\gamma})\) contains the sequence
\[
c_{11}, c_{12}, \ldots, c_{1p}, c_{21}, \ldots, c_{2p}, \ldots, c_{m1}, \ldots, c_{mp}.
\]
Figure 1. A computation graph for matrix multiplication.
4. **Termination Properties of Executions**

We shall assume throughout that $U_p(n) \neq 0$ and $W_p(n) \neq 0$ for all $n$. The case $U_p(n) = W_p(n) = 0$ for all $n$ is treated in [1]. We write

$$\sigma_p = A_p - T_p + 1, \text{ and } \Sigma_j = \{(i,P) \mid d_p \text{ is a branch from node } n_i \text{ to node } n_j\}.$$ 

**Theorem 3**

Let $G$ be a computation graph. Consider the system of inequalities

$$\min_{(i,P) \in \Sigma_j} \left[ \sigma_p + \sum_{l} U_p(n) - \sum_{l} W_p(n) \right] \leq 0 \quad \text{for } j=1,2,\ldots,t \quad (1)$$

If $\Sigma_j$ is empty for any $j$ or if (1) has no solution in non-negative integers then every execution $\mathcal{E}$ of $G$ is of infinite length.

If (1) has a nonnegative integer solution, then every execution $\mathcal{E}$ of $G$ is of finite length, and, for any $j$, the number of performances of $O_j$ is $x^*(j)$, the minimum integer solution of (1).

From this point on we assume that $\sum W_p(n) = \infty$.

**Lemma 1**

If, for some $(i,P) \in \Sigma_j$, $\sum U_p(n) < \infty$, $n_j$ terminates. (Note in this case $n_i$ need not terminate.) If, for some $(i,P) \in \Sigma_j$, $n_i$ terminates, then $n_j$ terminates.

**Lemma 2**

Let $G'$ be a strongly connected subgraph of $G$. Then in any execution $\mathcal{E}$ of $G$ either every node of $G'$ terminates or none does.
5. **Termination Properties of Proper Executions**

We consider, in what follows, the case $\Sigma U_p(n) = \infty$, as well as $\Sigma W_p(n) = \infty$.

**Lemma 3**

The node $n_j \in G$ terminates if and only if, for some $(i,p) \in \Sigma_j$, $n_i$ terminates. Also, if $n_j$ terminates, there exists $(i,p) \in \Sigma_j$ such that $n_i$ terminates and

$$x(j) \leq \Sigma_{l \in \Sigma_j} W_p(n) \geq \sigma_p + \Sigma_{l \in \Sigma_j} U_p(n)$$

(2)

where $x(k) = \max x(k,N)$ in any proper execution of $G$.

**Theorem 4**

Let $S$ be the set of indices of nodes terminating in $G$. For each $j \in S$, $x(j)$ is given by the minimum value of $x(j)$ in any nonnegative integer solution of

$$\min_{(i,p) \in \Sigma_j, l \in S} \bigg[ \sigma_p + \Sigma_{l \in \Sigma_j} U_p(n) - \Sigma_{l \in \Sigma_j} W_p(n) \bigg] \leq 0.$$  

(3)

Note: The proof of Theorem 4 establishes that the set $\{x(j), j \in S\}$ is a solution of (3). Since $x(j) = x^*(j)$, the minimum value of $x(j)$ in any nonnegative integer solution of (3), the set $\{x(j), j \in S\}$ simultaneously minimizes all of the quantities $x(j)$ over the set of all nonnegative integer solutions of (3). We call this the minimal solution of (3).
Theorem 5

The following iteration scheme converges in a finite number of steps to the minimal solution of (3), for \( j \in S \).

\[
x^{(0)}(j) = 0.
\]

Choose \( x^{(n+1)}_p(j) \) so that:

(i) Whenever

\[
\sum_{k=1}^{n} w_p(k) \geq \sigma_p + \sum_{k=1}^{n} u_p(k)
\]

\[
x^{(n+1)}_p(j) = x^{(n)}(j).
\]

(ii) Otherwise

\[
\sum_{k=1}^{n} w_p(k) > \sigma_p + \sum_{k=1}^{n} u_p(k)
\]

\[
x^{(n+1)}_p(j) = \min_{(i,p) \in \Sigma_j} x^{(n)}_p(r) \quad r \in S.
\]

**Proof:** We use induction on \( n \) to show that for all \( n \) and for \( j \in S \),

\[
x(j) \geq x^{(n)}(j).
\]

For \( n=0 \), \( x^{(0)}(j) = 0 \leq x(j) \).

Assume the result for \( n=r \). We show it true for \( n=r+1 \). Clearly \( x^{(r+1)}(j) \geq x^{(r)}(j) \). If \( x^{(r-1)}(j) = x^{(r)}(j) \), the result is immediate. Otherwise \( x^{(r+1)}(j) > x^{(r)}(j) \) so that for all \( (i,p) \in \Sigma_j \), (ii) of the iteration scheme applies.
Suppose, now, that the induction hypothesis fails for \( n=r+1 \). Then \( x^{(r+1)}(j) > x(j) \). Hence, for any \((i,p) \in \Sigma_j\), \( x_p^{(r+1)}(j) > x(j) \). Then by (ii) of the iteration scheme,

\[
\sum_{k=1}^{x(j)} W_p(k) \leq \sum_{k=1}^{x^{(r+1)}(j)-1} W_p(k) < \sigma_p + \sum_{k=1}^{x^{(r)}(i)} U_p(k) \quad \text{so that}
\]

\[
\sum_{k=1}^{x(j)} W_p(k) < \sigma_p + \sum_{k=1}^{x^{(r)}(i)} U_p(k) \leq \sigma_p + \sum_{k=1}^{x^{(r)}(i)} U_p(k) \leq \sigma_p + \sum_{k=1}^{x(i)} U_p(k) ,
\]

the last inequality following from the induction hypothesis for \( n=r \). But then, for all \((i,p) \in \Sigma_j\),

\[
\sigma_p + \sum_{k=1}^{x^{(r)}(i)} U_p(k) - \sum_{k=1}^{x^{(r)}(i)} W_p(k) > 0
\]

contradicting (3).

Hence \( x^{(n)}(j) \leq x(j) \) for all \( j \in S \). Since, for each \( j \), \( x^{(n)}(j) \) is bounded above by \( x(j) \), and since \( x^{(n+1)}(j) \geq x^{(n)}(j) \), the iteration must terminate with some \( n \) such that \( x^{(n+1)}(j) = x^{(n)}(j) \) for all \( j \in S \). For this value of \( n \), (i) of the iteration scheme must hold for all \( j \in S \) and some \((i,p) \in \Sigma_j\), otherwise (ii) would yield \( x^{(n+1)}(j) > x^{(n)}(j) \). For each such \( j \), and some \((i,p) \in \Sigma_j\), (i) yields

\[
\sum_{k=1}^{x^{(n)}(j)} W_p(k) \geq \sigma_p + \sum_{k=1}^{x^{(n)}(i)} U_p(k) \quad \text{i.e.}
\]

\[
\sigma_p + \sum_{k=1}^{x^{(n)}(i)} U_p(k) - \sum_{k=1}^{x^{(n)}(j)} W_p(k) \leq 0
\]
i.e. \((x^{(n)}(j), j \in S)\) is a solution of (3).

Combining this with the result \(x^{(n)}(j) \leq x(j)\) yields \(x^{(n)}(j) = x(j)\).

**Theorem 6**

Let \(G'\) be a strongly connected subgraph of a computation graph \(G\). Then \(G'\) is terminating if and only if:

(i) For some \(n_j \in G'\) there exists \((i,p) \in \Sigma_j\) such that \(n_i\) is terminating and \(n_j \not\in G'\), or

(ii) The system of inequalities

\[
\min_{(i,p) \in \Sigma_j, \quad d \in G'} \left[ \frac{x(i)}{p} + \sum_l U_p(n_l) - \sum_l W_p(n_l) \right] \leq 0
\]

(5)

for all \(j\) such that \(n_j \in G'\) has a solution in nonnegative integers.

**Theorem 7**

A maximal strongly connected subgraph \(G^S\) terminates in \(G\) if and only if there exists \(G^r\) such that \(G^r\) is self-terminating and \(G^r \geq G^S\).
6. **Conditions for Self-Termination**

The condition for self-termination for a loop with \( l \) nodes is the existence of a nonnegative integer solution of the following inequalities, obtained from Equation (5) of Theorem 6.

\[
\begin{align*}
\sum_1^{x(l)} W_l(n) & \geq x(l) \\
\sum_1^{x(l)} U_l(n) & + \sum_1^{x(l)} \sum_1^{x(l)} U_l(n)
\end{align*}
\]

and for \( k=1,2,\ldots,l-1 \)

\[
\begin{align*}
\sum_1^{x(k+1)} W_k(n) & \geq x(k) \\
\sum_1^{x(k+1)} U_k(n) & + \sum_1^{x(k+1)} \sum_1^{x(k+1)} U_k(n)
\end{align*}
\]

**Theorem 8**

A maximal strongly connected subgraph \( G' \) of a computation graph is self-terminating if and only if \( G' \) contains a self-terminating loop.

**Corollary 2**

Let \( \tilde{G} \) be a computation graph contained in a computation graph \( G \). If \( \tilde{G} \) is terminating in \( \tilde{G} \), then \( \tilde{G} \) is terminating in \( G \), and every node of \( G \) accessible from a node of \( \tilde{G} \) is terminating in \( G \). Also, if \( n_j \in \tilde{G} \), \( x(j,\tilde{G}) \geq x(j,G) \).

**Lemma 4**

Let \( \mathcal{E} = S_1, S_2, \ldots, S_N, \ldots \) be an execution of a loop \( L \). Then, if \( x(k+1,N) > 0 \)

\[
A_k + \sum_1^{x(k,N)} U_k(n) - \sum_1^{x(k+1,N)} W_k(n) \geq T_k - \tilde{W}_k
\]

where \( \tilde{W}_k = \max_n W_k(n) \).
PROOF: \( W_k \) exists since we have, for all \( n \), \( T_k \geq W_k(n) \).

We use induction on \( N \). For \( N=0 \), \( x(k+1,0) = 0 \) and there is nothing to be proved. Assuming the result for \( N=P \), we prove it for \( N=P+1 \). If \( x(k+1,P+1) = x(k+1,P) \) the result follows from the induction hypothesis. Otherwise \( k+1 \in S_{P+1} \). Thus \( x(k+1,P+1) = x(k+1,P)+1 \) and from Definition 2',

\[
A_k + \sum_1^P U_k(n) - \sum_1^P W_k(n) \geq T_k
\]

i.e.

\[
x(k,P) + \sum_1^P U_k(n) - \sum_1^P W_k(n) \geq T_k - W_k(x(k+1,P+1)) \geq T_k - W_k.
\]

The desired result now follows since \( x(k,P+1) \geq x(k,P) \).

In the case of a loop, the iteration scheme of Theorem 5 becomes:

\[
x^{(0)}(k+1) = 0.
\]

Choose \( x^{(n+1)}(k+1) \) so that:

(i) Whenever

\[
\sum_{r=1}^{x^{(n)}(k+1)} W_k(r) \geq \sigma_k + \sum_{r=1}^{x^{(n)}(k)} U_k(r), \quad x^{(n+1)}(k+1) = x^{(n)}(k+1) \quad (10)
\]

(ii) Otherwise

\[
\sum_{r=1}^{x^{(n+1)}(k+1)} W_k(r) \geq \sigma_k + \sum_{r=1}^{x^{(n)}(k)} U_k(r) \text{ and}
\]

\[
\sum_{r=1}^{x^{(n+1)}(k+1)-1} W_k(r) < \sigma_k + \sum_{r=1}^{x^{(n)}(k)} U_k(r).
\]

In order to prove Lemma 5 we require the following.
Lemma

\[ x^{(n+1)}(k+1) > x^{(n)}(k+1), \quad \implies x^{(n)}(k) > x^{(n-1)}(k). \]

PROOF: Since \( x^{(n+1)}(k+1) > x^{(n)}(k+1) \), condition (ii) of the iteration scheme applies so that

\[ \sum_{r=1}^{x^{(n+1)}(k+1)} w_{k}(r) \geq \sigma_k + \sum_{r=1}^{x^{(n)}(k)} u_{k}(r) > \sum_{r=1}^{x^{(n-1)}(k)} w_{k}(r). \]

But, in general,

\[ \sum_{r=1}^{x^{(n)}(k+1)} w_{k}(r) \geq \sigma_k + \sum_{r=1}^{x^{(n-1)}(k)} u_{k}(r). \]

Combining these two results yields

\[ \sigma_k + \sum_{r=1}^{x^{(n)}(k)} u_{k}(r) > \sigma_k + \sum_{r=1}^{x^{(n-1)}(k)} u_{k}(r) \]

i.e.

\[ \sum_{r=1}^{x^{(n)}(k)} u_{k}(r) > \sum_{r=1}^{x^{(n-1)}(k)} u_{k}(r) \]

i.e.

\[ x^{(n)}(k) > x^{(n-1)}(k). \]

Lemma 5

If \( l \) is the number of nodes in the loop \( L \), then, for \( k=1,2,\ldots,l \)

\( x(k,L) \geq 1 \) if and only if \( x^{(l)}(k) \geq 1 \). Also, if \( x(k,L) = 0 \), then \( L \) is self-terminating. Since \( x^{(l)}(k) = 0 \) implies \( x(k,L) = 0 \), the condition \( x^{(l)}(k) = 0 \) is sufficient for self-termination.
PROOF: Clearly, $x^{(l)}(k) \geq 1 \Rightarrow x(k,L) \geq 1$. Conversely, suppose $x(k+1,L) \geq 1$. Then there exists $n$ such that $x^{(i)}(k+1) = 0$ for $i=0,1,2,\ldots,n-1$ and $x^{(n)}(k+1) > 0$. Assume $n > l$.

Now $x^{(n)}(k+1) > x^{(n-1)}(k+1)$ so that by the above lemma, $x^{(n-1)}(k) > x^{(n-2)}(k)$. Applying the Lemma $l$ times yields $x^{(n-l)}(k+1) = 0 > x^{(n-l-1)}(k+1) = 0$, a contradiction.

Hence $x(k+1,L) \geq 1 \Rightarrow x^{(l)}(k+1) \geq 1$. The rest of the theorem follows.

We assume, in what follows, that $\{U_k(n)\}$ is periodic with period $p_k$ and $\{W_k(n)\}$ is periodic with period $q_k$. Write

$$
\tilde{U}_k = \frac{\sum_{n=1}^{p_k} U_k(n)}{p_k},
$$

$$
\tilde{W}_k = \frac{\sum_{n=1}^{q_k} W_k(n)}{q_k},
$$

$$
\alpha_k = \frac{\tilde{U}_k}{\tilde{W}_k}.
$$

The following inequalities are self evident.
\[
\begin{align*}
\sum_{n=1}^{x(k+1)} W_k(n) &\leq \tilde{W}_k \{x(k+1) + (q_k - 1)\} \\
\sum_{n=1}^{x(k)} U_k(n) &\geq \tilde{U}_k \{x(k) - (p_k - 1)\} \\
\sum_{n=1}^{x(k+1,N)} W_k(n) &\geq \tilde{W}_k \{x(k+1,N) - (q_k - 1)\} \\
\sum_{n=1}^{x(k,N)} U_k(n) &\leq \tilde{U}_k \{x(k,N) + (p_k - 1)\}
\end{align*}
\]

By applying the first two of these inequalities to the fundamental loop inequalities (6), we obtain the following system:

\[x(k+1) \geq \alpha_k x(k) + \frac{\sigma_k}{\tilde{W}_k} - \alpha_k (p_k - 1) - (q_k - 1).\]

Put

\[\beta_{k+1} = \frac{\sigma_k}{\tilde{W}_k} - \alpha_k (p_k - 1) - (q_k - 1).\]

Then this system can be written in a matrix form as

\[
\begin{bmatrix}
1 & 0 & \ldots & 0 & \ldots & \ldots & -\alpha_l \\
-\alpha_1 & 1 & 0 & \ldots & 0 & \ldots & 0 \\
0 & -\alpha_2 & 1 & 0 & \ldots & 0 & \ldots \\
\vdots & \vdots & \ddots & \vdots & \ddots & \ddots & \vdots \\
0 & \ldots & \ldots & -\alpha_{l-1} & 1 \\
\end{bmatrix}
\begin{bmatrix}
x(1) \\
x(2) \\
\vdots \\
x(l)
\end{bmatrix}
\geq
\begin{bmatrix}
\beta_1 \\
\beta_2 \\
\vdots \\
\beta_l
\end{bmatrix}
\]

i.e.

\[
(I - A) X \geq \beta
\]

(7)

Now apply the last two inequalities of (8) to the inequality of Lemma 4. We obtain

26
\[ x(k+1,N) \leq x_k x(k,N) + \frac{A_k - T_k + W_k}{W_k} + \alpha_k (p_k-1) + (q_k-1) \]

Put
\[ \gamma_{k+1} = \frac{A_k - T_k + W_k}{W_k} + \alpha_k (p_k-1) + (q_k-1). \]

Then this system can be written
\[ (I - A)X_N \leq \gamma \] (9)

where \( x(k,N) \) is the \( k \)-th component of \( X_N \) and \( \gamma_k \) is the \( k \)-th component of \( \gamma \).

The fact that \( A \) is a positive matrix yields the following:

**Lemma 6**

If the matrix inequality (7) holds, then for any positive integer \( n \)
\[ (I - A^n) X \geq (A^{n-1} + A^{n-2} + \ldots + A + 1) \beta. \]

If (9) holds, then for any positive integer \( n \)
\[ (I - A^n)X_N \leq (A^{n-1} + A^{n-2} + \ldots + A + 1) \gamma. \]

Define \( g = \alpha_1 \alpha_2 \ldots \alpha_r \). The matrix \( I - A^l \) is diagonal with each diagonal element equal to \( 1 - g \).

**A. Loops with \( g < 1 \)**

**Theorem 9**
\[ g < 1 \implies L \text{ is self-terminating}. \]

**PROOF:** Suppose \( L \) is not self-terminating. Then Lemma 5 implies that there exists \( P \) such that if \( N > P, x(k,N) \geq 1 \) for all \( k \). For all such \( N \),
\[ (I-A)X_N \leq \gamma \text{ by Lemma 4 and inequality (9)}. \]

By Lemma 6
\[ (I-A^l)X_N = (1-g)X_N \leq (A^{l-1} + \ldots + A + 1) \gamma. \]
Thus, each component of $X_N$ has an upper bound independent of $N$ so that $L$ is self-terminating.

B. **Loops with $g=1$**

When $g=1$, $\det (I-A) = 0$ so that the homogeneous system

$$(I - A)X = 0$$

has a nontrivial solution, as follows.

$$\begin{align*}
  x_1 &= a \\
  x_2 &= \alpha_1 a \\
  &\ldots \ldots \ldots \\
  x_l &= \alpha_1 \alpha_2 \ldots \alpha_{l-1} a
\end{align*}$$

where $a$ is an arbitrary parameter. Thus, by judiciously choosing $a$, we can arrange that each $x_i$, $i=1,2,\ldots,l$ be a multiple of $p_j q_j$, $j=1,2,\ldots,l$. i.e. $p_j q_j | x_i$, $i, j=1,2,\ldots,l$. This is possible since each $\alpha_k$ is rational. Let $\vec{X}$ be the minimal positive integer solution of $(I-A)X = 0$ subject to these conditions. $\vec{X}$ exists, for the minimal $a$ which yields a solution satisfying the above conditions will yield a component-wise minimal positive integer solution $\vec{X}$.

**Theorem 10**

If $L$ is self-terminating with $g=1$, then at least one component of $X$ is less than the corresponding component of $\vec{X}$.

**PROOF:** $L$ is self-terminating, which implies that $X$ is the smallest non-negative integer solution of (6), i.e.

$$x(k+1) \geq \sum_{n=1}^{x(k+1)} W_{k}(n) + \sum_{n=1}^{x(k)} U_{k}(n).$$
Therefore
\[
\sum_{n=1}^{x(k+1)} W_k(n) - \sum_{n=1}^{\tilde{x}(k+1)} W_k(n) \geq \sigma_k + \sum_{n=1}^{x(k)} U_k(n) - \sum_{n=1}^{\tilde{x}(k+1)} W_k(n).
\]

Assume, for all \( k \), that \( x(k) \geq \tilde{x}(k) \). Then
\[
\frac{x(k+1)}{\tilde{x}(k+1)+1} \geq \sigma_k + \frac{x(k)}{\sum_{l} U_k(n)} - \frac{\tilde{x}(k+1)}{\tilde{W}_k(n)}.
\]

But \( \tilde{x}(k+1) \) is a multiple of \( q_k \). Hence
\[
\frac{x(k+1) - \tilde{x}(k+1)}{\sum_{l} W_k(n)} \geq \sigma_k + \frac{x(k)}{\sum_{l} U_k(n)} - \frac{\tilde{x}(k+1)}{\tilde{W}_k(n)}.
\]

Finally, \( \tilde{x}(k+1) = \sigma_k \tilde{x}(k) \)
\[
= \frac{\tilde{U}_k}{\tilde{W}_k} \tilde{x}(k)
\]
so that \( \tilde{W}_k \tilde{x}(k+1) = \tilde{U}_k \tilde{x}(k) \)
\[
= \sum_{l} U_k(n)
\]
since \( \tilde{x}(k) \) is a multiple of \( p_k \). Then we have
\[
\frac{x(k+1) - \tilde{x}(k+1)}{\sum_{l} W_k(n)} \geq \sigma_k + \frac{x(k)}{\sum_{l} U_k(n)} - \frac{\tilde{x}(k)}{\tilde{W}_k(n)}
\]
\[
= \sigma_k - \frac{x(k)}{\tilde{x}(k)+1}
\]
\[
= \frac{x(k) - \tilde{x}(k)}{\sum_{l} U_k(n)}
\]
again since \( \tilde{x}(k) \) is a multiple of \( p_k \). But then \( x(k+1) - \tilde{x}(k+1) \) is a smaller nonnegative integer solution of (6), a contradiction.
This theorem leads to a finite procedure for determining self-termination. The iteration procedure (10), is followed until either:

(i) it terminates, yielding the number of performances of each operation, or else
(ii) for all \( k \), \( x^{(n)}(k) > \tilde{x}(k) \), establishing nontermination.

C. Loops with \( g > 1 \)

Put \( P = A^{l-1} + A^{l-2} + \ldots + A + I \). Then Lemma 6 yields \((I-A^l)X \geq PB\), or, \((1-g)X \geq PB\), or, since \( g > 1 \)

\[ X \leq \frac{1}{1-g} PB. \]

From this inequality it follows that if any component of \( PB \) is positive, the loop is not self-terminating. For if the loop is self-terminating, let \( X \) be the minimum nonnegative integer solution of (6). Then \( X \) satisfies

\[ X \leq \frac{1}{1-g} PB \quad (15) \]

so that some component of \( X \) is negative, a contradiction.

The upper bound on the values of \( x(k,L) \) given by (15) leads to the following finite procedure to determine whether the loop \( L \) is self-terminating or not: Apply the iteration scheme (10) until either

(i) it terminates, yielding the number of performances of each operation, or else
(ii) some component of \( x^{(n)}(k) \) exceeds the upper bound obtained from (15).
7. Properties of Queue Lengths

Define

\[ q(p,N) = A_p + \sum_l U_p(n) \cdot \sum_l W_p(n). \]

The queue length on \( d_p \) immediately after the \( x(j,N) \)-th initiation of \( O_j \) is \( q(p,N) \) if the \( x(i,N) \)-th performance of \( O_i \) has been completed; otherwise, the queue length is \( q(p,N) - U_p(x(i,N)) \).

Theorem 12

Let \( d_p \) be a branch of \( G \) from \( n_i \) to \( n_j \). If \( n_i \in G^r \) and \( n_j \in G^s \), where \( G^r \) and \( G^s \) are different maximal strongly connected subgraphs of \( G \), then the queue length of \( d_p \) is uniformly bounded if and only if \( n_i \) terminates.

Theorem 13

Let \( G' \) be a nonterminating maximal strongly connected subgraph of \( G \). Every loop of \( G' \) for which \( g > 1 \) contains at least one branch for which the queue length is not uniformly bounded. Also, the queue length for every branch which lies in a loop with \( g = 1 \) is uniformly bounded.

PROOF: Let \( L \) be a loop in \( G' \) consisting of the nodes \( n_1, n_2, \ldots, n_\ell \) and branches \( d_1, d_2, \ldots, d_\ell \), with \( d_k \) directed from \( n_k \) to \( n_{k+1} \), \( 1 \leq k \leq \ell-1 \), and \( d_\ell \) directed from \( n_\ell \) to \( n_1 \). Choose a proper execution of \( \mathcal{E} \), and consider the linear form

\[ F = q(1,N) + \frac{w_1}{U_2} q(2,N) + \frac{w_1 w_2}{U_2 U_3} q(3,N) + \ldots + \frac{w_1 w_2 \ldots w_{\ell-1}}{U_2 U_3 \ldots U_\ell} q(\ell,N). \]
Now \( q(k,N) = A_k + \sum_{l=1}^{x(k,N)} U_k(n) - \sum_{l=1}^{x(k+1,N)} W_k(n) \). By applying the inequalities (8) to this equation, we obtain

\[
q(k,N) \leq A_k + \bar{U}_k x(k,N) - \bar{W}_k x(k+1,N) + \{ \bar{U}_k(p_k-1) + \bar{W}_k(q_k-1) \}.
\]

Substitute for \( q(k,N) \), \( 1 \leq k \leq \ell \), in \( F \) yields

\[
F \leq \sum_{l=1}^{\ell} (\frac{g-1}{g}) x(l,N) + A_1 + \frac{\bar{W}_1}{\bar{U}_2} A_2 + \ldots + \frac{\bar{W}_1 \bar{W}_2 \ldots \bar{W}_{\ell-1}}{\bar{U}_2 \bar{U}_3 \ldots \bar{U}_{\ell}} A_{\ell}
\]

\[
+ \{ \bar{U}_1(p_1-1) + \bar{W}_1(q_1-1) \} + \frac{\bar{W}_1}{\bar{U}_2} \{ \bar{U}_2(p_2-1) + \bar{W}_2(q_2-1) \} + \ldots
\]

\[
+ \frac{\bar{W}_1 \bar{W}_2 \ldots \bar{W}_{\ell-1}}{\bar{U}_2 \bar{U}_3 \ldots \bar{U}_{\ell}} \{ \bar{U}_\ell(p_{\ell}-1) + \bar{W}_\ell(q_{\ell}-1) \}.
\]

Suppose \( g=1 \). Then by taking the \( \leq \) case in the above inequality, we see that \( F \) is bounded above by a constant, and since each \( q(k,N) \) is bounded below by zero, a uniform upper bound on each of the quantities \( q(k,N) \) must exist.

Suppose \( g > 1 \). Then the coefficient of \( x(1,N) \) is positive. Since \( G' \) is nonterminating, \( x(1,N) \) grows without limit in any proper execution of \( G \). By choosing the \( \geq \) case in the above inequality, we see that \( F \) must also grow without limit showing that some \( q(k,N) \) must tend to infinity.
8. **Conclusion**

An analysis has been presented of determination for computation graphs, and of termination and queueing in the case that \( \{U_p(n)\} \) and \( \{W_p(n)\} \) are periodic. An open problem is the analysis of termination and queueing in the case that \( \{U_p(n)\} \) and \( \{W_p(n)\} \) are not periodic. However, such an analysis would appear to be of little practical value since most arithmetic calculations which one would wish to effect by a computation graph seem to be iterative.

Many other problem areas immediately suggest themselves. Among these are:

(i) the synthesis of a computation graph given a computation to be performed

(ii) the determination of a proper execution which minimizes the time for a computation

(iii) programming techniques

(iv) the allocation of storage registers for data queues and their utilization so as to minimize the number required.
References
