Ch. 3 General Principles

- Discrete-event simulation

- The basic building blocks of all discrete-event simulation models

  : entities and attributes, activities and events.

- A system is modeled in terms of
  - its state at each point in time
  - the entities that pass through the system and the entities that represent system resources
  - the activities and events that cause system state to change.

- Discrete-event models are appropriate for those systems for which changes in system state occur only at discrete points in time.

- This chapter deals exclusively with dynamic, stochastic systems (i.e., involving time and containing random elements) which change in a discrete manner.

3.1Concepts in Discrete-Event Simulation (1)

- System : A collection of entities (e.g., people and machines) that interact together over time to accomplish one or more goals.
- Model : An abstract representation of a system, usually containing structural, logical, or mathematical relationships which describe a system in terms of state, entities and their attributes, sets, processes, events, activities, and delays.
- System state : A collection of variables that contain all the information necessary to describe the system at any time.
- Entity : Any object or component in the system which requires explicit representation in the model (e.g., a server, a customer, a machine).
- Attributes : The properties of a given entity (e.g., the priority of a waiting customer, the routing of a job through a job shop).

3.1Concepts in Discrete-Event Simulation (2)

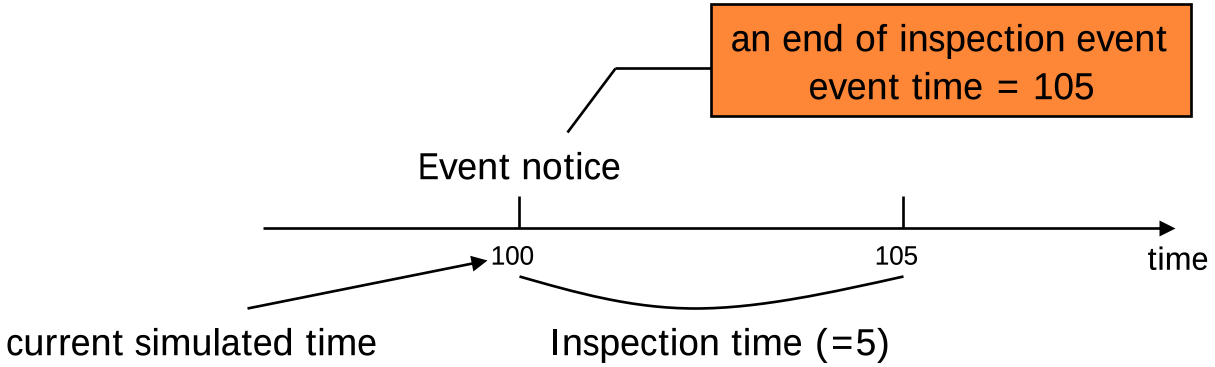
- List : A collection of (permanently or temporarily) associated entities, ordered in some logical fashion (such as all customers currently in a waiting line, ordered by first come, first served, or by priority).
- Event : An instantaneous occurrence that changes the state of a system (such as an arrival of a new customer).
- Event notice : A record of an event to occur at the current or some future time, along with any associated data necessary to execute the event; at a minimum, the record includes the event type and the event time.
- Event list : A list of event notices for future events, ordered by time of occurrence also known as the future event list (FEL).
- Activity : A duration of time of specified length (e.g., a service time or interarrival time), which is known when it begins (although it may be defined in terms of a statistical distribution).

3.1Concepts in Discrete-Event Simulation (3)

- Delay : A duration of time of unspecified indefinite length, which is not known until it ends (e.g., a customer's delay in a last-in, first-out waiting line which, when it begins, depends on future arrivals).
- Clock : A variable representing simulated time, called CLOCK in the examples to follow.
- An activity typically represents a service time, an interarrival time, or any other processing time whose duration has been characterized and defined by the modeler.

- An activity's duration may be specified in a number of ways:
  - 1. Deterministic-for example, always exactly 5 minutes;
  - 2. Statistical-for example, as a random draw from among 2, 5, 7 with equal probabilities;
  - 3. A function depending on system variables and/or entity attributes-for example, loading time for an iron ore ship as a function of the ship's allowed cargo weight and the loading rate in tons per hour.

3.1Concepts in Discrete-Event Simulation (4)

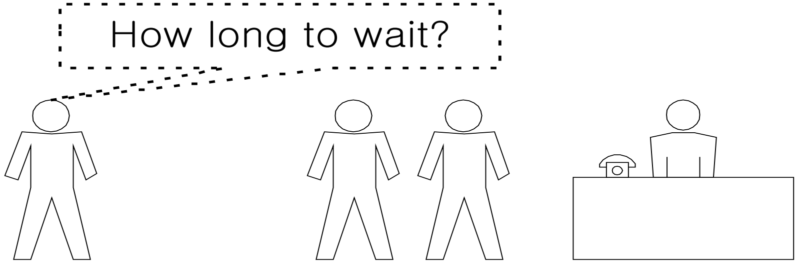


- The duration of an activity is computable from its specification at the instant it begins.

- To keep track of activities and their expected completion time, at the simulated instant that an activity duration begins, an event notice is created having an event time equal to the activity's completion time.

3.1Concepts in Discrete-Event Simulation (5)

- A delay's duration
  - Not specified by the modeler ahead of time, But rather determined by system conditions.
  - Quite often, a delay's duration is measured and is one of the desired outputs of a model run.



- A customer's delay in a waiting line may be dependent on the number and duration of service of other customers ahead in line as well as the availability of servers and equipment.

3.1Concepts in Discrete-Event Simulation (7)

- EXAMPLE 3.1 (Able and Baker, Revisited)
  - Consider the Able-Baker carhop system of Example 2.2.
  - System state
    - $L_Q(t)$: the number of cars waiting to be served at time t
    - $L_A(t)$: 0 or 1 to indicate Able being idle or busy at time t
    - $L_B(t)$: 0 or 1 to indicate Baker being idle or busy at time t
  - Entities : Neither the customers (i.e., cars) nor the servers need to be explicitly represented, except in terms of the state variables, unless certain customer averages are desired (compare Examples 3.4 and 3.5)
  - Events
    - Arrival event
    - Service completion by Able
    - Service completion by Baker

3.1Concepts in Discrete-Event Simulation (8)

- EXAMPLE 3.1 (Cont.)
  - Activities
    - Interarrival time, defined in Table 2.11
    - Service time by Able, defined in Table 2.12
    - Service time by Baker, defined in Table 2.13
  - Delay : A customer's wait in queue until Able or Baker becomes free

- The definition of the model components provides a static description of the model.

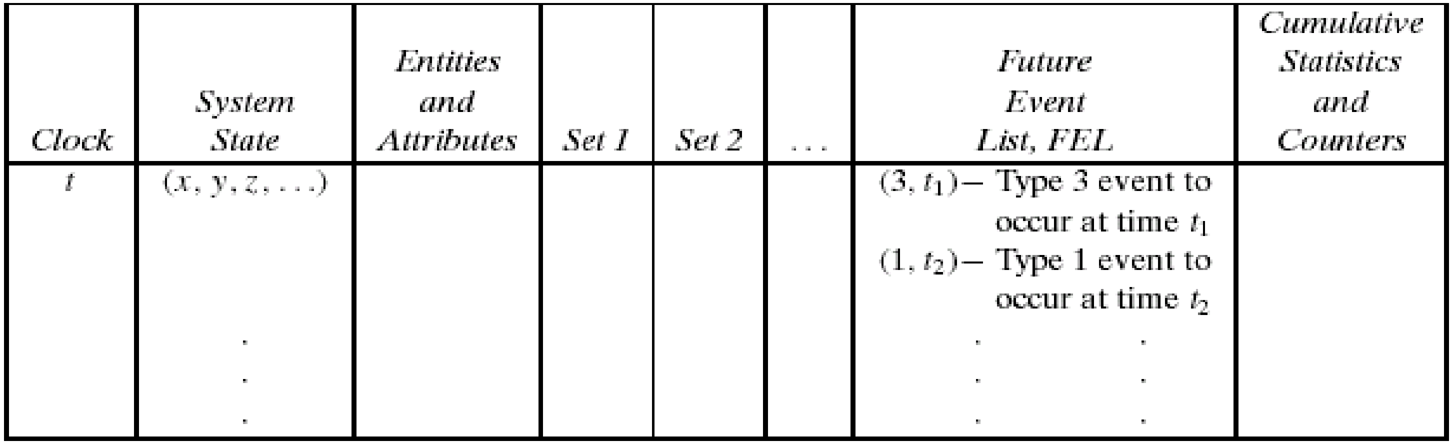
- A description of the dynamic relationships and interactions between the components is also needed.

3.1Concepts in Discrete-Event Simulation (9)

- A discrete-event simulation

  : the modeling over time of a system all of whose state changes occur at discrete points in time-those points when an event occurs.

- A discrete-event simulation proceeds by producing a sequence of system snapshots (or system images) which represent the evolution of the system through time.



Clock	*System State*	*Entities and Attributes*	*Set 1*	*Set 2*	...	*Future Event List, FEL*	*Cumulative Statistics and Counters*
t	$(x, y, z, \ldots)$ ⋮					$(3, t_1)$– Type 3 event to occur at time t_1 $(1, t_2)$– Type 1 event to occur at time t_2 ⋮ ⋮	

Figure 3.1 Prototype system snapshot at simulation time t

3.1.1. The Event-Scheduling/Time-Advanced Algorithm (1)

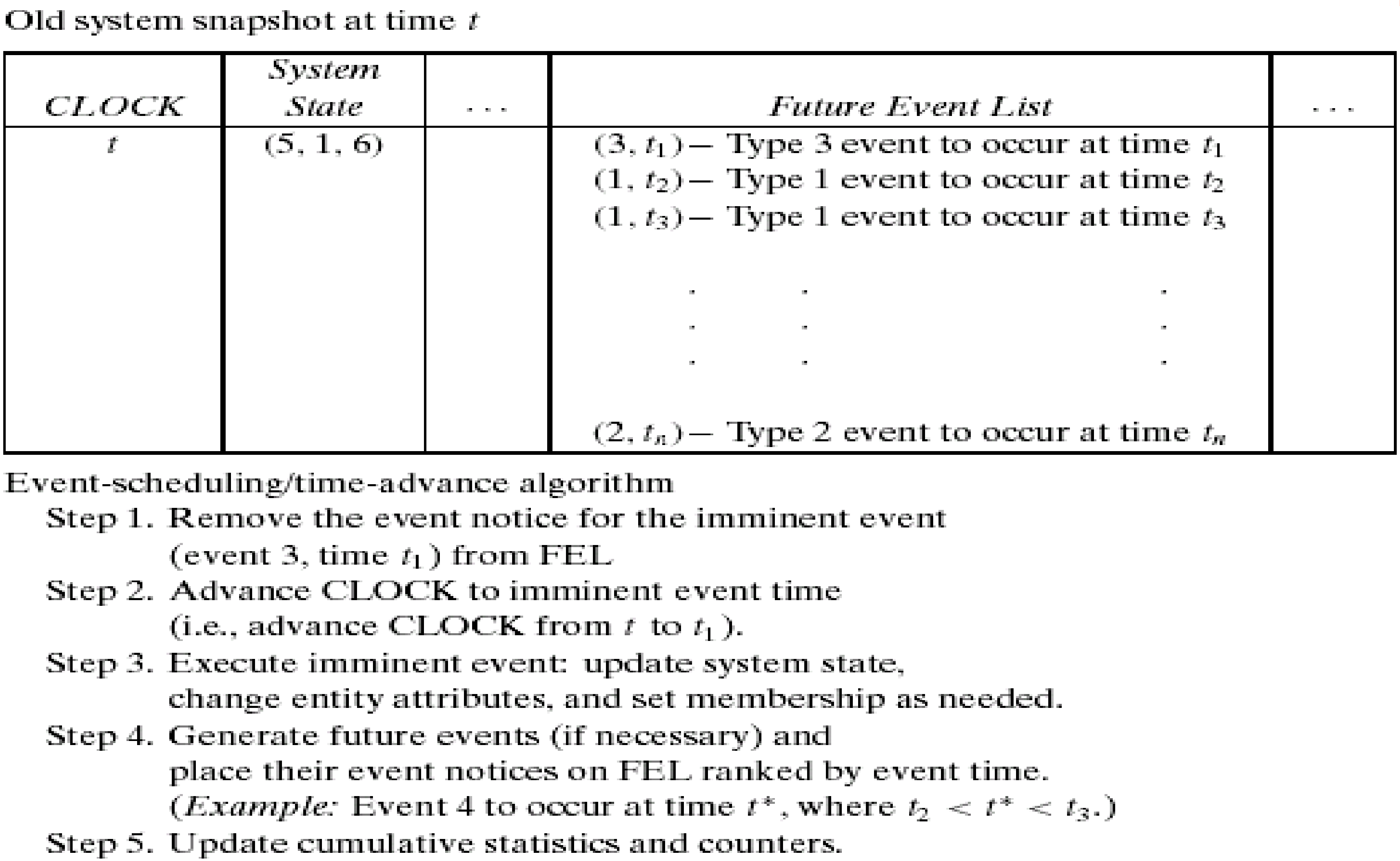
- The mechanism for advancing simulation time and guaranteeing that all events occur in correct chronological order is based on the future event list (FEL).
- Future Event List (FEL)
  - to contain all event notices for events that have been scheduled to occur at a future time.
  - to be ordered by event time, meaning that the events are arranged chronologically; that is, the event times satisfy

$$t < t_1 \leq t_2 \leq \quad \cdots \quad \leq t_n$$

current value of simulated time

Imminent event

- Scheduling a future event means that at the instant an activity begins, its duration is computed or drawn as a sample from a statistical distribution and the end-activity event, together with its event time, is placed on the future event list.



Old system snapshot at time t

CLOCK	*System State*	. . .	*Future Event List*	. . .
t	(5, 1, 6)		$(3, t_1)$— Type 3 event to occur at time t_1 $(1, t_2)$— Type 1 event to occur at time t_2 $(1, t_3)$— Type 1 event to occur at time t_3 . . . . $(2, t_n)$— Type 2 event to occur at time t_n	

Event-scheduling/time-advance algorithm

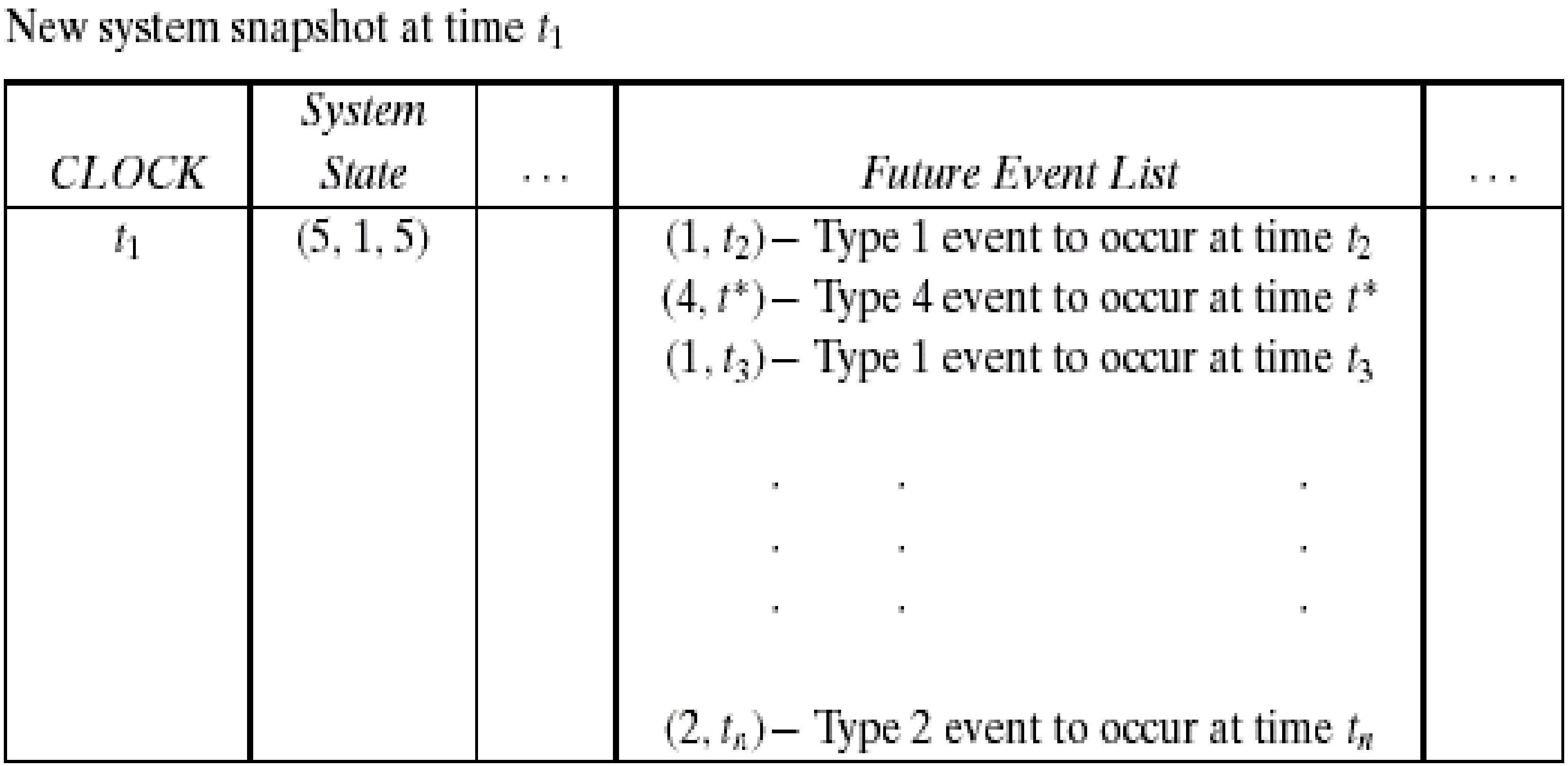
Step 1. Remove the event notice for the imminent event (event 3, time t_1) from FEL

Step 2. Advance CLOCK to imminent event time (i.e., advance CLOCK from t to t_1).

Step 3. Execute imminent event: update system state, change entity attributes, and set membership as needed.

Step 4. Generate future events (if necessary) and place their event notices on FEL ranked by event time. (*Example:* Event 4 to occur at time t^*, where $t_2 < t^* < t_3$.)

Step 5. Update cumulative statistics and counters.



New system snapshot at time t_1

CLOCK	*System State*	. . .	*Future Event List*	. . .
t_1	(5, 1, 5)		$(1, t_2)$– Type 1 event to occur at time t_2 $(4, t^*)$– Type 4 event to occur at time t^* $(1, t_3)$– Type 1 event to occur at time t_3 . . . . $(2, t_n)$– Type 2 event to occur at time t_n	

Figure 3.2 Advancing simulation time and updating system image.

3.1.1. The Event-Scheduling/Time-Advanced Algorithm (2)

- List processing : the management of a list .
  - the removal of the imminent event

    : As the imminent event is usually at the top of the list, its removal is as

    efficient as possible.
  - the addition of a new event to the list, and occasionally removal of some event (called cancellation of an event)

    : Addition of a new event (and cancellation of an old event) requires a

    search of the list.
- The efficiency of this search depends on the logical organization of the list and on how the search is conducted.
- The removal and addition of events from the FEL is illustrated in Figure 3.2.

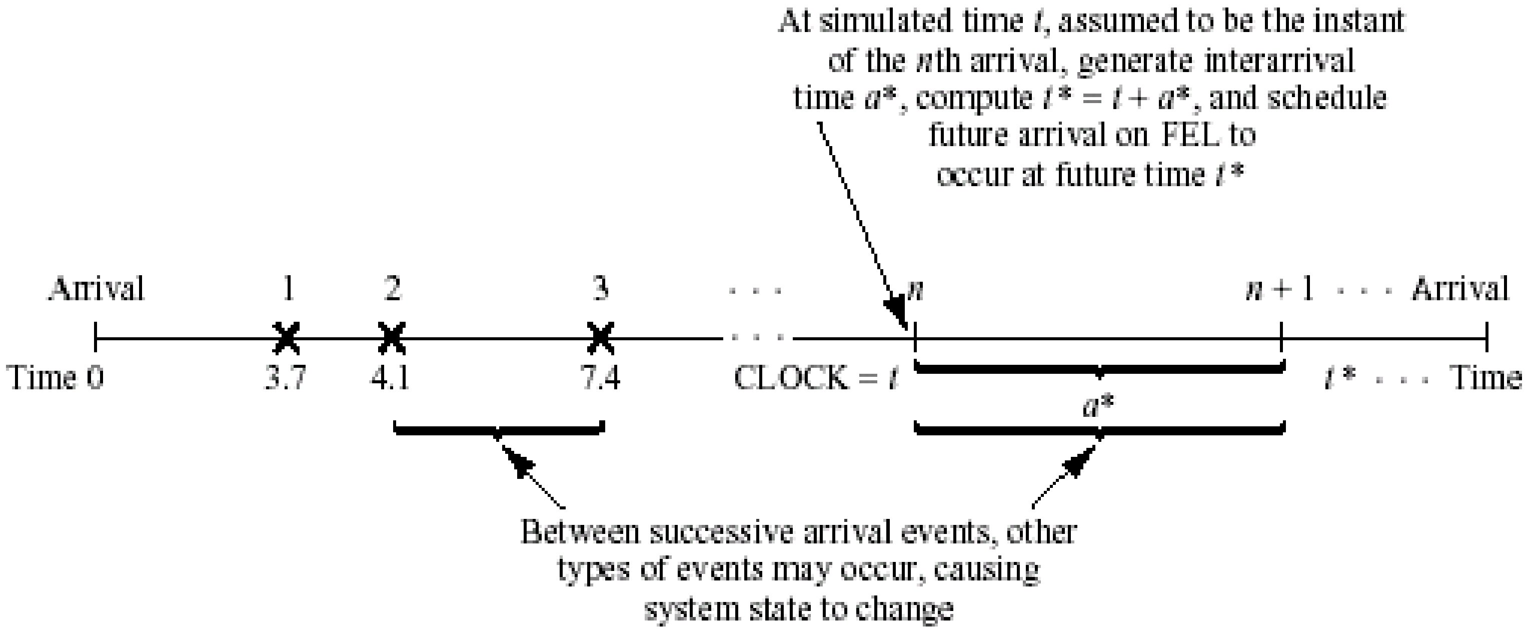
3.1.1. The Event-Scheduling/Time-Advanced Algorithm (3)

- The system snapshot at time 0 is defined by the initial conditions and the generation of the so-called exogenous events.
- An exogenous event : a happening "outside the system" which

  impinges on the system.
- The specified initial conditions define the system state at time 0.
  - In Figure 3.2, if t = 0, then the state (5, 1, 6) might represent the initial number of customers at three different points in the system.
- How future events are generated?
  - to generate an arrival to a queueing system
  - by a service-completion event in a queueing simulation
  - to generate runtimes and downtimes for a machine subject to breakdowns

3.1.1. The Event-Scheduling/Time-Advanced Algorithm (4)

- To generate an arrival to a queueing system
  - The end of an interarrival interval is an example of a primary event.


At simulated time *t*, assumed to be the instant of the *n*th arrival, generate interarrival time *a**, compute *t** = *t* + *a**, and schedule future arrival on FEL to occur at future time *t**

Between successive arrival events, other types of events may occur, causing system state to change

Figure 3.3 Generation of an external arrival stream by bootstrapping.

3.1.1. The Event-Scheduling/Time-Advanced Algorithm (5)

- By a service-completion event in a queueing simulation
  - A new service time, s^*, will be generated for the next customer.
    - When one customer completes service, at current time CLOCK = t
    - If the next customer is present

      ⇒ The next service-completion event will be scheduled to occur at future time $t^* = t + s^*$ by placing onto the FEL a new event notice of type service completion.
  - A service-completion event will be generated and scheduled at the time of an arrival event, provided that, upon arrival, there is at least one idle server in the server group.
  - Beginning service : a conditional event triggered only on the condition that a customer is present and a server is free.
  - Service completion : a primary event.
  - Service time : an activity

3.1.1. The Event-Scheduling/Time-Advanced Algorithm (6)

- By a service-completion event in a queueing simulation (Cont.)
  - A conditional event is triggered by a primary event occurring
  - Only primary events appear on the FEL.
- To generate runtimes and downtimes for a machine subject to breakdowns
  - At time 0, the first runtime will be generated and an end-of-runtime event scheduled.
  - Whenever an end-of-runtime event occurs, a downtime will be generated and an end-of-downtime event scheduled on the FEL.
  - When the CLOCK is eventually advanced to the time of this end-of-downtime event, a runtime is generated and an end-of-runtime event scheduled on the FEL.
  - An end of runtime and an end of downtime : primary events.
  - A runtime and a downtime : activities

3.1.1. The Event-Scheduling/Time-Advanced Algorithm (7)

- Every simulation must have a stopping event, here called E, which defines how long the simulation will run.

- There are generally two ways to stop a simulation:
  - 1. At time 0, schedule a stop simulation event at a specified future time T_E.
    Ex) Simulate a job shop for T_E = 40 hours,that is,over the time interval [0, 40].
  - 2. Run length T_E is determined by the simulation itself. Generally, T_E is the time of occurrence of some specified event E.
    Ex) the time of the 100th service completion at a certain service center.
    the time of breakdown of a complex system.
    the time of disengagement or total kill in a combat simulation.
    the time at which a distribution center ships the last carton in a day's orders.

- In case 2, T_E is not known ahead of time. Indeed, it may be one of the statistics of primary interest to be produced by the simulation.

3.1.2. World Views (2)

- The process-interaction approach (Cont.)
  - Figure 3.4 shows the interaction between two customer processes as customer n+1 is delayed until the previous customer's “end-service event” occurs.

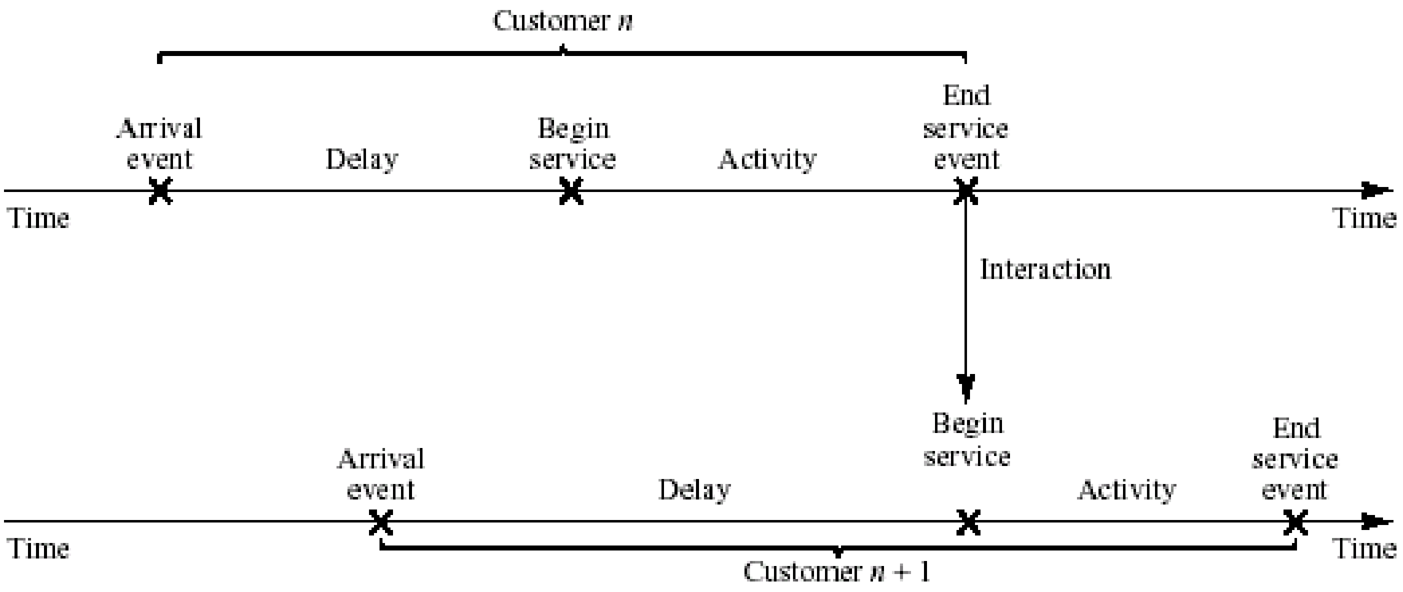


Figure 3.4 Two interacting customer processes in a single-server queue.

3.1.3. Manual Simulation Using Event Scheduling (1)

- Example 3.3 (Single-Channel Queue)
  - Reconsider Example 2.1
  - System state (LQ(t), LS(t)) :
    - LQ(t) is the number of customers in the waiting line
    - LS(t) is the number being served (0 or 1) at time t
  - Entities : The server and customers are not explicitly modeled, except in terms of the state variables above.
  - Events :
    - Arrival (A)
    - Departure (D)
    - Stopping event (E), scheduled to occur at time 60.

3.1.3. Manual Simulation Using Event Scheduling (2)

- Example 3.3 (Cont.)
  - Event notices (event type, event time) :
    - (A, t), representing an arrival event to occur at future time t
    - (D, t), representing a customer departure at future time t
    - (E, 60), representing the simulation-stop event at future time 60.
  - Activities :
    - Interarrival time, defined in Table 2.6
    - Service time, defined in Table 2.7
  - Delay : Customer time spent in waiting line.
  - The effect of the arrival and departure events was first shown in Figures 2.2 and 2.3 and is shown in more detail in Figures 3.5 and 3.6.

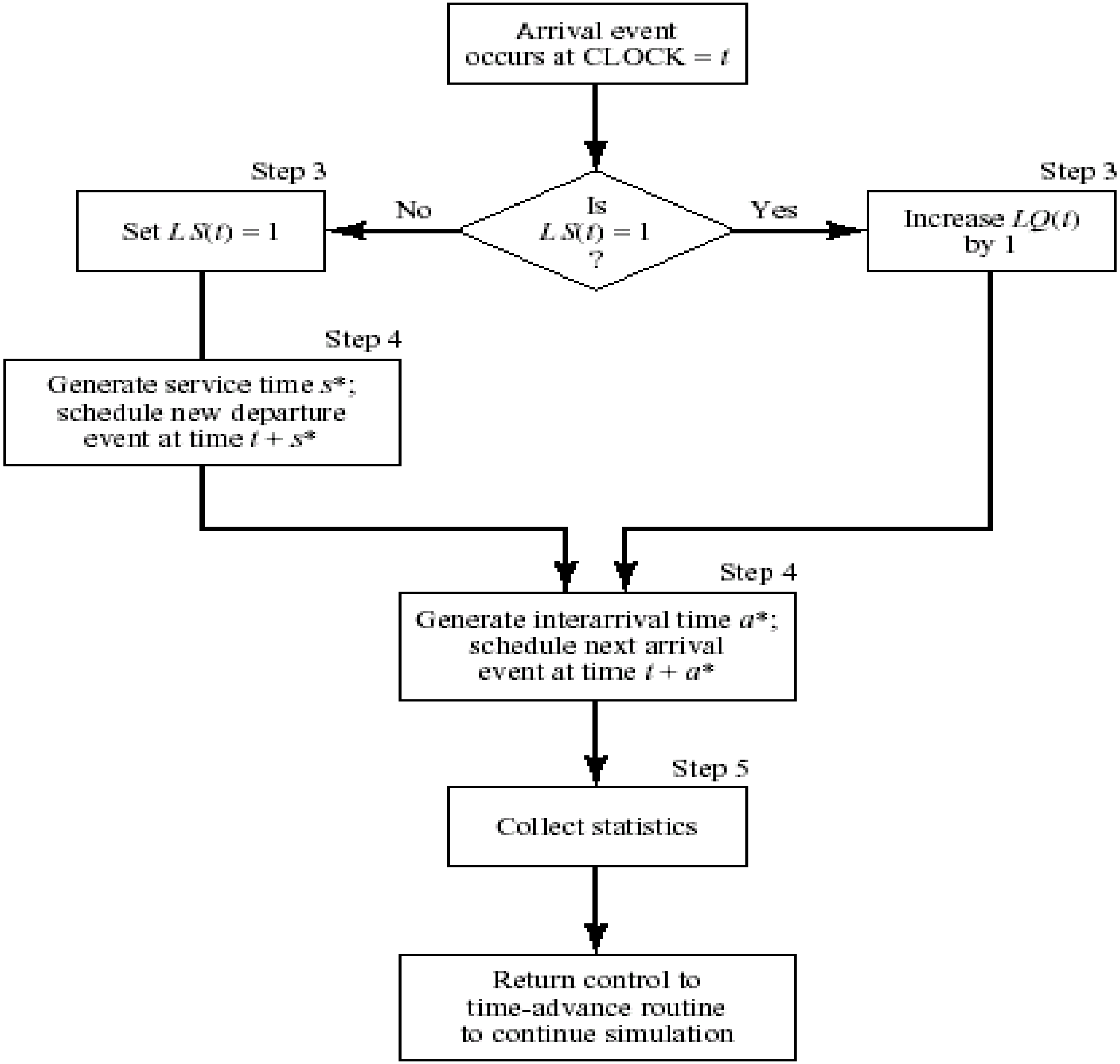

Arrival event occurs at CLOCK = t
Is $LS(t) = 1$?
No
Yes
Step 3
Set $LS(t) = 1$
Step 3
Increase $LQ(t)$ by 1
Step 4
Generate service time s^*; schedule new departure event at time $t + s^*$
Step 4
Generate interarrival time a^*; schedule next arrival event at time $t + a^*$
Step 5
Collect statistics
Return control to time-advance routine to continue simulation

Figure 3.5 Execution of the arrival event.

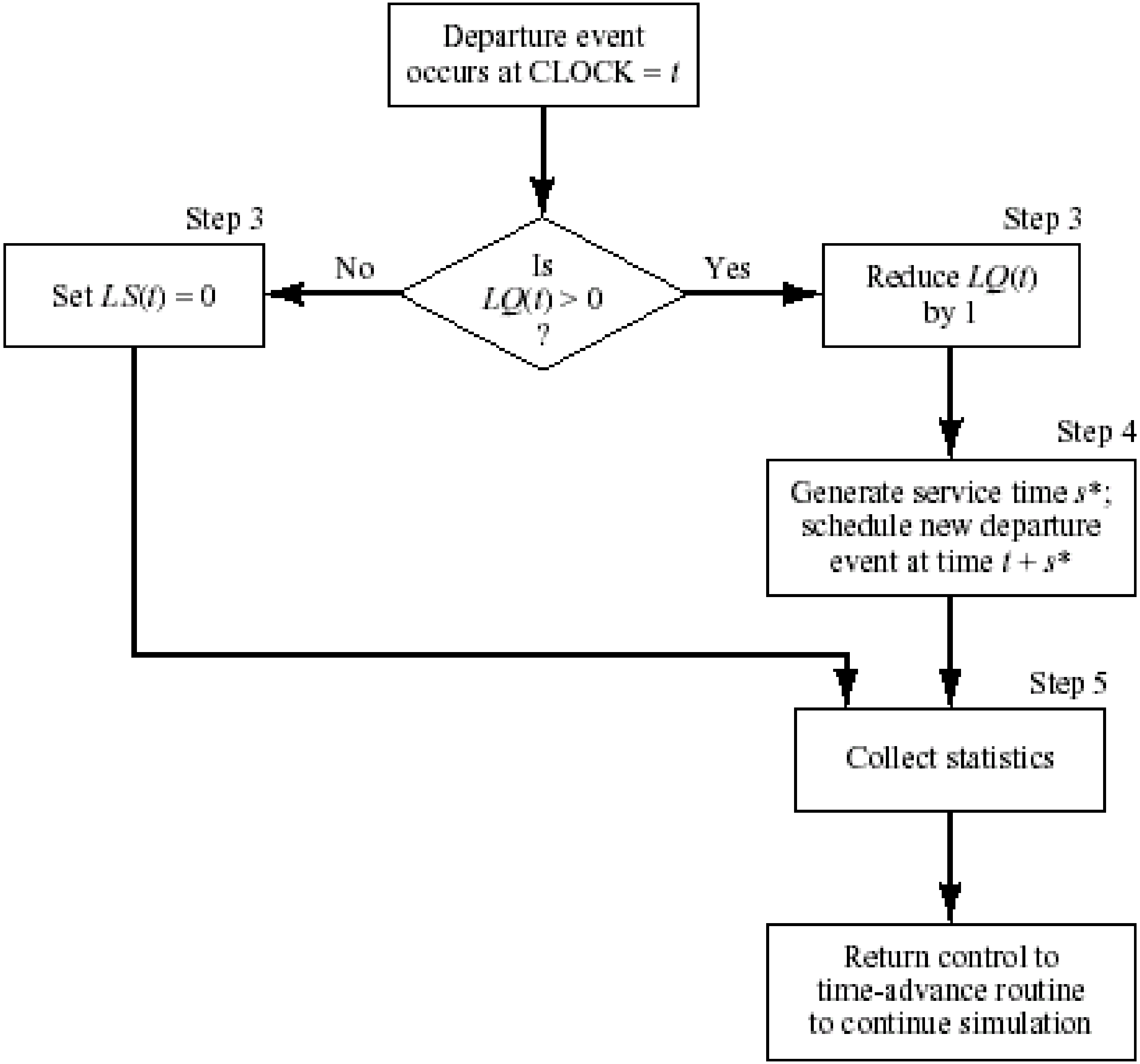
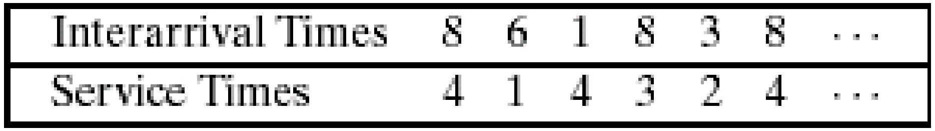

Departure event occurs at CLOCK = t
Step 3
Set $LS(t) = 0$
No
Is $LQ(t) > 0$?
Yes
Step 3
Reduce $LQ(t)$ by 1
Step 4
Generate service time s^*; schedule new departure event at time $t + s^*$
Step 5
Collect statistics
Return control to time-advance routine to continue simulation

Figure 3.6 Execution of the departure event.

3.1.3. Manual Simulation Using Event Scheduling (3)

- Example 3.3 (Cont.)
  - The interarrival times and service times will be identical to those used in Table 2.10

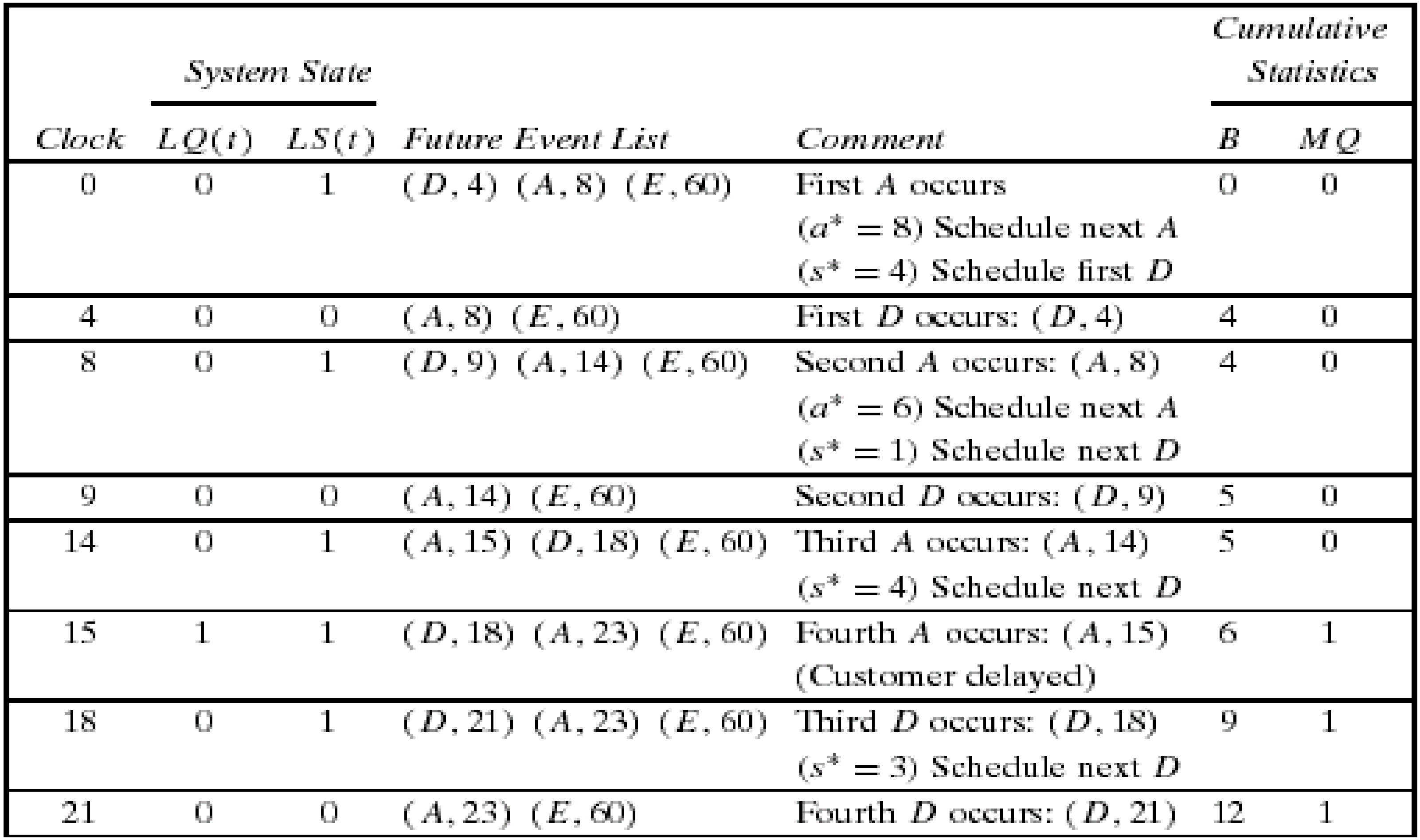


Interarrival Times	8	6	1	8	3	8	...
Service Times	4	1	4	3	2	4	...

  - Initial conditions
    - the system snapshot at time zero (CLOCK = 0)
    - LQ(0) = 0, LS(0) = 1
    - both a departure event and arrival event on the FEL.
  - The simulation is scheduled to stop at time 60.
  - Server utilization : total server busy time (B) / total time (T_E).
  - a^* : the generated interarrival time
  - s^* : the generated service times
  - The simulation in Table 3.1 covers the time interval [0, 21].

3.1.3. Manual Simulation Using Event Scheduling (4)

Table 3.1 Simulation Table for Checkout Counter (Example 3.3)



	System State				*Cumulative Statistics*	
Clock	$LQ(t)$	$LS(t)$	*Future Event List*	*Comment*	B	MQ
0	0	1	$(D,4)$ $(A,8)$ $(E,60)$	First A occurs $(a^* = 8)$ Schedule next A $(s^* = 4)$ Schedule first D	0	0
4	0	0	$(A,8)$ $(E,60)$	First D occurs: $(D,4)$	4	0
8	0	1	$(D,9)$ $(A,14)$ $(E,60)$	Second A occurs: $(A,8)$ $(a^* = 6)$ Schedule next A $(s^* = 1)$ Schedule next D	4	0
9	0	0	$(A,14)$ $(E,60)$	Second D occurs: $(D,9)$	5	0
14	0	1	$(A,15)$ $(D,18)$ $(E,60)$	Third A occurs: $(A,14)$ $(s^* = 4)$ Schedule next D	5	0
15	1	1	$(D,18)$ $(A,23)$ $(E,60)$	Fourth A occurs: $(A,15)$ (Customer delayed)	6	1
18	0	1	$(D,21)$ $(A,23)$ $(E,60)$	Third D occurs: $(D,18)$ $(s^* = 3)$ Schedule next D	9	1
21	0	0	$(A,23)$ $(E,60)$	Fourth D occurs: $(D,21)$	12	1

3.1.3. Manual Simulation Using Event Scheduling (5)

- Example 3.4 (The Checkout-Counter Simulation, Continued)
  - In Example 3.3, to estimate :
    - mean response time : the average length of time a customer spends in the system
    - mean proportion of customers who spend 4 or more minutes in the system.
  - Entities (Ci, t) : representing customer Ci who arrived at time t
  - Event notices :
    - (A, t, Ci), the arrival of customer Ci at future time t
    - (D, t, Cj), the departure of customer Cj at future time t
  - Set : “CHECKOUTLINE,” the set of all customers currently at the checkout counter (being served or waiting to be served), ordered by time of arrival
  - A customer entity with arrival time as an attribute is added in order to estimate mean response time.

3.1.3. Manual Simulation Using Event Scheduling (6)

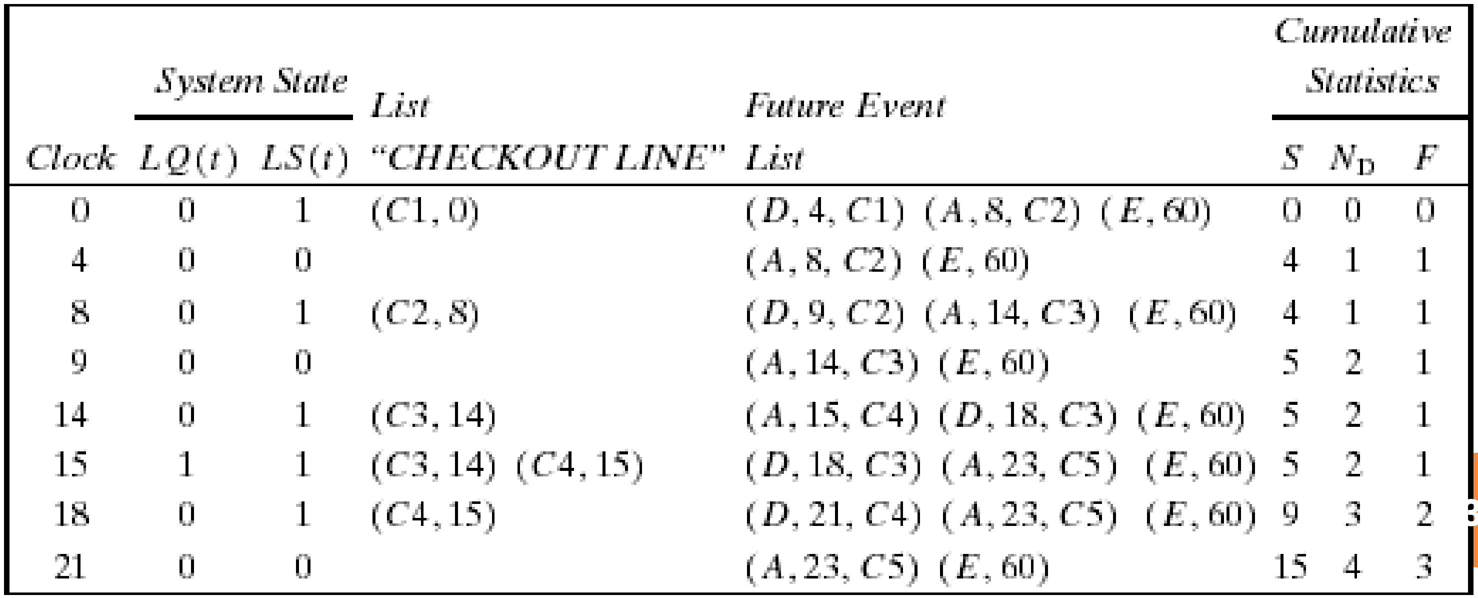
- Example 3.4 (Cont.)
  - Three new cumulative statistics will be collected :
    - S : the sum of customer response times for all customers who have departed by the current time
    - F : the total number of customers who spend 4 or more minutes at the checkout counter
    - N_D : the total number of departures up to the current simulation time.
  - These three cumulative statistics will be updated whenever the departure event occurs.
  - The simulation table for Example 3.4 is shown in Table 3.2.
  - The response time for customer is computed by

    Response time = CLOCK TIME - attribute “time of arrival”

3.1.3. Manual Simulation Using Event Scheduling (7)

- Example 3.4 (Cont.)
  - For a simulation run length of 21 minutes
    - the average response time was $S/N_D = 15/4 = 3.75$ minutes
    - the observed proportion of customers who spent 4 or more minutes in the system was $F/N_D = 0.75$.

Table 3.2 Simulation Table for Example 3.4



	System State		*List*	*Future Event*	*Cumulative Statistics*		
Clock	$LQ(t)$	$LS(t)$	*"CHECKOUT LINE"*	*List*	S	N_D	F
0	0	1	$(C1, 0)$	$(D, 4, C1)$ $(A, 8, C2)$ $(E, 60)$	0	0	0
4	0	0		$(A, 8, C2)$ $(E, 60)$	4	1	1
8	0	1	$(C2, 8)$	$(D, 9, C2)$ $(A, 14, C3)$ $(E, 60)$	4	1	1
9	0	0		$(A, 14, C3)$ $(E, 60)$	5	2	1
14	0	1	$(C3, 14)$	$(A, 15, C4)$ $(D, 18, C3)$ $(E, 60)$	5	2	1
15	1	1	$(C3, 14)$ $(C4, 15)$	$(D, 18, C3)$ $(A, 23, C5)$ $(E, 60)$	5	2	1
18	0	1	$(C4, 15)$	$(D, 21, C4)$ $(A, 23, C5)$ $(E, 60)$	9	3	2
21	0	0		$(A, 23, C5)$ $(E, 60)$	15	4	3