Ch4 Analyzing Workflows

1. Analysis Techniques
2. Reachability Analysis
3. Structural Analysis
4. Performance Analysis
5. Capacity Planning

4.1 Analysis Techniques

- qualitative analysis
  - mainly concern the *logical correctness* of the defined process, that is, the absence of anomalies such as "deadlocks" (when a case is "blocked" and no longer proceeds through the process) and "livelocks"
- quantitative analysis
  - mainly concern the *performance* of the defined process. An analysis of the quantitative aspects focuses upon establishing the performance indicators, such as average completion time, level of service, and utilization of capacity.

4.2 Reachability Analysis

- A Petri net and its initial state determines which states are reachable and in what order they can be reached.
- *reachability graph* is used to illustrate workflow behavior.

4.2 Reachability Analysis

- *non-deterministic choice*

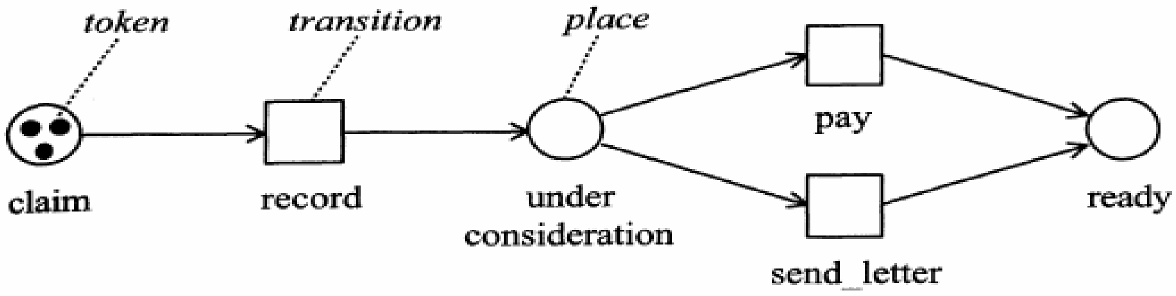


Figure 4.2
A classic Petri net

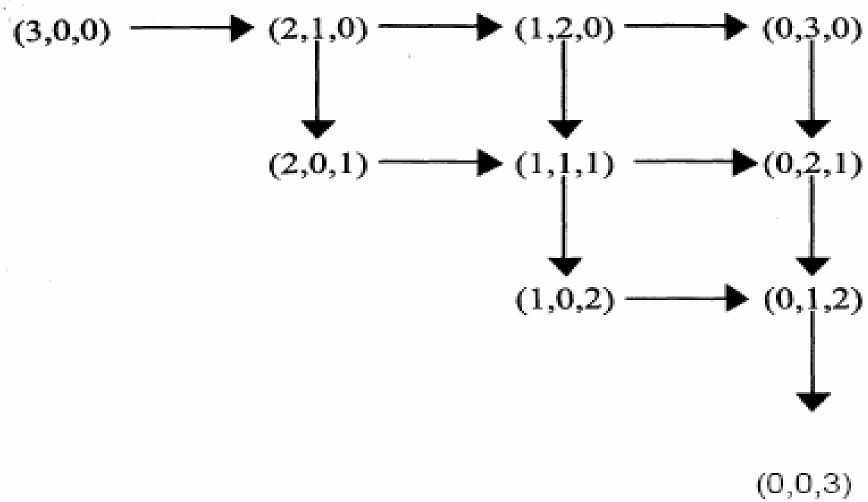


Figure 4.3
The reachability graph for the Petri net shown in figure 4.2

4.2 Reachability Analysis

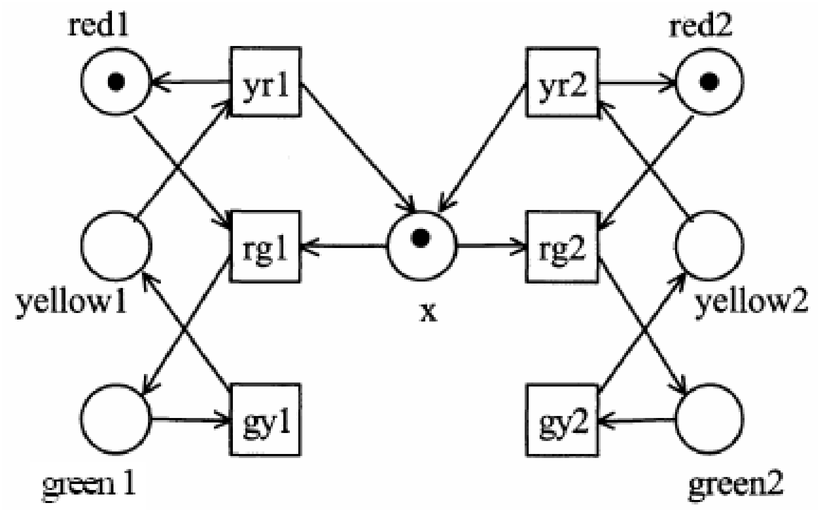


Figure 4.4
Two sets of traffic lights

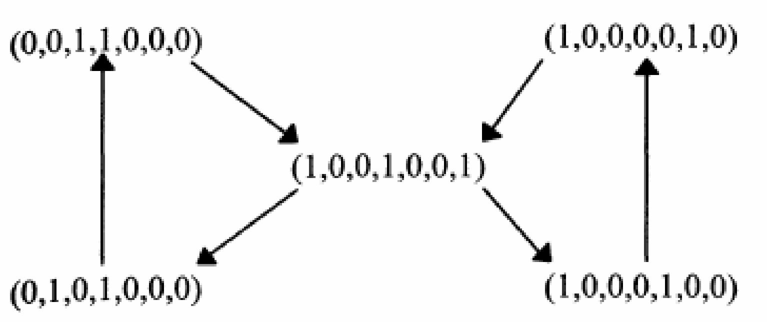


Figure 4.5
The reachability graph for the Petri net shown in figure 4.4

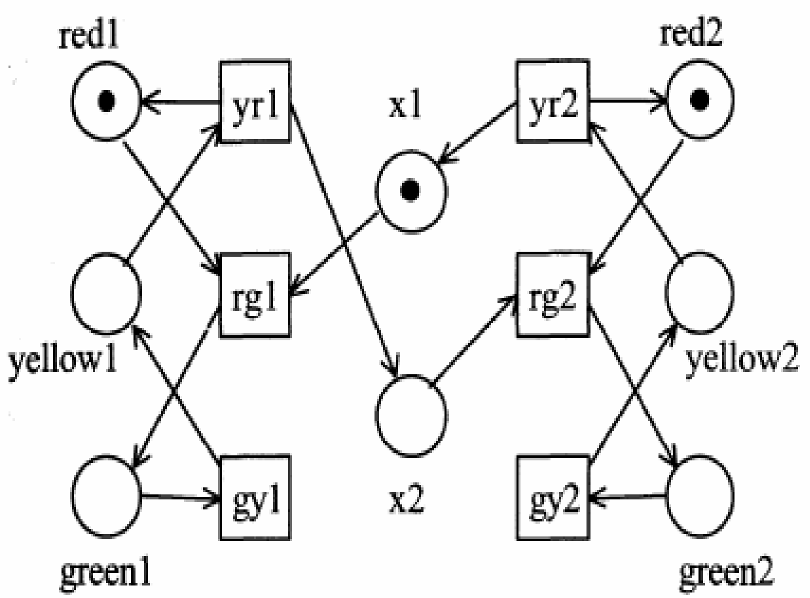


Figure 4.6
The two traffic lights now change to green alternately

4.3 Structural Analysis

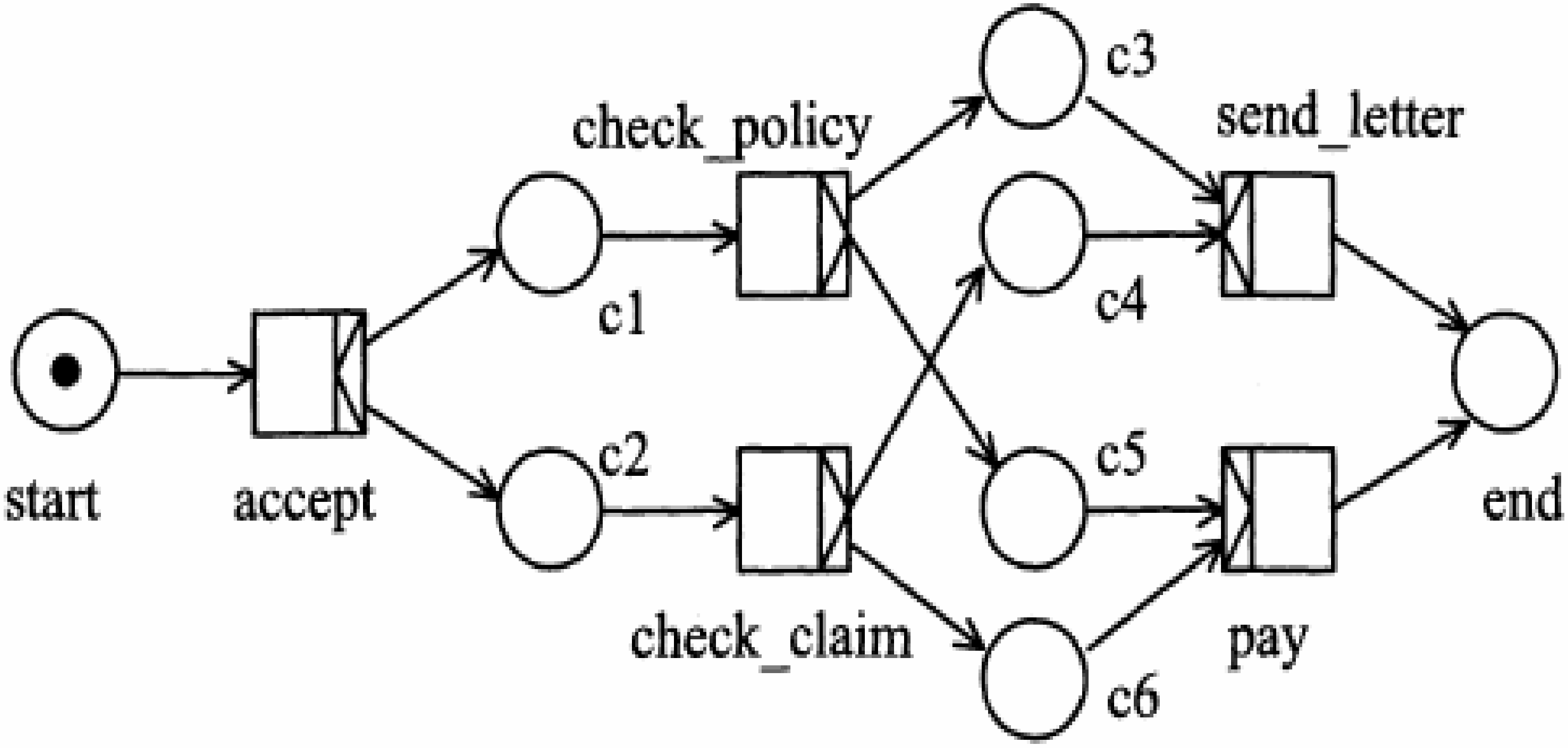


Figure 4.7
An example of an incorrect process

4.3 Structural Analysis

- four situations that can result in incorrect processes
- common *errors*
  - Tasks without input and/or output conditions.
  - Dead tasks: tasks that can never be carried out.
  - Deadlock: jamming a case before the condition "end" is reached.
  - Livelock: trapping a case in an endless cycle.
  - Activities still take place after the condition "end" is reached.
  - Tokens remain in the process definition after the case has been completed.

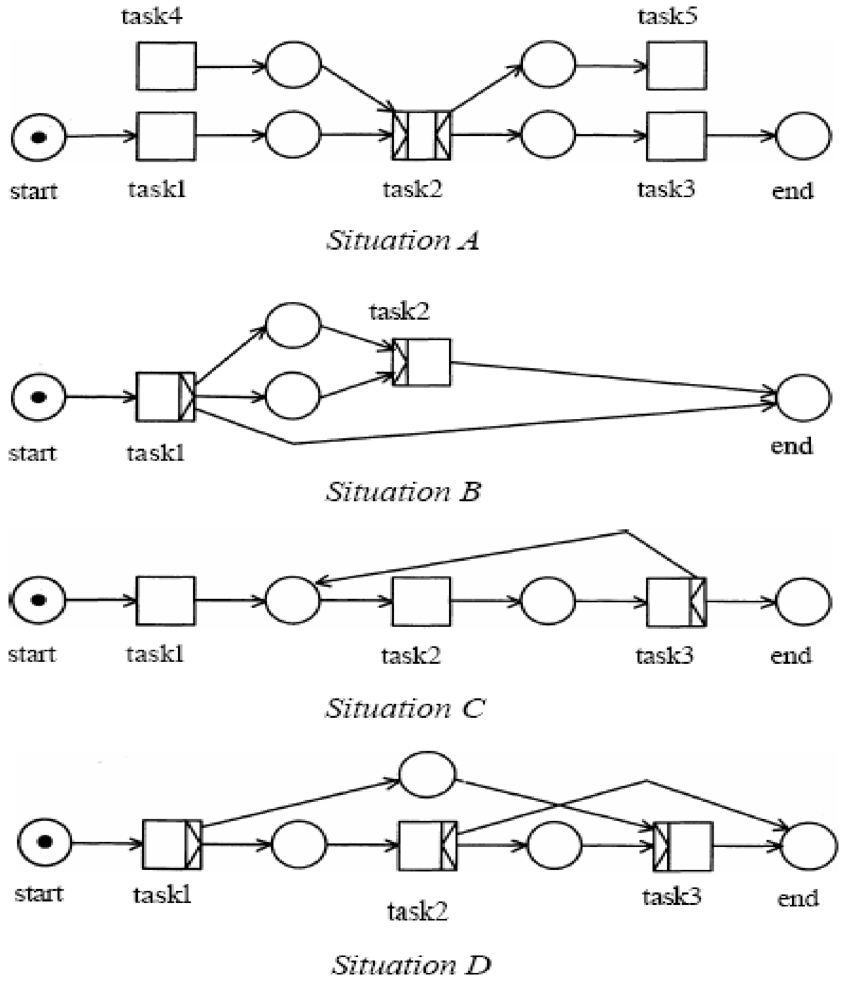


Figure 4.8 Four flawed situations

4.3 Structural Analysis

- Soundness of a process
  - A process contains no unnecessary tasks
  - every case submitted to the process must be completed in full and with no references to it (that is,case tokens) remaining in the process.

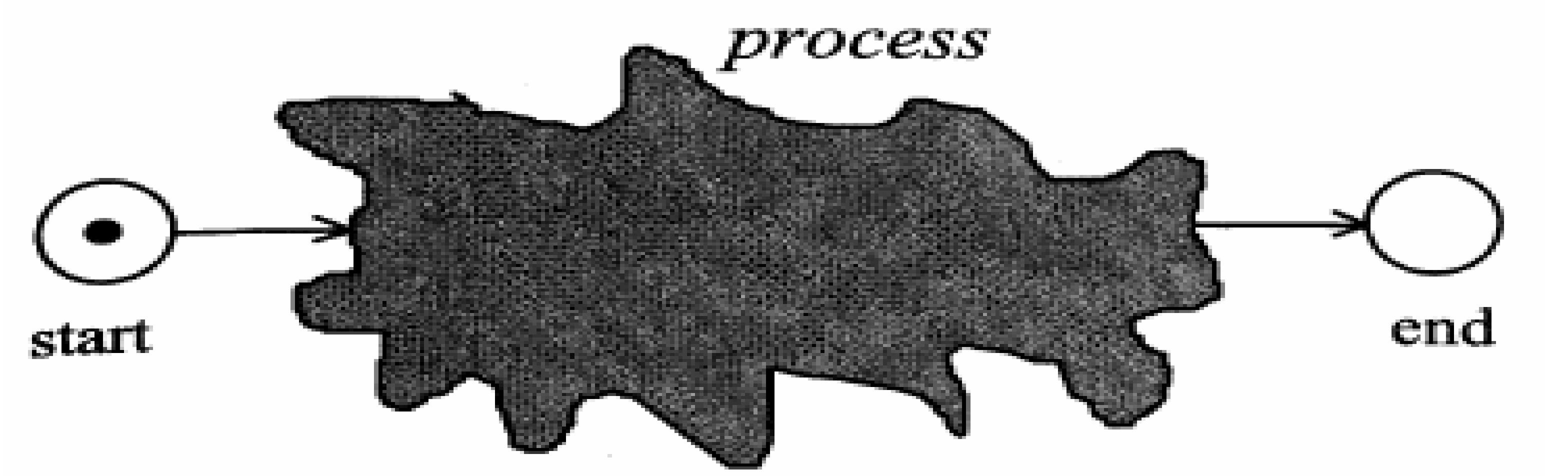


Figure 4.9
A process has one entrance and one exit

4.3 Structural Analysis

- workflow nets (WF-nets)
  - a Petri net
  - has a single input place start and a single output place end.
  - each transition (task) or place (condition) lies on a directed path from start to end.
- Soundness of WF-nets
  - For each token put in the place start, one (and only one) token eventually appears in the place end;
  - When the token appears in the place end, all the other places are empty
  - For each transition (task), it is possible to move from the initial state to a state in which that transition is enabled.
- Assumptions
  - if a task can potentially be executed, then it is not possible to postpone its execution indefinitely.
  - two tasks cannot "starve" a third task indefinitely.

4.3 Structural Analysis

- How can we establish whether a given process corresponds to a sound workflow net?
  - check whether the Petri net representing the process is a workflow net.
    - This can be checked by examining the structure of the process.
  - Check whether the process is sound
    - reachability graph: two drawbacks
    - coverability graph

4.3 Structural Analysis

- Method with computer support
  - short-circuited net: WF-net + t*
    - t*: end as its input point and start as its output point.
    - Liveness: A Petri net is live when it is possible to reach—for each transition t and from every state reachable from the initial one—a state in which transition t is enabled.
    - Boundedness: there is an upper limit to the number of tokens in each place.
  - A process is sound if its WF-net, with the additional transition t*, is live and bounded.
  - For a number of important subcategories—including the so-called free-choice Petri nets—liveness and boundedness of a network can be established in polynomial time.

4.3 Structural Analysis

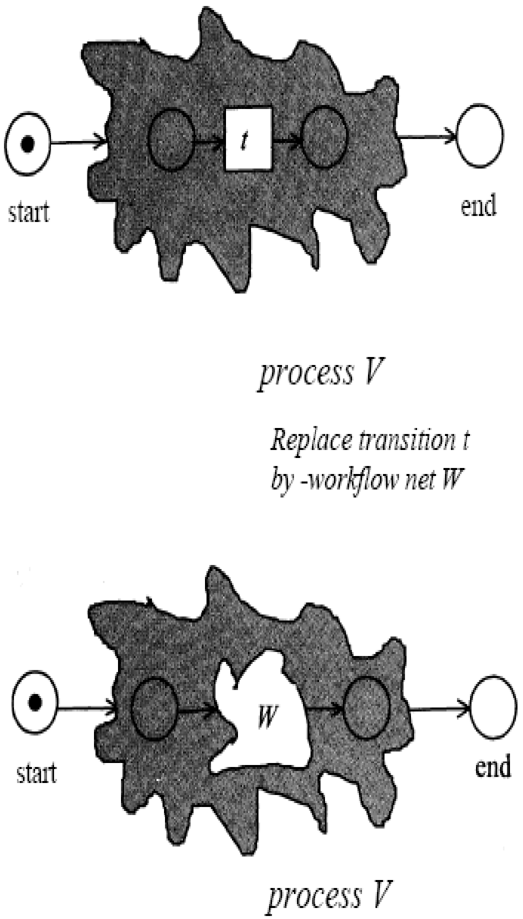


Figure 4.10
If a transition is replaced by a sound workflow net, then the resulting workflow net is also sound (assuming safeness)

- Method without computer support
  - a workflow net is safe, if the number of tokens in each place will never be larger than one.
  - If we have two sound and safe workflow nets V and W and we have a task t in V which has precisely one input and one output place, then we may replace task t in V by W and then the resulting workflow net is sound and safe again.

4.3 Structural Ana

- Suppose we have some set of sound and safe workflow nets, called "building blocks," to start with. If it is possible to derive the workflow net under consideration by a sequence of substitutions of nets from this set of building blocks, then we have proved that our net is sound and save as well.

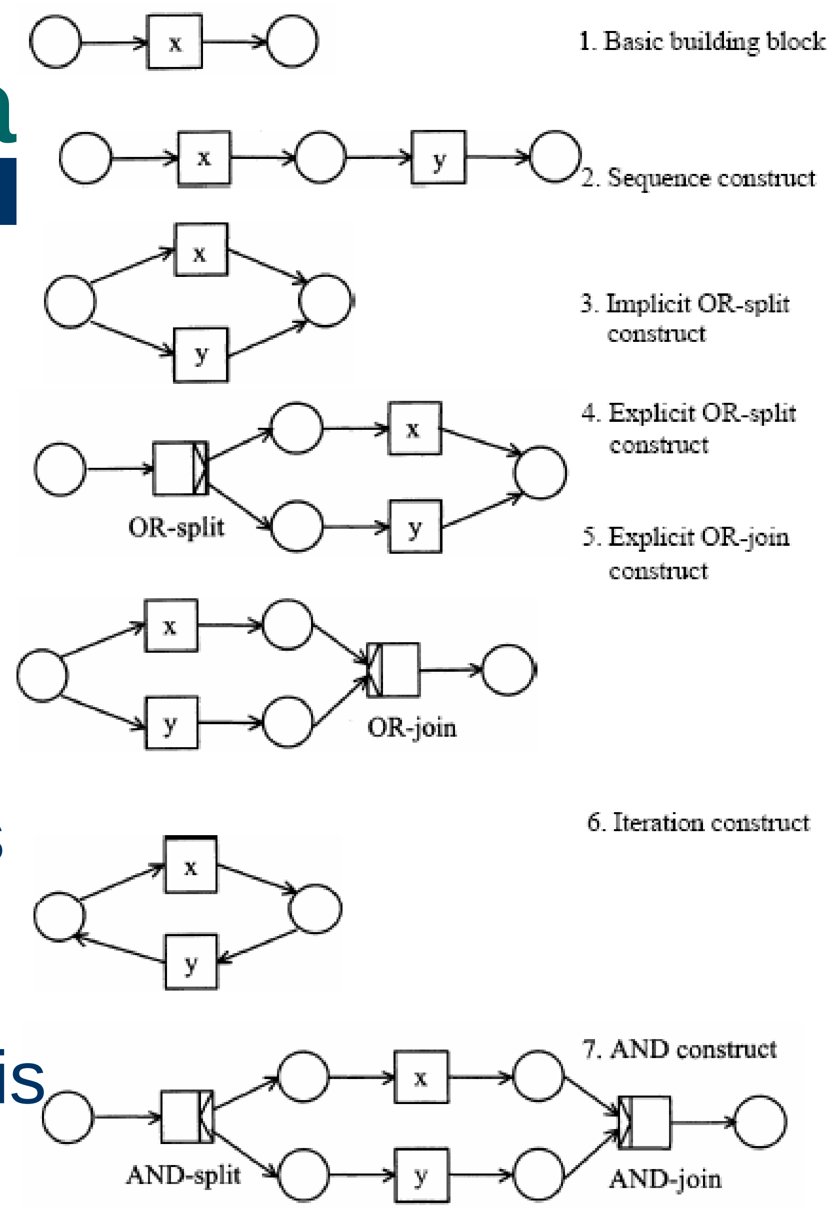


Figure 4.11
Sound and safe nets

4.3 Structural Analysis

- An example

Figure 4.12
A safe and sound process

Analyzing Workflows 113

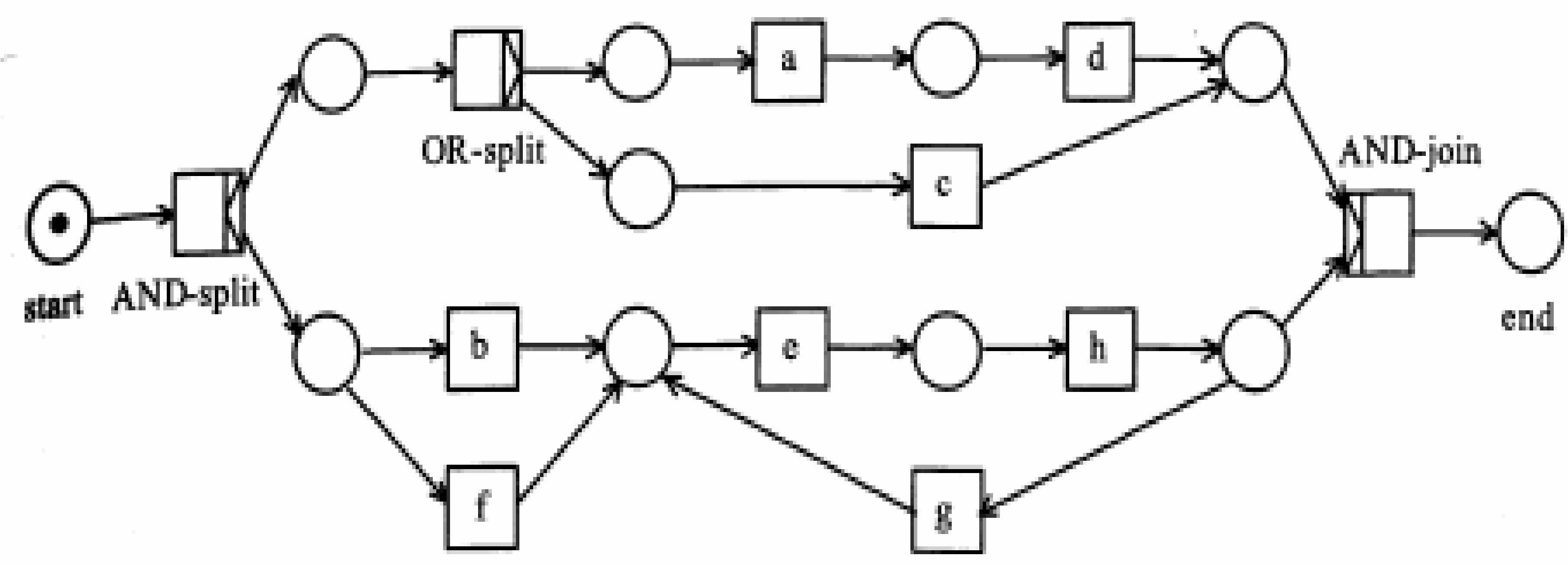


Figure 4.13
Apply the AND construct to a (Step 1)

4.3 Structural Analysis

Figure 4.13
Apply the AND construct to a (Step 1)

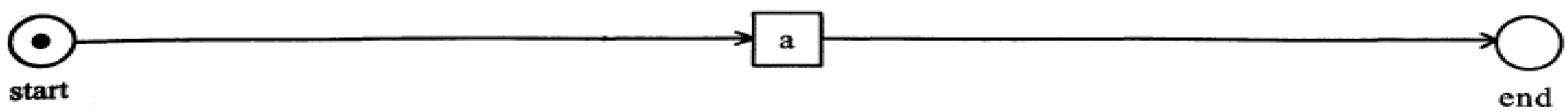


Figure 4.14
Apply the explicit OR-split construct to a (Step 2)

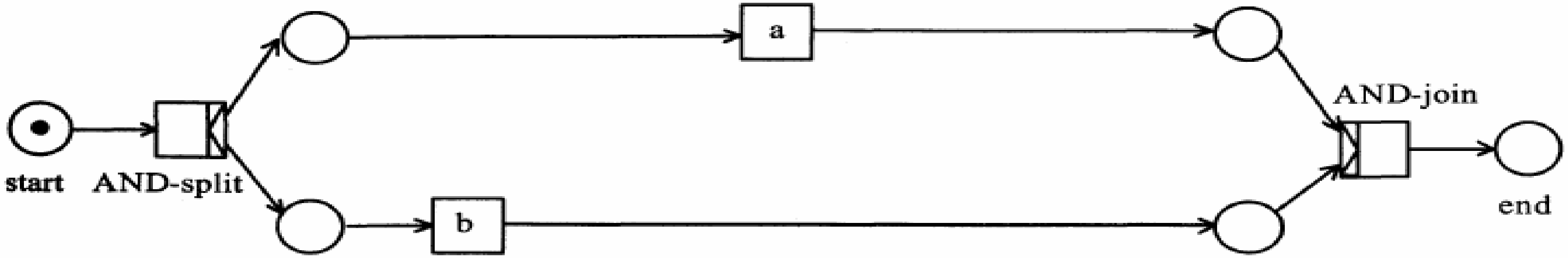
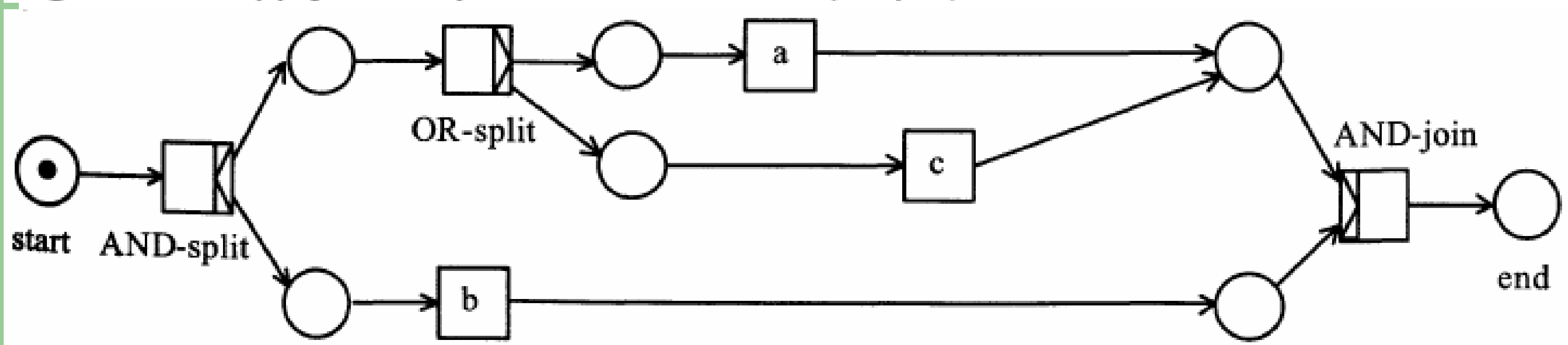


Figure 4.15 Apply the sequence construct to a (Step 3)



4.3 Structural Analysis

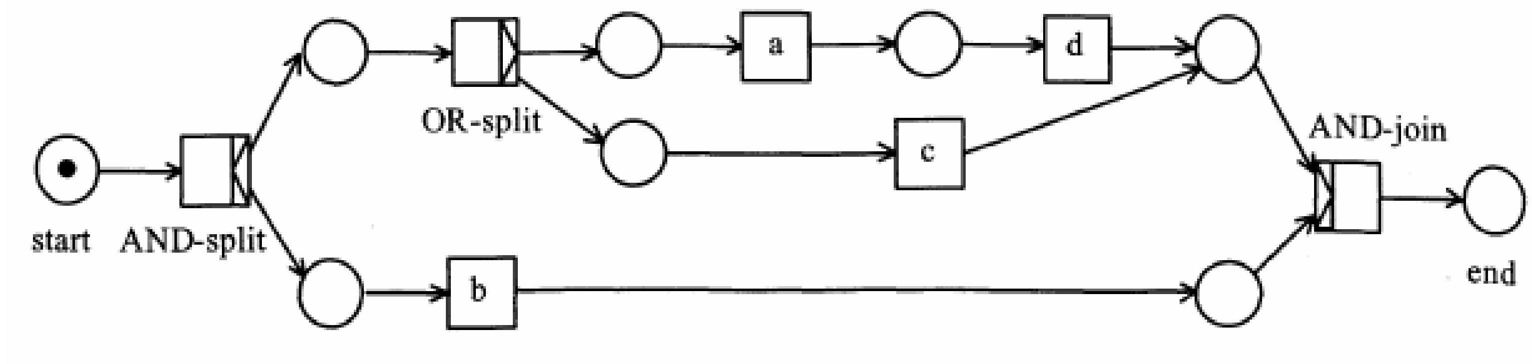


Figure 4.16
Apply the sequence construct to b (Step 4)

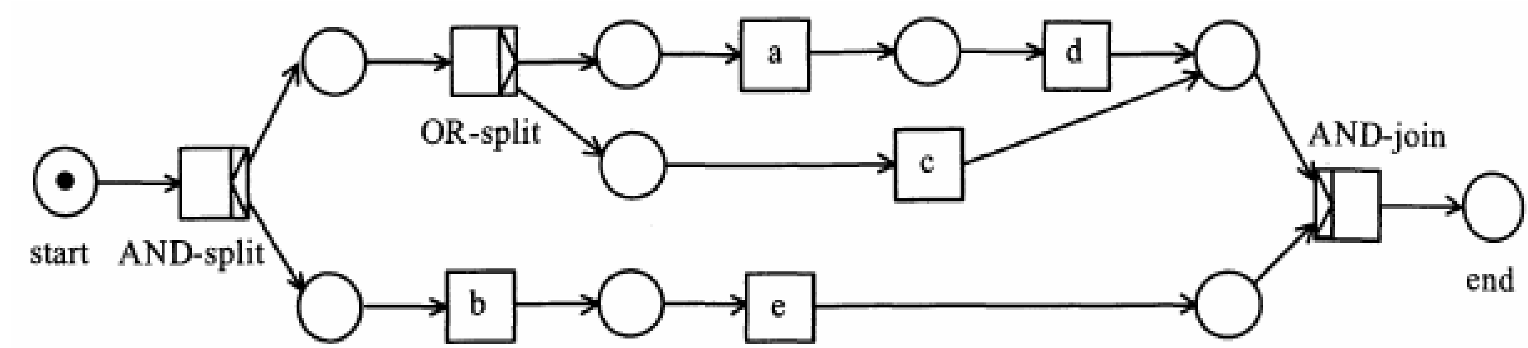


Figure 4.17
Apply the implicit OR-split construct to *b* (Step 5)

Figure 4.18
Apply the iteration construct to e (Step 6)

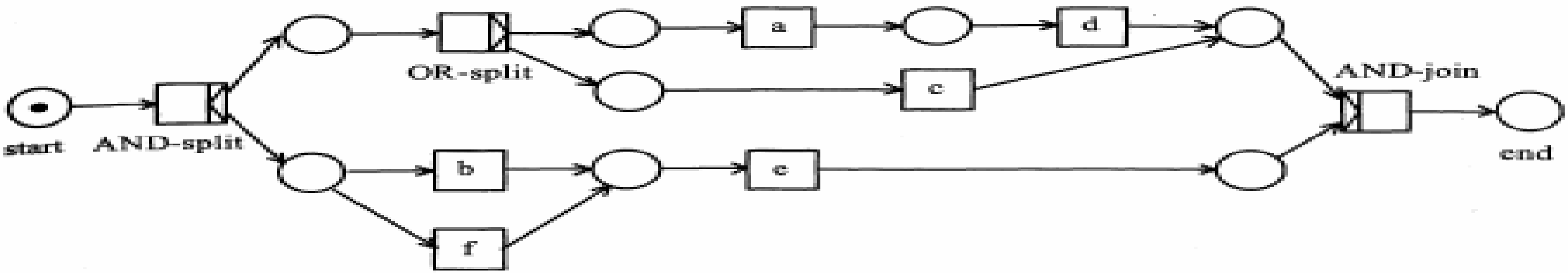


Figure 4.19
Apply the sequence constrcut to e (Step 7)

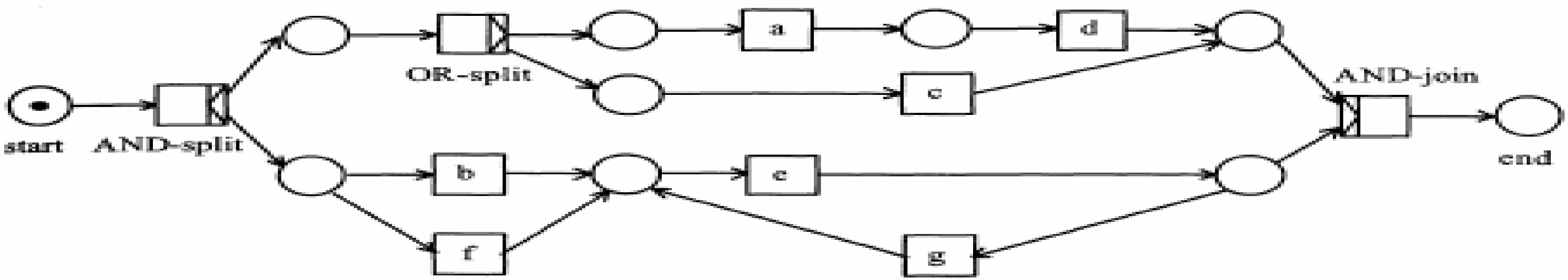
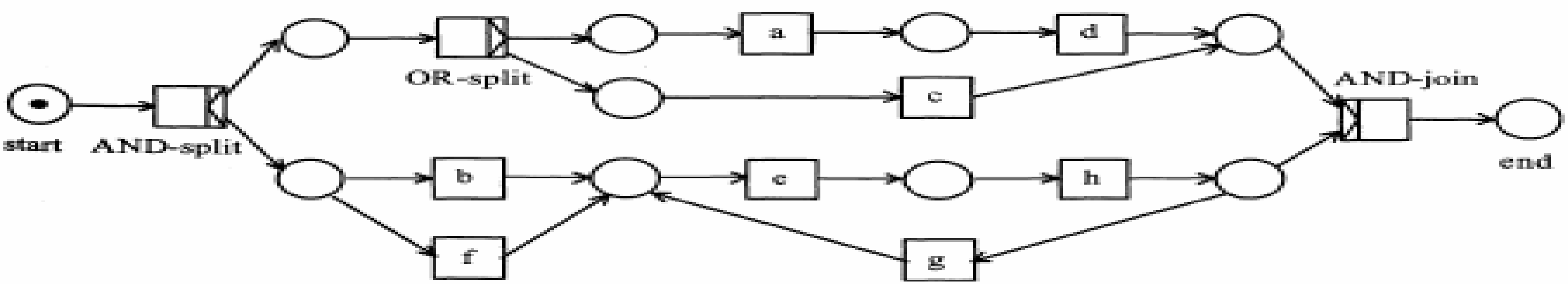


Figure 4.20
The complete process



4.3 Structural Analysis

- Not all sound and safe nets can be constructed
- it is always permissible to add a sound and safe net to our collection of building blocks

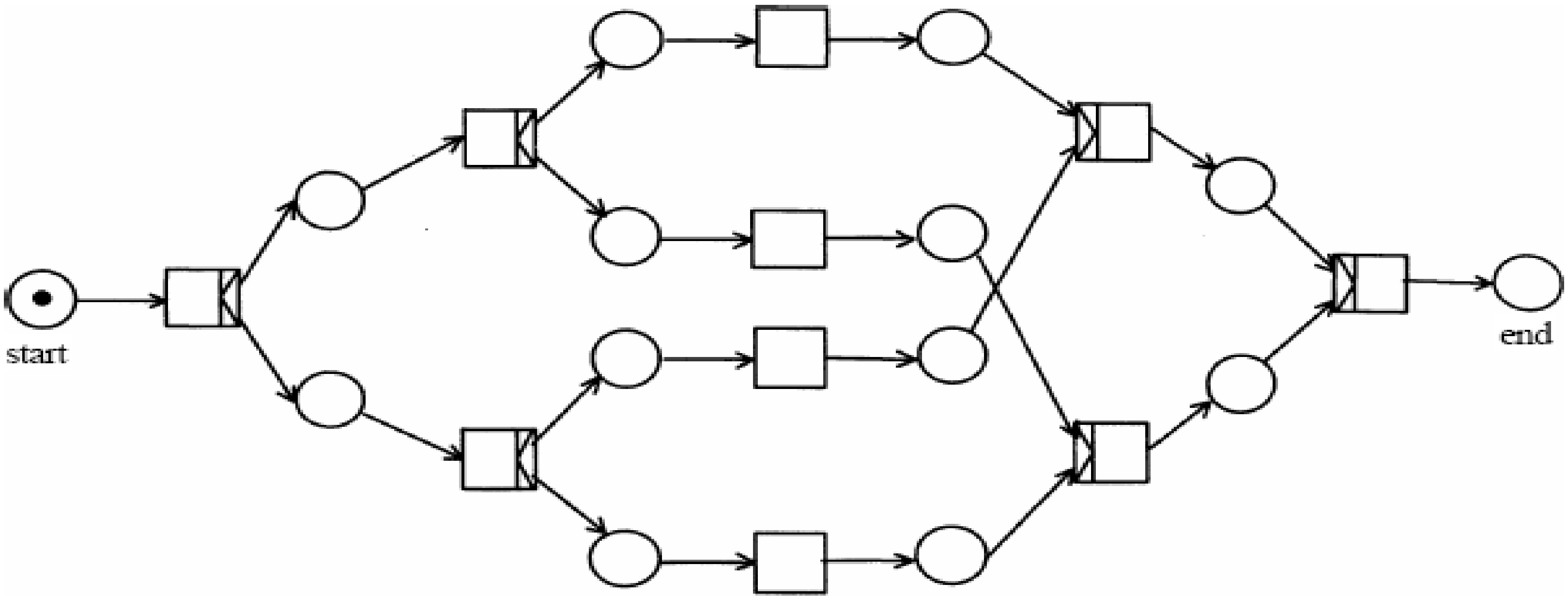


Figure 4.21
A process that cannot be constructed using the standard constructs shown in figure 4.11

4.3 Structural Analysis

- A particular extension of our replacement rules: every place (excluding source and sink places) may be replaced by a place and a task for which this place is the input as well as the output place.

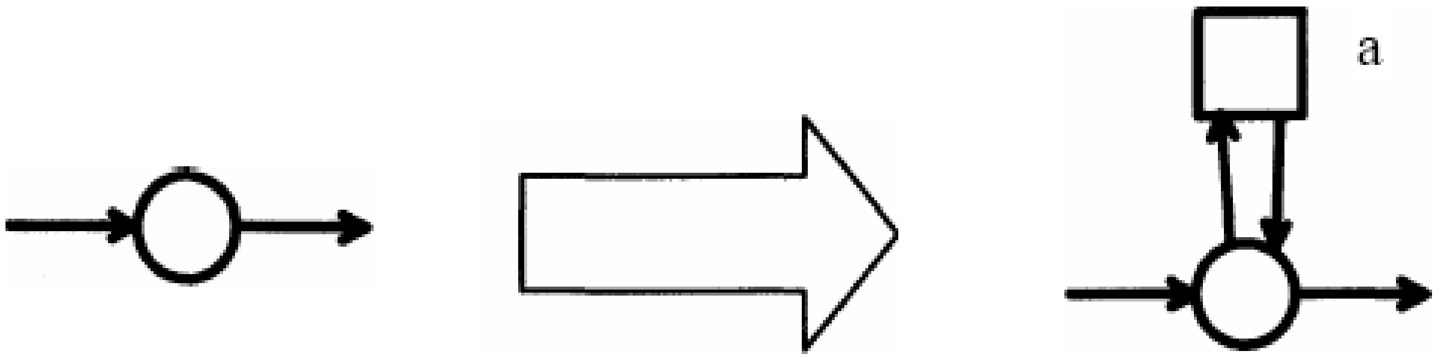


Figure 4.22
The loop construct

4.3 Structural Analysis

- we can characterize each step in a derivation by a triplet: the selected task, the used building block, and the name of the new task.

Table 4.1
Each Step in a Derivation

step	set of tasks	selected task	used block	new task
1	a	a	AND	b
2	a,b	a	explicit OR-split	c
3	a,b,c	a	sequence	d
4	a,b,c,d	b	sequence	e
5	a,b,c,d,e	b	implicit OR-split	f
6	a,b,c,d,e,f	e	iteration	g
7	a,b,c,d,e,f,g	e	sequence	h

4.3 Structural Analysis

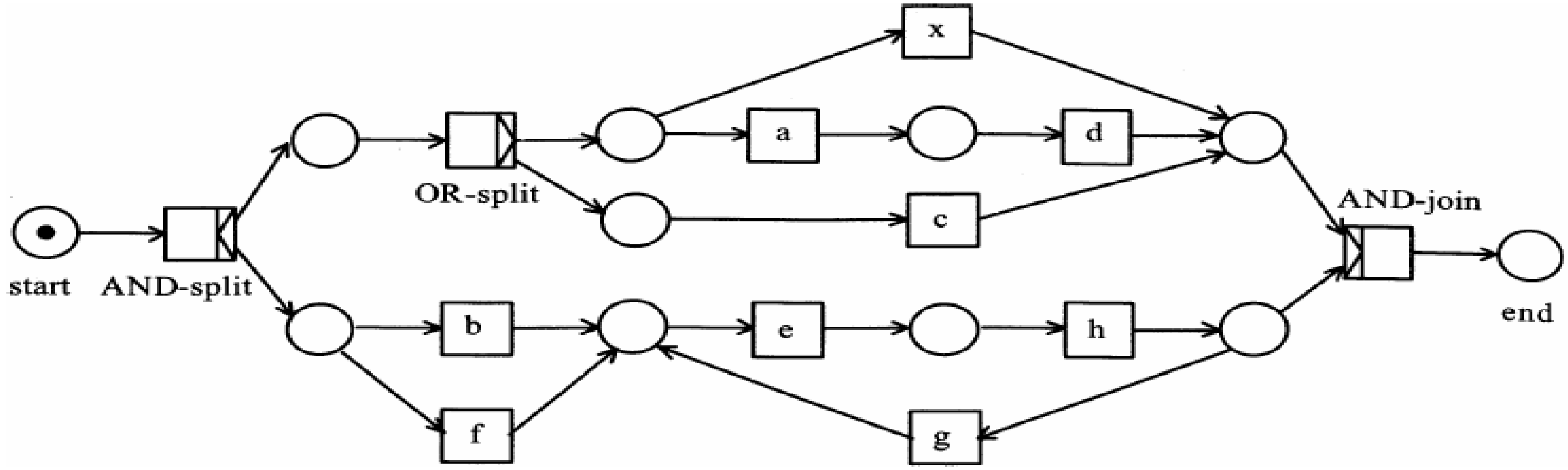


Table 4.2
Each Step in a Derivation (with 2.5)

step	set of tasks	selected task	used block	new task
1	a	a	AND	b
2	a,b	a	explicit OR-split	c
2.5	a,b,c	a	implicit OR-split	x
3	a,b,c,x	a	sequence	d
4	a,b,c,d,x	b	sequence	e
5	a,b,c,d,e,x	b	implicit OR-split	f
6	a,b,c,d,e,f,x	e	iteration	g
7	a,b,c,d,e,f,g,x	e	sequence	h

4.4 Performance Analysis

- quantitative aspects as completion times of cases, the number of cases which can be processed per time unit, the utilization of staff, and the percentage of cases that can be completed within a preset, standard time.
- three techniques
  - Markovian analysis
    - Markov chain is a reachability graph with the probability of transitions added to it.
    - what are the chances of a case taking a particular route through a process. By expanding Markov chains with cost and time aspects, a range of performance indicators can be generated.
  - Queueing theory
    - Queueing theory is intended for the analysis of systems in which the emphasis is placed upon such performance indicators as waiting times, completion times, and utilization of capacity.
    - single queue, queueing networks
    - in the presence of parallel routing, it is often impossible to apply the results obtained from queueing theory.
  - Simulation.

4.4 Performance Analysis-Simulation

- simulation boils down to the following of a path in the reachability graph. In doing so, particular choices are made based upon various probability distributions.
- It is a technique that is accessible to people without a mathematical background.
- the establishment and analysis of a model for a detailed simulation can be a time-consuming affair. Moreover, the careful processing of simulation results requires thorough statistical knowledge.
- Now, in BPR, simulation usually is the only tool available for carrying out quantitative analyses.

4.4 Performance Analysis-Simulation

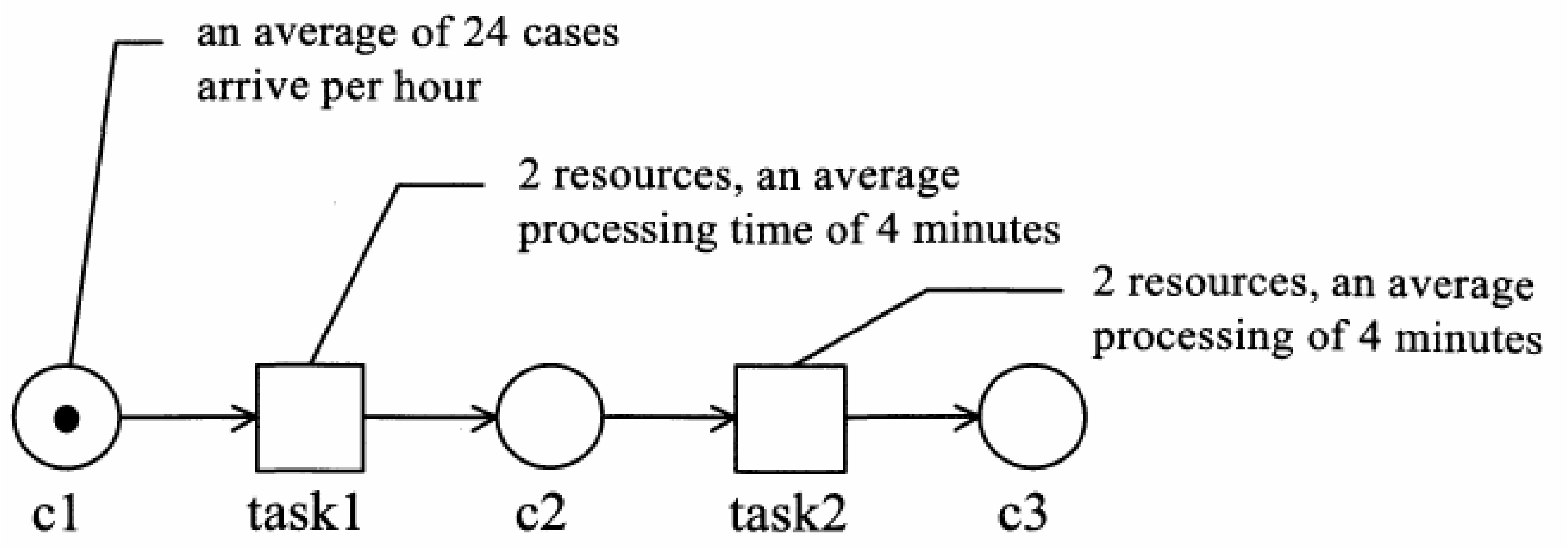


Figure 4.24
Situation 1

- average resource utilization: 80%
- average completion time for a case:
  - assume that the interarrival times are distributed in a negative exponential way.
  - it can be determined using either simulation or queueing theory that the average completion time is approximately 22.2 minutes.

4.4 Performance Analysis-Simulation

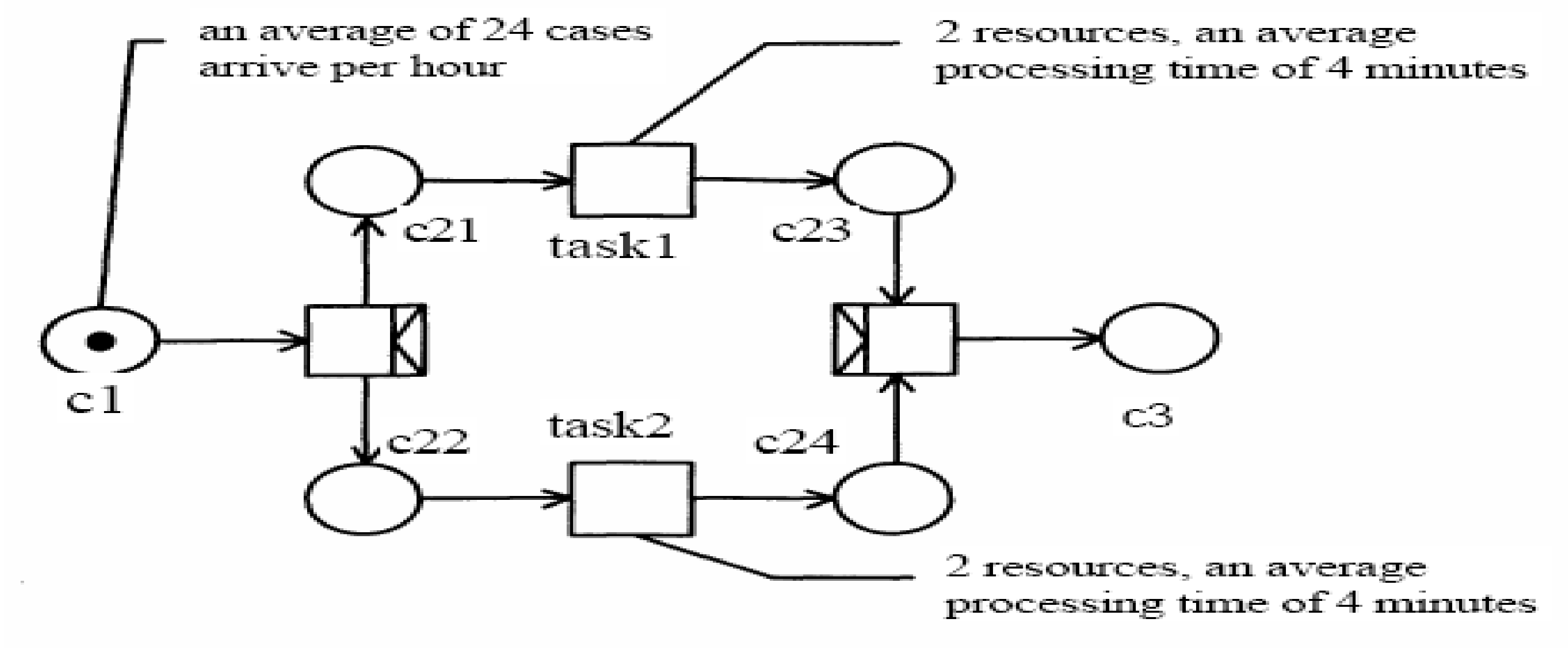


Figure 4.25
Situation 2

- average resource utilization: 80%
- average completion time for a case: 15min

4.4 Performance Analysis-Simulation

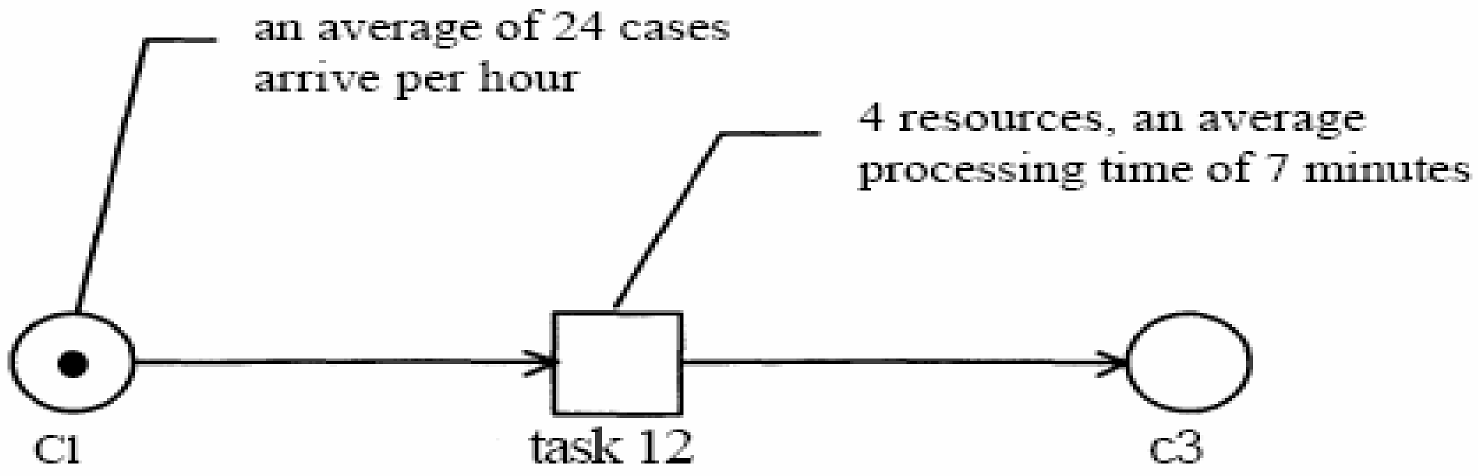


Figure 4.26
Situation 3

- average processing time for taskl2 is 7 minutes. The task12 can be performed by each of the 4 resources.
- average resource utilization: 70%
- average completion time for a case: 9.5min

4.4 Performance Analysis-Simulation

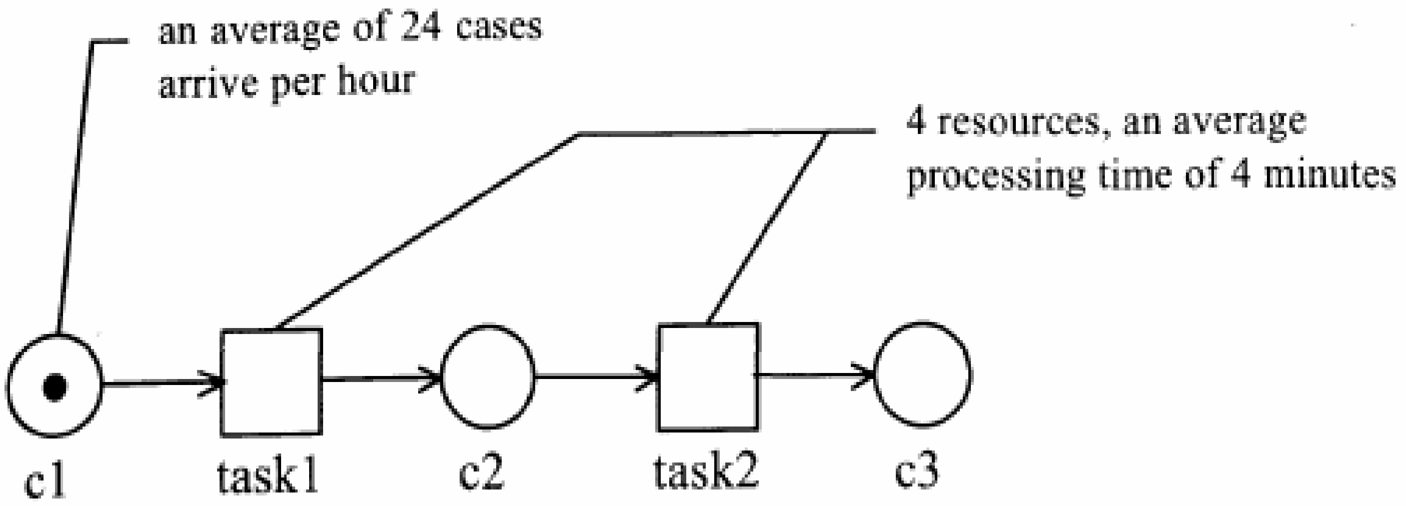


Figure 4.27
Situation 4

- average completion time for a case: 14min

4.4 Performance Analysis-Simulation

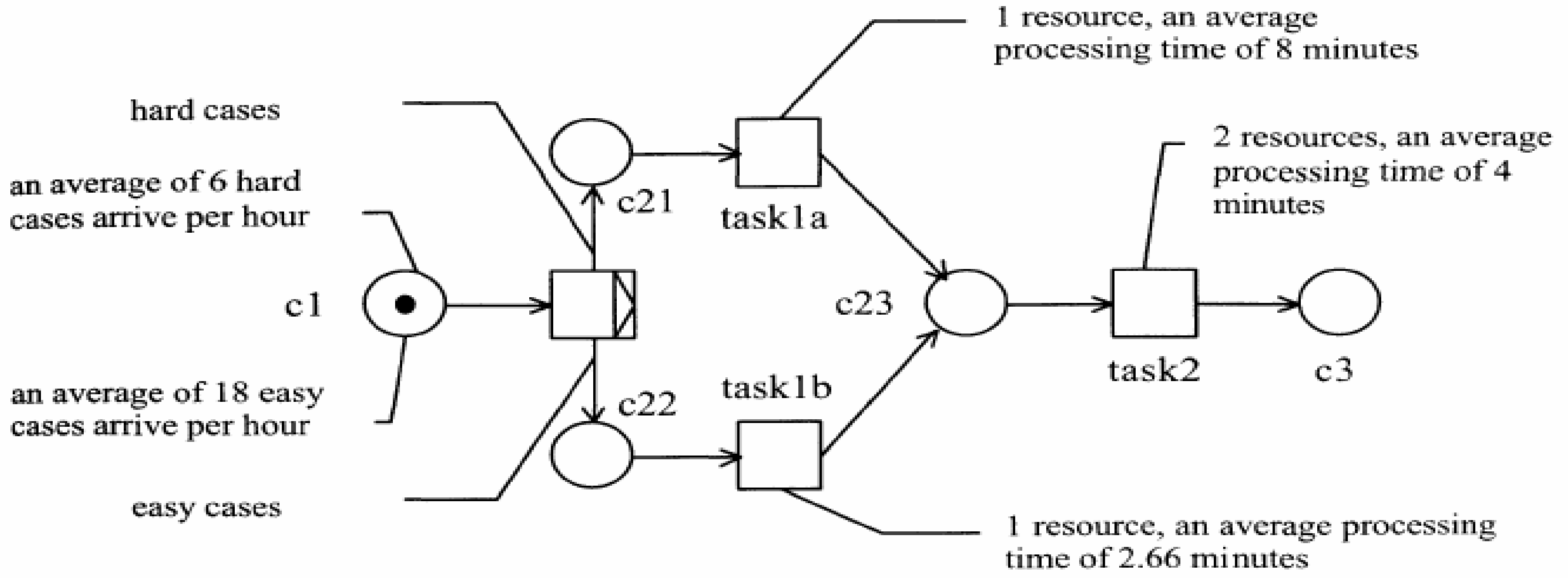


Figure 4.28
Situation 5

- Intent: the total average completion time can be reduced by separating the two flows.
- Reality: average completion time for a case: 31.1min

4.4 Performance Analysis-Simulation

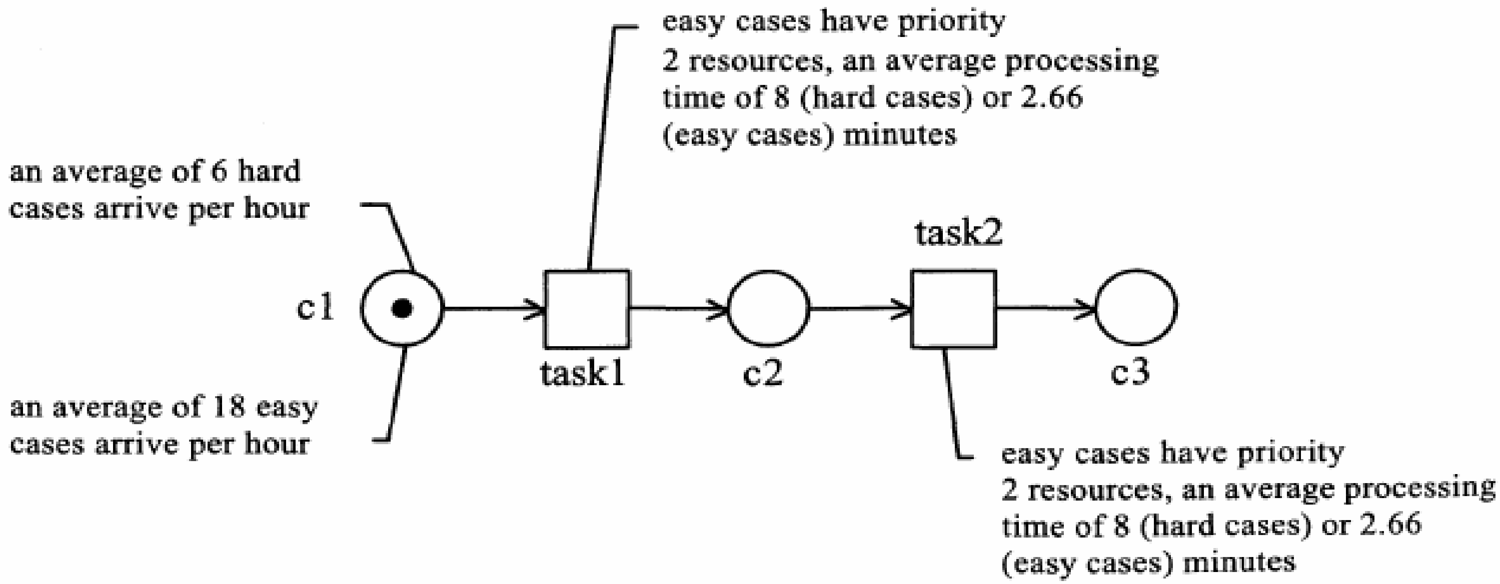


Figure 4.29
Situation 6

- Triage:each task the easy cases are given priority over the hard ones
- average completion time for a case: 14min

4.4 Performance Analysis-Simulation

Situation	Description	Average completion time	Average processing time	Average waiting time
Situation 1	Sequential	22.2	8.0	14.2
Situation 2	Parallel	15	4	11
Situation 3	Composition	9.5	7.0	2.5
Situation 4	Flexibilization	14.0	8.0	6.0
Situation 5	Triage	31.1	8.0	23.1
Situation 6	Prioritization	14.0	8.0	6.0

Figure 4.30
A summary of the performances in the six situations described

- three guidelines:
  - When possible, perform tasks in parallel.
  - Strive for high resource flexibility.
  - When possible, handle cases in order of processing time.

4.5 Capacity Planning

- the *capacity plan* is always based upon a particular *capacity requirement.*
- The plan shows what resources, and of which type, are needed for each period.
- Capacity planning may be both short term and long term.

4.5 Capacity Planning

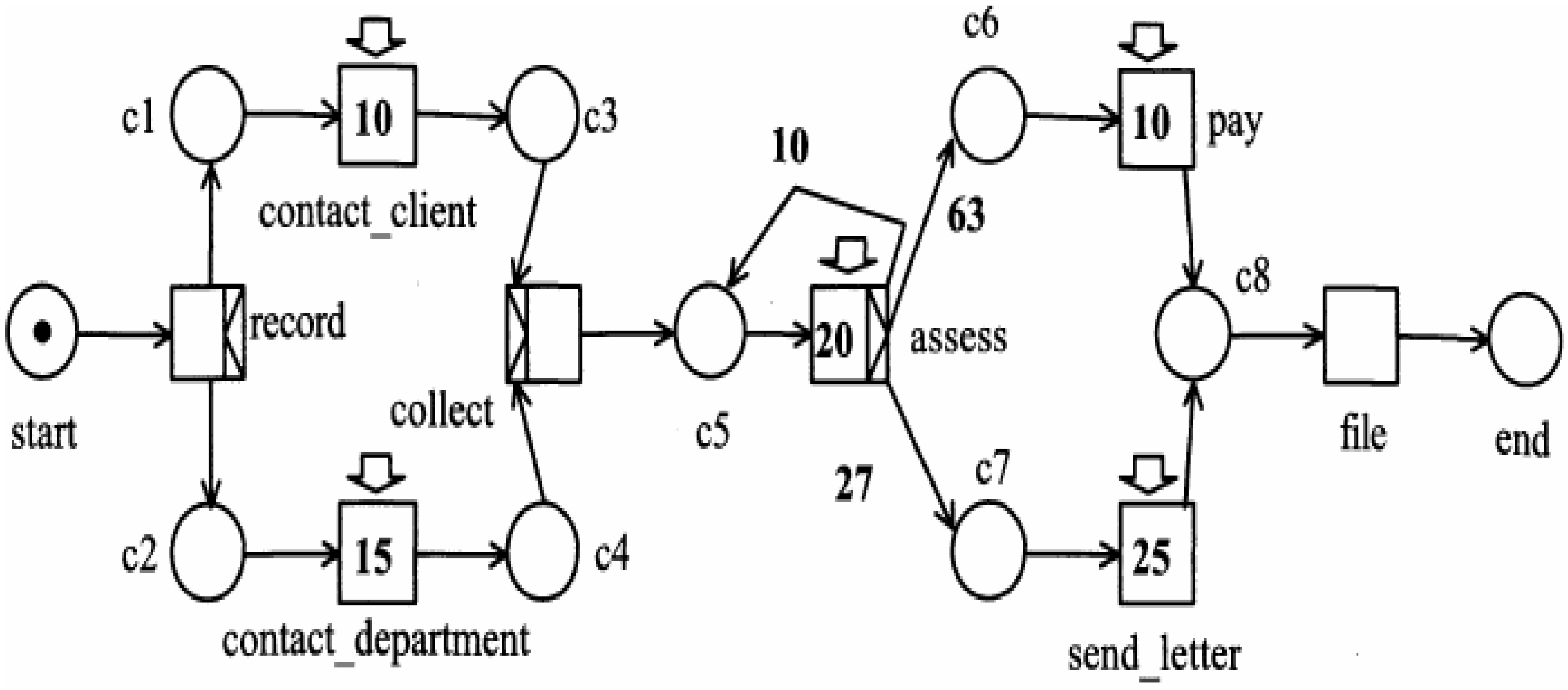


Figure 4.31

The process "handle complaint," showing the average processing time per task

4.5 Capacity Planning

Task	Average number per day	Average processing time	Average number of minutes
record	50	0	0
contact_client	50	10	500
contact_dept.	50	15	750
collect	50	0	0
assess	56	20	1111
pay	35	10	350
send_letter	15	25	375
file	50	0	0

Figure 4.32
The capacity required per task

Resource class	Average number of minutes	Number of resources at 80% of capacity	Number of resources at 60% of capacity
Employee	1975	5.14	6.86
Assessor	1111	2.90	3.86
Complaints	2736	7.13	9.50
Finances	350	0.91	1.22

Figure 4.33
The capacity requirement per resource class

4.5 Capacity Planning

- **Method to calculate capacity requirement**
  - To determine the capacity required it is important to know the average number of times each task is executed.
  - 2 Methods:
    - construct a Markov chain that is isomorphic with the reachability graph and add the appropriate cost functions.
    - approach based on the design patterns described in figure 4.11.
  - If the average number of new cases per time unit and the average number each task is executed are known, then the average number of times a given task is executed can be calculated by taking the product of these two figures.
  - If the average processing time and corresponding resource class of each task are known, it is straightforward to derive the total number of capacity per time unit per role (assuming a utilization of 100%).

4.5 Cap

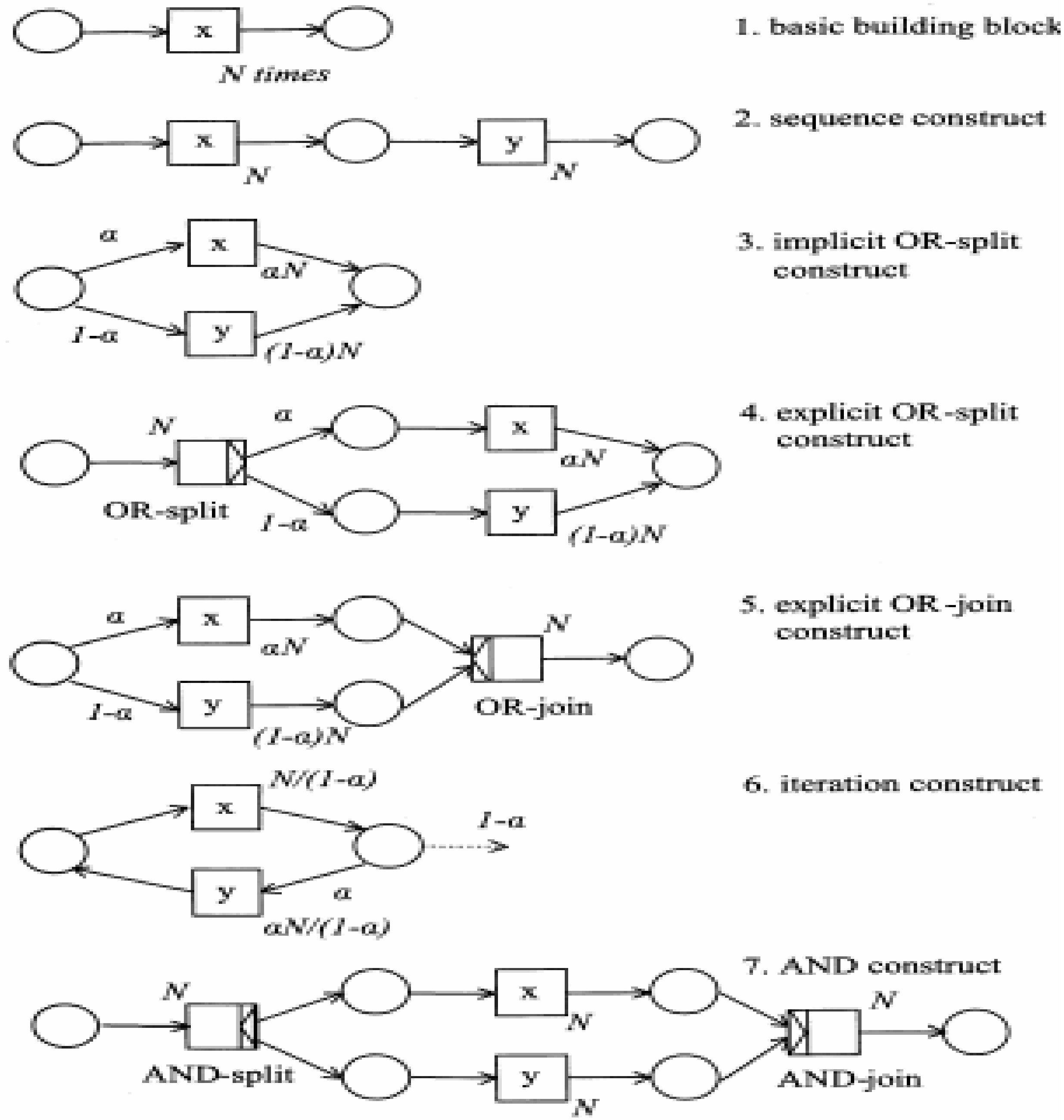


Figure 4.34
The number of times each task executed relative to the number of times task x is executed in the original situation

4.5 Capacity Planning

- Some basic queueing theory to take variability into account
  - During each time unit, λ new cases arrive that
  - need to be processed by one resource. This resource is able to complete μ cases per time unit. The utilized capacity, ρ , of this resource is therefore:

$$p = \lambda / \mu$$

  - If we assume that processing times and case interarrival times are distributed in a negative exponential way, the average number of cases in progress is L, where:

$$L = \rho /(1 - \rho)$$

4.5 Capacity Planning

- The average waiting time, W—that is, the completion time minus the processing time—is: $W = L/\mu = \rho/(\mu - \lambda)$
- The average system time, *S*—that is, the total completion time (waiting time and processing time)—is:

$$S = W + 1/\mu = 1/(\mu - \lambda)$$

- It is not at all sensible to seek a capacity utilization of more than 80 percent.
  - With a capacity utilization of 80 percent, the average completion time is thus 30(24 + 6) minutes.
  - At a capacity utilization of 95 percent and an average processing time of 6 minutes, the average completion time would rise to no less than 2 hours.

4.5 Capacity Planning

- Figure 4.35 shows the impact of utilization on the average number of cases in progress. The impact resulting from the duplication of utilization from 0.25 to 0.50 (+0.66 cases) is much smaller than the impact from the small increase from 0.98 to 0.99 (+50 cases).
- The situation just described corresponds with the *M/M/1 queue.*

Utilization (p)	Average number in progress (L)	Utilization (p)	Average number in progress (L)	Utilization (p)	Average number in progress (L)
0.10	0.11	0.80	4.00	0.98	49
0.25	0.33	0.85	5.66	0.99	99
0.50	1.00	0.90	9.00	0.999	999
0.75	3.00	0.95	19.00	0.9999	9999

Figure 4.35
The average number of cases in progress given a utilization ratio

4.5 Capacity Planning

- For *M/G/1* queue, average processing time is $1/\mu$ and that the standard deviation is *a.* then coefficient of variation, C = μa
  - capacity utilization is also equal to $\rho = \lambda/\mu$.
  - average number of cases in progress

    $L = \rho+(\rho 2/(2(1-\rho)))(1 + C2)$ (Pollaczek-Khinchin formula)
  - The average waiting time $W = (\rho/(2\mu(1-\rho)))(1 + C2)$
  - These formulae show that large variations in processing times can result in long completion times. Conversely, regular processing times will deliver shorter completion times.
  - in case of negative exponentially distributed processing times, C equals 1 and the Pollaczek-Khinchin formula reduces to the formula given earlier.
- For M/M/n, M/G/n, G/G/n…
  - L = λS (Little formula)

HomeWork

- p138 Exercise 4.5 Performance analysis I