Outline for January 11, 2001

1. Greetings and felicitations!
   a. First part of project due Friday
   b. Web page up and running!

2. Process models
   a. Theorem: If a system is mutually noninterfering, it is determinate.
   b. Theorem: Let $f_p$ be an interpretation of process $p$. Let $\prod$ be a system of processes, with $p \in \prod$. If for all such $p$, $\text{domain}(p) \neq \emptyset$ and $\text{range}(p) \neq \emptyset$, but $f_p$ unspecified, is determinate for all $f_p$, then all processes in $\prod$ are mutually noninterfering.
   c. Maximally parallel system: determinate system for which the removal of any pair from the relation $\rightarrow$ makes the two processes in the pair interfering processes.

3. Critical section problem
   a. Mutual exclusion
   b. Progress
   c. Bounded wait

4. Classical problems
   a. Producer/consumer
   b. Readers/writers (first: readers priority; second: writers priority)
   c. Dining philosophers

5. Basic language constructs
   a. Semaphores
   b. Send/receive

6. Evaluating higher-level language constructs
   a. Modularity
   b. Constraints
   c. Expressive power
   d. Ease of use
   e. Portability
   f. Relationship with program structure
   g. Process failures, unanticipated faults (exception handling)
   h. Real-time systems

7. Higher-level language constructs
   a. Monitors
   b. Crowd monitors
   c. Invariant expressions
   d. CSP
   e. RPC
   f. ADA™
Mutual Non-Interference and Determinism

Introduction

A determinate system of processes is a set of process that always produces the same output given the same input. A mutually non-interfering set of processes is a set of processes that do not interfere with the input or output of one another. The question is, to what degree are these the same concepts?

Formal Definitions and Notations

- A system of processes \( S = (\Pi, \to) \) is a set of processes \( \Pi = \{ p_1, \ldots, p_n \} \) and a precedence relation \( \to : \Pi \times \Pi \). The \( \to \) relation is a partial ordering (we define \( p \to p \) as true). When \( p \to q \), process \( p \) must complete before process \( q \) may begin.
- Each process \( p \in \Pi \) has an associated set of input memory locations called \( \text{domain}(p) \) and an associated set of output memory locations \( \text{range}(p) \neq \emptyset \). An interpretation \( f_p \) of \( p \) associates values with each set of memory locations. The set of all inputs to \( S \) is abbreviated \( \text{domain}(S) \), and the set of all outputs from \( S \) is abbreviated \( \text{range}(S) \).
- Two systems of processes \( S = (\Pi, \to) \) and \( S' = (\Pi', \to') \) are equivalent if
  
  a. \( \Pi = \Pi' \);
  
  b. \( \to \neq \to' \); and
  
  c. if \( S \) and \( S' \) are given the same element of \( \text{domain}(S) \), then they output the same element of \( \text{range}(S) \).
- An execution sequence \( \alpha \) is any string of process initiation and termination events satisfying the precedence constraints of the system.
- \( V(M_i, \alpha) \) is the sequence of values written into memory location \( M_i \) at the termination of processes in \( \alpha \). The final value stored in \( M_i \) after execution sequence \( \alpha \) completes is represented by \( F(M_i, \alpha) \).
- A determinate system of processes is a system of processes \( S \) for which each element of \( \text{domain}(S) \) produces the same set \( \text{range}(S) \) regardless of the order or overlapping of the elements of \( S \). More formally, a system \( S \) is determinate if, for any initial state and for all execution sequences \( \alpha \) and \( \alpha' \) of \( S \), \( V(M_i, \alpha) = V(M_i, \alpha') \)
- A mutually noninterfering system of processes is a system of processes \( S \) in which all pairs of processes meet the Bernstein conditions. Processes \( p \) and \( q \) are noninterfering if either process is a predecessor of the other, or the processes satisfy the Bernstein conditions.
- The initiation of a process \( p \) is written \( \overline{p} \), and the termination of \( p \) is written \( p \).

Relationship of Determinate Systems and Mutually Noninterfering Systems

**Theorem 1:** If a system is mutually non-interfering, it is determinate.

**Theorem 2:** Let \( S \) be a system with \( \text{domain}(p) \) and \( \text{range}(p) \) specified, \( \text{range}(p) \neq \emptyset \), for all \( p \in \Pi \), and \( f_p \) unspecified. Then if \( S \) is determinate for all \( f_p \), it is mutually non-interfering.

Proofs

The following lemma is helpful:

**Lemma:** Let \( S \) be a mutually noninterfering system. Let \( p \) be a terminal process of \( S \). If \( \alpha = \beta p \gamma \delta \) is an execution sequence of \( S \), then \( \alpha' = \beta' \gamma' \delta \) is an execution sequence of \( S \) for which \( V(M_i, \alpha) = V(M_i, \alpha') \) for all \( i \).

**Proof:** As \( p \) is a terminal process in \( S \), it has no successors in \( S \). Hence \( \alpha' \) satisfies the precedence constraints of \( S \). So \( \alpha' \) is an execution sequence. We now consider two cases.

1. \( M_i \notin \text{range}(p) \). Note \( p \) does not write memory locations not in \( \text{range}(p) \). Consider any process \( p' \) with \( \overline{p'} \) in \( \delta \). As \( p \) and \( p' \) are mutually noninterfering, \( \text{range}(p) \cap \text{domain}(p') = \emptyset \). So all such \( p' \) find the same values in \( \text{domain}(p') \) whether the execution sequence is \( \alpha \) or \( \alpha' \). Thus, \( V(M_i, \alpha) = V(M_i, \alpha') \).

2. \( M_i \in \text{range}(p) \). Let \( \overline{p'} \) in \( \gamma \delta \). As \( p \) and \( p' \) are mutually noninterfering, \( \text{domain}(p) \cap \text{range}(p') = \emptyset \). So no \( p' \) in \( \gamma \delta \) writes into an element of \( \text{domain}(p) \). Hence for all \( M_j \in \text{domain}(p) \), \( V(M_j, \beta) = V(M_j, \beta' \gamma \delta) \). By definition, for all \( M_j \in \text{domain}(p) \), \( F(M_j, \beta) = F(M_j, \beta' \gamma \delta) \). As \( p \) has the same input for both \( \alpha \) and \( \alpha' \), it writes the same value into
each $M_i \in \text{range}(p)$ in $\alpha$ and $\alpha'$. Let $v$ denote the value that $p$ writes into $M_i$ in $\alpha$. Then
\[
V(M_i, \alpha) = V(M_i, \overline{p_{\gamma}(\delta)}) \quad \text{as no process } p' \text{ in } \delta \text{ writes into an element of } \text{range}(p)
= (V(M_i, \overline{p_{\gamma}}), v) \quad \text{as } p \text{ writes } v \text{ into } M_i
= (V(M_i, \overline{p}), v) \quad \text{as no process } p' \text{ in } \gamma \text{ writes into an element of } \text{range}(p)
= (V(M_i, b\overline{\delta}), v) \quad \text{as no process } p' \text{ in } \gamma \text{ writes into an element of } \text{range}(p)
= (V(M_i, \overline{p_{\gamma}p}), v) \quad \text{as } p \text{ writes } v \text{ into } M_i
= V(M_i, \alpha')
\]
This proves the lemma.

**Proof of Theorem 1:** We proceed by induction on the number $k$ of processes in a system.

*Basis:* $k = 1$. The claim is trivially true.

*Hypothesis:* For $k = 1, \ldots, n-1$, if a system of $k$ processes is mutually noninterfering, it is determinate.

*Step:* Let $S$ be an $n$ process system of mutually noninterfering processes.

If $S$ has exactly one execution sequence, it is determinate. So, assume that $S$ has two distinct execution sequences $\alpha$ and $\beta$.

Let $p$ be a terminal process of $S$, and form $\alpha'$ and $\beta'$ according to the lemma. Then
\[
\alpha' = \alpha'_{\overline{p}} \quad V(M_i, \alpha) = V(M_i, \alpha') \quad \text{for all } i \text{ such that } 1 \leq i \leq m
\]
\[
\beta' = \beta'_{\overline{p}} \quad V(M_i, \beta) = V(M_i, \beta') \quad \text{for all } i \text{ such that } 1 \leq i \leq m
\]
Now form the $n-1$ process system $S' = (\prod - \{ p \}, \rightarrow)$, where $\rightarrow$ is formed by deleting from $\rightarrow$ all pairs with $p$ in them. Clearly, $\alpha''$ and $\beta''$ are execution sequences of $S'$. Further, by the induction hypothesis, $V(M_i, \alpha'') = V(M_i, \beta'')$ for all $i$ such that $1 \leq i \leq m$. This means that the values in the elements of $\text{domain}(p)$ are the same in both $\alpha''$ and $\beta''$; in other words, $F(M_j, \alpha'') = F(M_j, \beta'')$ for all $M_j \in \text{domain}(p)$. As the inputs for $p$ are the same in both execution sequences, the outputs will also be the same. It follows that $p$ writes the same value $v$ into $M_i \in \text{range}(p)$ in both $\alpha'$ and $\beta'$.

Hence for $M_i \not\in \text{range}(p)$:
\[
V(M_i, \alpha) = V(M_i, \alpha') \quad \text{by the lemma}
= V(M_i, \alpha'') \quad \text{as } M_i \not\in \text{range}(p)
= V(M_i, \beta'') \quad \text{by the induction hypothesis}
= V(M_i, \beta') \quad \text{as } M_i \not\in \text{range}(p)
= V(M_i, \beta) \quad \text{by the lemma}
\]
and for $M_i \in \text{range}(p)$:
\[
V(M_i, \alpha) = V(M_i, \alpha') \quad \text{by the lemma}
= (V(M_i, \alpha''), v) \quad p \text{ writes } v \text{ into } M_i
= (V(M_i, \beta''), v) \quad \text{by the induction hypothesis}
= (V(M_i, \beta'), v) \quad p \text{ writes } v \text{ into } M_i
= V(M_i, \beta) \quad \text{by the lemma}
\]
Either way, $V(M_i, \alpha) = V(M_i, \beta)$. Hence $S$ is determinate, completing the induction step and the proof.

**Proof of Theorem 2:** We prove this theorem by contradiction. Let $S$ be a determinate system. Let $p, p' \in \prod$ be interfering processes. Then there exist execution sequences
\[
\alpha = \overline{p_{p_{\gamma}}p_{\gamma}p}
\alpha' = \overline{p_{p_{\gamma}}p_{\gamma}p}
\]
Consider the Bernstein conditions. As $p$ and $p'$ are interfering, at least one of those conditions does not hold. We examine them separately.

1. Let $M_i \in \text{range}(p) \cap \text{range}(p')$. We choose the interpretation $f_p$ so that $p$ writes the value $u$ into $M_i$, and we choose the interpretation $f_{p'}$ so that $p'$ writes the value $v$ into $M_i$ and $u \neq v$. But then
\[
V(M_i, \overline{p_{\gamma}p_{\gamma}p}) = (V(M_i, \overline{p}), u, v)
\]
and
\[
V(M_i, \overline{p_{\gamma}p_{\gamma}p}) = (V(M_i, \overline{p}), v, u).
\]
This means $S$ is not determinate, contradicting hypothesis. So $\text{range}(p) \cap \text{range}(p') = \emptyset$.

2. Let $M_i \in \text{domain}(p) \cap \text{range}(p')$. As $\text{range}(p) \neq \emptyset$, take $M_i \in \text{range}(p)$. Choose the interpretation $f_p$, so that $p$
reads different values in $\alpha$ and $\alpha'$; that is, $F(M_j, \beta) \neq F(M_j, \beta \overrightarrow{p} p')$ for some $j$ such that $1 \leq j \leq m$. Also, choose $f_p$ so that $p$ writes $u$ in $\alpha$ and $v$ in $\alpha'$, where $u \neq v$. But then

$V(M_i, \beta \overrightarrow{p} p') = V(M_i, \beta \overrightarrow{p} p)$

as $\text{range}(p) \cap \text{range}(p') = \emptyset$

$V(M_i, \beta \overrightarrow{p} p) = (V(M_i, \beta), u)$

As $u \neq v$, this means that $S$ is not determinate, contradicting hypothesis. So $\text{domain}(p) \cap \text{range}(p') = \emptyset$. [As an aside, if $\text{range}(p) = \emptyset$, then $M_j \notin \text{range}(p)$ and $p$ and $p'$ are noninterfering. Hence there is no contradiction.]

3. By symmetry, the argument for case 2 also shows that $\text{range}(p) \cap \text{domain}(p') = \emptyset$.

In all three cases, the Bernstein conditions must hold. This completes the proof.  

$\square$