

THE UNIVERSITY OF MICHIGAN
COMPUTING RESEARCH LABORATORY¹

DETERMINISM IN PARALLEL SYSTEMS

Vaclav Rajlich

CRL-TR-4-82

OCTOBER 1982

Room 1079, East Engineering Building
Ann Arbor, Michigan 48109
USA
Tel: (313) 763-8000

¹This research was supported in part by the National Science Foundation under Grant No. MCS-78 00763. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the funding agencies.

Determinism in Parallel Systems

Vaclav Rajlich

Department of Computer and Communication Sciences

University of Michigan

Ann Arbor, MI 48109

This research was supported in part by the National Science Foundation
under Grant No. MCS-78 00763.

ABSTRACT

The report deals with determinism in parallel systems, in which an event can be infinitely postponed. (the so-called "unfair systems" or systems with "starvation to death"). A hierarchy of deterministic, locally deterministic, and strongly locally deterministic systems is defined. It is proved that they form a proper hierarchy. The proof is based in part on a variant of Church-Rosser properties.

1. INTRODUCTION

One of the important problems in parallel systems is the problem of determinism. The determinism in persistent systems was investigated in [3, 5]. In this paper, we deal with a generalization to the systems which are not persistent, i.e. systems where an event can be postponed, possibly infinitely. The systems of this kind were also called "unfair systems" or systems with "starvation to death". The techniques used here are an application of the ones published in [1].

The paper defines a hierarchy of deterministic, locally deterministic, and strongly locally deterministic systems and proves that they form a proper hierarchy. Parts of the proof are based on a variant of Church-Rosser properties, as investigated in [2] and other papers. (See [2] for a list of references.)

Chapter 2 contains definitions of systems, traces, histories, and related notions, and Chapter 3 contains the main result of the paper. The result can be viewed as a criterion for determining a determinism in systems, and it is fully compatible with H-graphs [4] and similar definitions of systems.

2. SYSTEMS

We shall start our formal exposition with the definition of systems:

DEFINITION 2.1.

Let L be a set of locations, V a set of values, then $s:L \rightarrow V$ will be called a state.

REMARK.

As usual, we shall assume functions to be subsets of Cartesian product $s \subset L \times V$ such that

- [i] for every $a \in L$, there exists $v \in V$ such that $\langle a, v \rangle \in s$
- [ii] if $\langle a, v \rangle \in s$ and $\langle a, w \rangle \in s$ then $v=w$. Let $L_1 \subset L$, then denote $s/L_1 = \{\langle a, v \rangle \mid \langle a, v \rangle \in s \text{ and } a \in L_1\}$.

We say $s(a)=v$ iff $\langle a, v \rangle \in s$.

DEFINITION 2.2.

Let L be a set of locations, V a set of values. Then e is a virtual event iff $e = \langle e_L, e_R \rangle$ (denoted $e_L \Rightarrow e_R$) and there exists $L_1 \subset L$, such that $e_L: L_1 \rightarrow V$, $e_R: L_1 \rightarrow V$. Then e_L, e_R are called left and right side of virtual event e , respectively. Denote $\text{Dom } e = \{a \in L_1 \mid e_L(a) \neq e_R(a)\}$.

DEFINITION 2.3.

A system is a quadruple $\{L, V, s_0, E\}$ where L is a set of locations, V is a set of values, $s_0: L \rightarrow V$ is the original state, and E is a set of virtual events.

The state of the system undergoes changes, as defined in the following definitions:

DEFINITION 2.4.

Let s be a state, e be a virtual event such that $e_L \subset s$. Then define a new state $t = (s - e_L) \cup e_R$ and $\langle s, t \rangle$ is called actual event. The relationship of s, e , and t will be denoted as $s \cdot e = t$. If $e_L \subset s$, we also say e is scheduled in s .

DEFINITION 2.5.

Let $S = \langle L, V, s_0, E \rangle$ be a system. State s is reachable in the system S iff $s = s_0$ or there exists a reachable state t such that $t \cdot e = s$.

Traces are a tool for description of the "dynamics" of the systems. They are sequences (finite or infinite) of virtual events which consecutively happened in the system. Formally, they are described in the following definitions:

DEFINITION 2.6.

Let A be an alphabet, then a finite nonempty word is a function $w:\{1,\dots,n\}\rightarrow A$ where $n\geq 1$. An infinite word is a function $w:\{1,2,\dots\}\rightarrow A$, and empty word $\lambda=\emptyset$. Word is either is finite, infinite, or empty word. Length of a word $|w|$ is equal to 0, n , and ω for empty, finite, and infinite words, respectively. Prefix of word w of length n (denoted w_n) is defined in the following way: for $n\geq |w|$, $w_n=w$, for $n<|w|$, $w_n=w|\{1,\dots,n\}$. Denote operation concatenation in the following way: If both $v:\{1,\dots,m\}\rightarrow A$ and $w:\{1,\dots,n\}\rightarrow A$ are finite words, then $v\cdot w$ is a function $v\cdot w:\{1,\dots,m+n\}\rightarrow A$ so that $v\cdot w|\{1,\dots,m\}=v$ and for every i , for which $1<i\leq n$, $v\cdot w(i+m)=w(i)$. For empty word λ , $\lambda\cdot w=w$, $\lambda\cdot\lambda=\lambda$. If w is infinite, then for every v , $w\cdot v=w$. If $v:\{1,\dots,m\}\rightarrow A$ is finite, and w infinite, then $v\cdot w$ is an infinite word $v\cdot w:\{1,2,\dots\}\rightarrow A$ so that $v\cdot w|\{1,\dots,m\}=v$ and for every $i>1$, $v\cdot w(i+m)=w(i)$.

Let u,v be words, then u is prefix of v (denoted $u\leq v$) iff there exists a word w such that $u\cdot w=v$. Words v,w are called compatible (denoted $v\leq w$) if either $v\leq w$ or $w\leq v$. Let v_0, v_1, v_2, \dots be words such that $v_0\leq v_1\leq v_2\leq \dots\leq v$, denote $v=\lim_{i\rightarrow\infty} v_i$.

It should be noted that the infinite words and the corresponding topology were extensively studied in [1]. We will use the following lemma of [1]: If $v_0\leq v_1\leq \dots$, then there exists a unique v such that $v=\lim_{i\rightarrow\infty} v_i$.

DEFINITION 2.7.

Let $\langle L, V, so, E \rangle$ be a system, then trace is defined in the following way: Let s be a reachable state, then $\langle s, \lambda \rangle$ is an empty trace. If $\langle s, u \rangle$ is

a trace and there exists $e \in E$ such that e is scheduled in $s \cdot u$, then $\langle s, u \cdot e \rangle$ is a trace. Let $\langle s, u_1 \rangle, \langle s, u_2 \rangle, \dots$ be a sequence of traces such that $u_1 \leq u_2 \leq \dots$, then $\langle s, \lim_{i \rightarrow \infty} u_i \rangle$ is a trace. Trace $\langle s, u \rangle$ is complete iff $s = s_0$ and either u is infinite or $s \cdot u$ is a terminal state, i.e. for no $e \in E$, e is scheduled in $s \cdot u$. If $\langle s, u \rangle$ is a trace and s is obvious from the context, then u will sometimes also be called trace.

In the next definition, history is defined as a sequence of values held at a location $a \in L$:

DEFINITION 2.8.

History of $a \in L$ in trace $\langle s, u \rangle$ (denoted $h(a, \langle s, u \rangle)$) is defined in the following way:

- (i) For empty trace, $h(a, \langle s, \lambda \rangle) = s(a)$.
- (ii) Suppose that $h(a, \langle s, u \rangle)$ is defined and event $e \in E$ is scheduled in $s \cdot u$. Then we have the following two cases:
 - (a) $a \notin \text{Dom } e$ (i.e. event e does not affect the value in location a), then $h(a, \langle s, u \cdot e \rangle) = h(a, \langle s, u \rangle)$.
 - (b) $a \in \text{Dom } e$, then $h(a, \langle s, u \cdot e \rangle) = h(a, \langle s, u \rangle) \cdot e_R(a)$ (i.e. the new value will be added to the history).

NOTE: if $u_1 \leq u_2 \leq \dots$, then $h(a, \langle s, u_1 \rangle) \leq h(a, \langle s, u_2 \rangle) \leq \dots$

- (iii) Let $u_1 \leq u_2 \leq \dots$, then $h(a, \langle s, \lim_{i \rightarrow \infty} u_i \rangle) = \lim_{i \rightarrow \infty} h(a, \langle s, u_i \rangle)$.

We may state the following lemma:

LEMMA 2.1.

For every $n \leq |h(a, \langle s, w \rangle)|$ there exists $m \geq n$ such that $\lim_{i \rightarrow \infty} h(a, \langle s, w \rangle) = h(a, \langle s, w \rangle)_m$.

PROOF

By induction on m .

3. DETERMINISM

This chapter contains several definitions of determinism in systems. Intuitively speaking, a system is deterministic if no matter what the particular choice of sequence of events, the histories of each location are compatible. In [3], a stronger definition of determinism requiring uniqueness of the history for each location was used. In this chapter, we shall also develop several criteria for determining whether a system is deterministic: strong local determinism and local determinism, and prove that strong local determinism, local determinism, and determinism form a proper hierarchy.

DEFINITION 3.1.

Let $\langle L, V, s_0, E \rangle$ be a system, and $A \subseteq L$ be a subset of the set of locations L . The system is deterministic on A iff for every $a \in A$ and every couple of complete traces $\langle s_0, u \rangle, \langle s_0, v \rangle$, $h(a, \langle s_0, u \rangle) \leq h(a, \langle s_0, v \rangle)$.

DEFINITION 3.2.

Let $\langle L, V, s_0, E \rangle$ be a system, and $A \subseteq L$ be a subset of the set of locations L . The system is locally deterministic on A iff for every $a \in A$ and every couple of finite traces u, v there exist traces u^1, v^1 such that $h(a, \langle s, u \cdot u^1 \rangle) = h(a, \langle s, v \cdot v^1 \rangle)$.

DEFINITION 3.3.

Let $\langle L, V, s_0, E \rangle$ be a system. It is strongly locally deterministic on A iff for every $a \in A$, for every reachable state s , and for all virtual events e, f which are scheduled in s , there exist traces x and y , where $|y| \leq 1$, $h(a, \langle s, e \cdot x \rangle) = h(a, \langle s, f \cdot y \rangle)$, and $s \cdot e \cdot x = s \cdot f \cdot y$.

We may illustrate the definitions by the following lemmas.

LEMMA 3.1.

Let $B \subset A$, then S is deterministic (locally deterministic, strongly locally deterministic) on A implies S is deterministic (locally deterministic, strongly locally deterministic) on B .

PROOF

Obvious.

LEMMA 3.2.

Let $S = \langle L, V, s_0, E \rangle$. Then the system is strongly locally deterministic on L if for every $a \in L$, for every reachable state s , and for all virtual events e, f which are scheduled in s , there exist traces x and y where $|y| \leq |x|$, $h(a, \langle s, e \cdot x \rangle) = h(a, \langle s, f \cdot y \rangle)$.

PROOF

The only thing we have to prove is that assumptions of Lemma 3.2 imply $s \cdot e \cdot x = s \cdot f \cdot y$. Note that for every $a \in L$, $h(a, \langle s, e \cdot x \rangle)$ is finite, because $f \cdot y$ is finite and Lemma 2.1. says $|h(a, \langle s, e \cdot x \rangle)| \leq |e \cdot x|$. Denote $t(a)$ the last value of $h(a, \langle s, e \cdot x \rangle)$, then $t = \{ \langle a, t(a) \rangle \mid a \in L \} = s \cdot f \cdot y = s \cdot e \cdot x$, which completes the proof.

The main result of the paper is the following theorem.

THEOREM 3.1.

Let S be a system, A a subset of all locations. Then if S is strongly locally deterministic on A then S is locally deterministic on A and this implies S is deterministic on A .

PROOF

- (i) First we will prove the strong local determinism implies local determinism.

(The proof follows techniques of [2], Lemma 2.5.) Suppose $S = \langle L, V, s_0, E \rangle$ is a system, $A \subset L$, and S is strongly locally deterministic on A . Then we can prove the following statement: Let e be an event, v a finite trace and $a \in A$, then there exists traces x^1 and y^1 where $|y| \leq 1$, $s \cdot e \cdot x^1 = s \cdot v \cdot y^1$, and $h(a, \langle s, e \cdot x^1 \rangle) = h(a, \langle s, v \cdot y \rangle)$. (The proof is by induction on $|v|$).

With this statement, we can prove local determinism. (The proof is by induction on $|u|$).

(ii) Now we will prove that the local determinism implies determinism. Suppose S is locally deterministic in A , but not deterministic on A . Then there exist two complete traces u, v and $a \in A$ such that it is not true that $h(a, \langle s_0, u \rangle) \leq h(a, \langle s_0, v \rangle)$. Then there exists m such that

$$h_m(a, \langle s_0, u \rangle) \neq h_m(a, \langle s_0, v \rangle) \quad (*)$$

Hence by Lemma 2.1, there exists k, n such that $h_m(a, \langle s_0, u \rangle) = h_m(a, \langle s_0, u \rangle_k)$ and $h_m(a, \langle s_0, v \rangle) = h_m(a, \langle s_0, v \rangle_n)$. However, from the definition of the local determinism, there exist traces u^1, v^1 such that $h(a, \langle s_0, u \cdot u^1 \rangle) = h(a, \langle s_0, v \cdot v^1 \rangle)$ which is a contradiction with (*).

The next two counter-examples will establish the facts that the hierarchy of Theorem 3.1 is the proper one, i.e. that the implications cannot be replaced by equivalences.

EXAMPLE 3.1.

This is an example of a system which is deterministic on A but not locally deterministic on A .

Let $S = \langle L, V, s_0, E \rangle$ where $L = \{a, b\}$, $A = L$, $V = \{0, 1\}$,

$s_0 = \{\langle a, 0 \rangle, \langle b, 0 \rangle\}$, $E = \{e_1, e_2\}$ where

$e_1 = \langle a, 0 \rangle \Rightarrow \langle a, 1 \rangle$,

$e_2 = \{\langle a, 0 \rangle\} \Rightarrow \{\langle a, 1 \rangle, \langle b, 1 \rangle\}$

$\{\langle b, 0 \rangle\}$

Then there are two complete traces $\langle s_0, e_1 \rangle$ and $\langle s_0, e_2 \rangle$, where
 $h(a, \langle s_0, e_1 \rangle) = \langle 0, 1 \rangle$, $h(a, \langle s_0, e_2 \rangle) = \langle 0, 1 \rangle$, $h(b, \langle s_0, e_1 \rangle) = \langle 0 \rangle$, $h(b, \langle s_0, e_2 \rangle) = \langle 0, 1 \rangle$,
hence the system is deterministic but not locally deterministic.

EXAMPLE 3.2.

This is an example of a system which is locally deterministic but not strongly locally deterministic.

Let $S = \langle L, V, s_0, E \rangle$, $L = \langle a, b, c, d \rangle$, $A=L$, $V=\{0,1\}$,

$s_0 = \{ \langle a, 0 \rangle, \langle b, 0 \rangle, \langle c, 0 \rangle \}$, $E = \{ e_1, e_2, e_3, e_4, e_5, e_6 \}$,

where $e_1 = \langle a, 0 \rangle \Rightarrow \langle a, 1 \rangle$,

$e_2 = \langle c, 0 \rangle \Rightarrow \langle c, 1 \rangle$,

$e_3 = \{ \langle a, 1 \rangle, \langle b, 0 \rangle \} \Rightarrow \{ \langle a, 1 \rangle, \langle b, 1 \rangle \}$,

$e_4 = \{ \langle b, 1 \rangle, \langle c, 0 \rangle \} \Rightarrow \{ \langle b, 1 \rangle, \langle c, 1 \rangle \}$,

$e_5 = \{ \langle b, 0 \rangle, \langle c, 1 \rangle \} \Rightarrow \{ \langle b, 1 \rangle, \langle c, 1 \rangle \}$,

$e_6 = \{ \langle a, 0 \rangle, \langle b, 1 \rangle \} \Rightarrow \{ \langle a, 1 \rangle, \langle b, 1 \rangle \}$.

Then this system is locally deterministic, but not strongly locally deterministic.

In the following example, we shall give the illustration of a strongly locally deterministic system for which $h(a, \langle s_0, u \rangle) \leq h(a, \langle s_0, v \rangle)$ (u , and v are complete traces), but $h(a, \langle s_0, u \rangle) \neq h(a, \langle s_0, v \rangle)$. In another terminology, it is an illustration of a system which is unfair [3], or nonpersistent, or contains starvation to death.

EXAMPLE 3.3

Let $S = \langle L, V, s_0, E \rangle$ where $L = \{a, b\}$, $V = \{0, 1\}$,

$s_0 = \{ \langle a, 0 \rangle, \langle b, 0 \rangle \}$, $E = \{ e_1, e_2, e_3, e_4 \}$ where

$e_1 = \langle a, 0 \rangle \Rightarrow \langle a, 1 \rangle$,

$$e_2 = \langle a, 1 \rangle \Rightarrow \langle a, 0 \rangle,$$

$$e_3 = \langle b, 0 \rangle \Rightarrow \langle b, 1 \rangle,$$

$$e_4 = \langle b, 1 \rangle \Rightarrow \langle b, 0 \rangle.$$

Then the system is strongly locally deterministic. Consider the following two complete traces:

$u = e_{12}e_{12}e_{12}\dots, v = e_{34}e_{34}e_{34}\dots$. Then $h(a, \langle s_0, u \rangle) = \langle 0, 1, 0, 1, \dots \rangle$, while $h(a, \langle s_0, v \rangle) = \langle 0 \rangle$.

REFERENCES

- [1] L. Boasson, M. Nivat, Adherence of Languages, J. of Computer and System Science 20 (1980) 295-309.
- [2] G. Huet, Confluent Reductions: Abstract Properties and Applications to term Rewriting Systems, J. of the Assoc, Comput. M. 22 (1980) 797-821.
- [3] R.M. Karp, R.E. Miller, Parallel Program Schemata, J. of Computer and System Science 3, (1969) 147-195.
- [4] T. Pratt, Application of Formal Grammars and Automata to Programming Language Definition, in Yeh, R.T. (ed.), Applied Computation Theory (Prentice-Hall, 1976).
- [5] V. Rajlich, Determinism in Relational Systems, in Claus, V., Ehrig, H., and Rosenberg, G., ed., Graph-Grammars and their Application to Computer Science and Biology, Lecture Notes in Computer Science 73, (Springer Verlag 1979) 401-408.