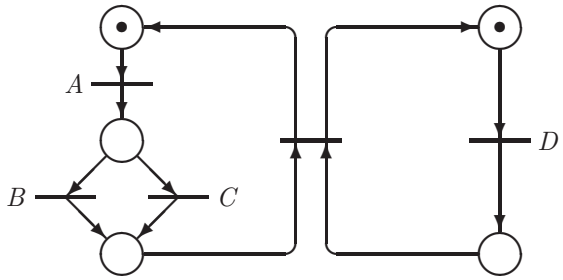
Solutions for CP Exercise Class 1

1. Solution for Petri.2

(a) The simplest Petri Net becomes:



It is seen that it is necessary to introduce an anonymous transition that ensures that the two sequential processes are synchronized in each round. (This synchronization may also be expressed in other, less obvious, ways.)

(b) From the above net it is seen that $(A, D), (B, D), (C, D)$ can be executed concurrently. (Since there exists behaviours of the net in which the corresponding transitions may fire simultanuously.)

(c) For the first round, we get the following six possible interleavings:

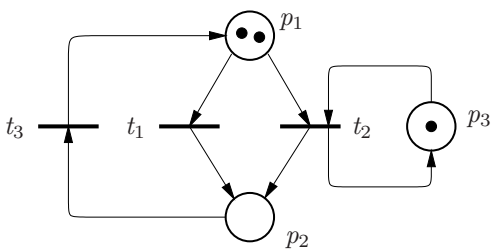
A, B, D	A, C, D
A, D, B	A, D, C
D, A, B	D, A, C

2. A place with a single token is added, and a loop from/to this place is added to each of the transitions A, B, C, and D.

3. Advantages of formal models are clearly that they are precise and unambigious. Furthermore, graphical models like Petri Nets may even be very intuitive. A disadvantage is that formal models requires the reader to know the modelling language well. Futhermore, the relationship with the real-world phenomenon being modelled may be less obvious.

4. A fork over a place-node reflects a *choice*. A fork over a transition-node reflects *process creation*.

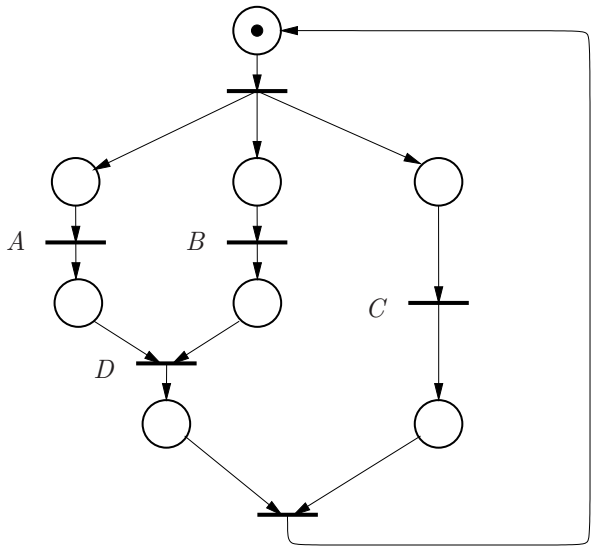
5.



Possible firings: $M_0 \xrightarrow{\{t_1\}} (1,1,1)$, $M_0 \xrightarrow{\{t_2\}} (1,1,1)$, $M_0 \xrightarrow{\{t_1,t_2\}} (0,2,1)$, $M_0 \xrightarrow{\{t_1,t_1\}} (0,2,1)$

Note that t_1 may fire together with itself, but t_2 may not!

6.



7. From the Petri-net, we see the following pattern:

repeat
  co
    co $A \parallel B$ **oc**; D
  $\parallel$
    C
  oc
forever

8. We define a slave thread for each operation and let the main thread act as a master, controlling these according to synchronization of the Petri Net:

```
class Synchronize {

  class TA extends Thread { public void run(){ A; }}
  class TB extends Thread { public void run(){ B; }}
  class TC extends Thread { public void run(){ C; }}
  class TD extends Thread { public void run(){ D; }}

  public void main(String[] argv){
    while (true) {
      Thread ta = new TA();
      Thread tb = new TB();
      Thread tc = new TC();
      Thread td = new TD();

      ta.start();  tb.start();  tc.start();
      ta.join();  tb.join();
      td.start();
      tc.join();  td.join();
    }
  }
}
```

Rather than starting/joining with `td`, the main thread could itself execute the operation D.

9. Solution for Exercise Trans.1

(1) $a_1\ a_2\ a_3\ b_1\ b_2$
(2) $a_1\ a_2\ b_1\ a_3\ b_2$
(3) $a_1\ a_2\ b_1\ b_2\ a_3$
(4) $a_1\ b_1\ a_2\ a_3\ b_2$
(5) $a_1\ b_1\ a_2\ b_2\ a_3$
(6) $a_1\ b_1\ b_2\ a_2\ a_3$
(7) $b_1\ a_1\ a_2\ a_3\ b_2$
(8) $b_1\ a_1\ a_2\ b_2\ a_3$
(9) $b_1\ a_1\ b_2\ a_2\ a_3$
(10) $b_1\ b_2\ a_1\ a_2\ a_3$

10. Solution for Exercise Trans.2

The number of interleavings is given by:

$$\left(\begin{array}{c} n_1 + n_2 \\ n_1 \end{array} \right) = \left(\begin{array}{c} n_1 + n_2 \\ n_2 \end{array} \right) = \frac{(n_1 + n_2)!}{n_1! * n_2!}$$

An argument:

> Each interleaving must contain n_1 actions from P_1 and n_2 actions from P_2, ie. in total $n_1 + n_2$ actions. We may say that each interleaving has $n_1 + n_2$ places and an interleaving is the uniquely given by selecting n_1 of these for the actions of P_1 (the actions are supposed to come in the given order). As known from combinatorics, the number of ways n_1 elements can be selected out of $n_1 + n_2$ elements is given by the above expression.

An other argument:

> For any interleaving, the actions from P_1 can be permuted in $n_1!$ ways and the actions from P_2 in $n_2!$ ways. From any interleaving, we may thus generate $n_1! * n_2!$ permuations of the total of $(n_1 + n_2)!$ permutations of all actions of P_1 and P_2. From this, the expression follows.

11. Solution for Andrews Ex. 2.11

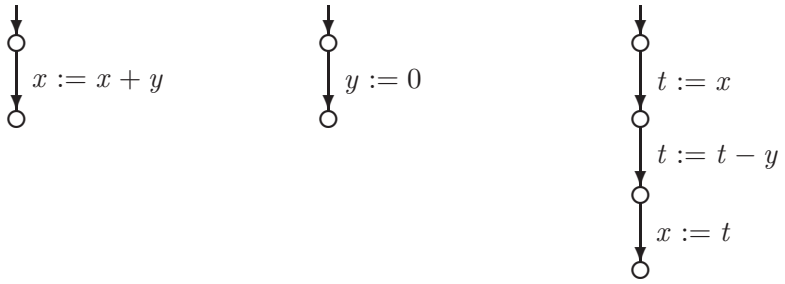
(a) The expression evaluation must be divided into three atomic reading steps. Now, each variable may or may not have been changed when read. Since all changes incidentally increment each variable by 3, zero to three increments may be seen. Thus, the possible final values are $\{3, 6, 9, 12\}$.

(b) Since the variables are updated by independent processes, each variable still may or may not have been changed when read irrespective of the ordering of the readings. Thus, the possible results are again: $\{3, 6, 9, 12\}$

12. Solution for Andrews Ex. 2.14

(a) No, the statement of the last process, $x := x - y$, has three critical references, since both x and y are read and written by other processes. The statement $< x := x + y >$ is by definition atomic and thus not subject to the rule.

(b) Rewriting the last process to atomic statements $< t := x >$; $< t := t - y >$; $< x := t >$, the three processes are represented by the transition diagrams:



These give rise to 20 interleavings. Of course, all of these lead to $y = 0$. Analyzing the interleavings, we find that x may get the values $\{0, 1, 2\}$.