UNIVERSITY
OF SKÖVDE

School of

WRITTEN EXAMINATION

Course: Concurrent Programming

Examination

Course code: IT404G

Credits for written examination: 4.5

Date: 2025-01-15

Examination time: 5 hours

Examination responsible: Richard Senington

Teachers concerned: Birgitta Lindström

Aid at the exam/appendices

Other

Instructions

- ☐ Take a new sheet of paper for each teacher.
- ☐ Take a new sheet of paper when starting a new question.
- ☒ Write only on one side of the paper.
- ☒ Write your name and personal ID No. on all pages you hand in.
- ☒ Use page numbering.
- ☒ Don´t use a red pen.
- ☒ Mark answered questions with a cross on the cover sheet.

Grade points

Examination results should be made public within 18 working days

Good luck!

Total number of pages

Parallella processer G1F, IT404G. 2025-01-15

Grading

There are five main questions in the written exam corresponding to the course objectives. Each main question consists of a set of three sub-questions, which are graded pass or fail. To pass the exam, you need to ***pass at least one sub-question for each of the main questions****. The more sub-questions you pass, the higher your grade will be. The detailed grading scheme is published in Canvas. For your convenience, each section lists the relevant examination criterion.*

Main question 1

Examination criterion: *Redogöra för olika frågor som måste hanteras i program med samtidigt exekverande processer, inklusive tävlan om resurser och ömsesidig uteslutning*

Sub-question 1a

Please describe a memory based spin lock and compare it to a semaphore with respect to all three safety properties, mutual exclusion, absence of deadlock and absence of unnecessary delay. Please also discuss efficiency in your answer.

Sub-question 1b

Please evaluate the two monitors and the two processes below, with respect to mutual exclusion, absence of deadlock, liveness and absence of unnecessary delay.

Monitor a{
op doX(){
p=b.doY()
return p+1
}
op doZ(){
return random()
}
}
	Monitor b{
	op doY(){
	// assume generate is a safe operation
	// outside the monitors
	return generate()
	}
	op doW(){
	return a.doZ()+1
	}
	}

Process 1
While true{
a.doX()
}
	Process 2
	While true{
	b.doW()
	}

Sub-question 1c

Describe the similarities and differences between a counting semaphore and a binary semaphore. Include any possible impact on the safety and liveness properties (e.g. mutual exclusion, absence of deadlock, absence of unnecessary delay and liveness) in your answer.

Parallella processer G1F, IT404G. 2025-01-15

Main question 2

Examination criterion: *Identifiera, beskriva och diskutera klassiska synkroniseringsproblem mellan parallella processer såsom synkronisering av läsare och skrivare eller av producenter och konsumenter*

Sub-question 2a

Please identify and describe the classic problem seen in the following description. Please identify different processes, what these processes are doing, and what the synchronization and communication mechanisms might be.

This example is set in a restaurant kitchen. There is a chef who prepares the meals. Once a meal is prepared they place it on a table and ring a bell. There are two waiters. If a waiter has nothing to do they are waiting to hear the bell (or see a meal waiting) and then they take the meal and deliver it to the customers table.

Sub-question 2b

Please identify and describe the classic problem seen in the following code. Please identify what each process is doing. Please identify any problems that you can see.

Binary Semaphore S1, S2, S3, S4 = 0;				

Process 1
While true{
P(S1)
P(S2)
V(S1)
V(S2)
}
	Process 2
	While true{
	P(S2)
	P(S4)
	P(S3)
	V(S2)
	V(S3)
	V(S4)
	}
		Process 3
		While true{
		P(S4)
		V(S4)
		}
			Process 4
			While true{
			P(S3)
			P(S1)
			V(S3)
			V(S1)
			}
				Process 5
				V(S1)
				V(S2)
				V(S3)
				V(S4)

Sub-question 2c

Please identify and describe all the classic problems seen in the following code. Please identify what each process is doing, and how it interacts with those it is connected to.

Chan C1,C2,C3,C4,C5,C6
Available = [3,4,5]

Process 1

```
For i in 20{
  x = generate()
  C1.send(x)
  C6.receive()
}
```

Process 2

```
Y = Nothing
While true{
  If y is Nothing{
    C1.recieve(y)
    C6.send()
  }
  if length(Available)>0{
    k=Available.pop()
    if k==3{
      C2.send(y)
    } else if k==4{
      C3.send(y)
    } else if k==5{
      C4.send(y)
    }
    Y = Nothing
  }
  If c5.has_messages(){
    C5.receive(k)
    Available.append(k)
  }
}
```

Process 3

```
While true{
  C2.receive(x)
  // act on x
  C5.send(3)
}
```

Process 4

```
While true{
  C3.receive(x)
  // act on x
  C5.send(4)
}
```

Process 5

```
While true{
  C4.receive(x)
  // act on x
  C5.send(5)
}
```

Main question 3

Examination criterion: *Beskriva för- och nackdelar med olika tekniker för att lösa synkroniseringsproblem, inklusive semaforer, monitorer och tekniker för meddelandeöverföring*

Sub-question 3a

Describe the advantages and disadvantages of using Monitors or Semaphores for the reader writer problem. You will have to explain how the different processes would communicate and possibly give pseudo code of different processes to help you explain the context.

Sub-question 3b

Describe the differences between the filter architecture for message passing vs the client-server architecture for message passing. Give examples of where each would be an effective choice.

Sub-question 3c

Describe the advantages and disadvantages of using Monitors vs A-synchronous message passing for the dining philosopher problem. You will have to explain how the different processes would communicate and possibly give pseudo code of different processes to help you explain the context.

Main question 4

Examination criterion: *Använda grundläggande tekniker såsom semaforer och meddelandeöverföring för att lösa synkronisering och kommunikation i program med parallella processer*

Sub-question 4a

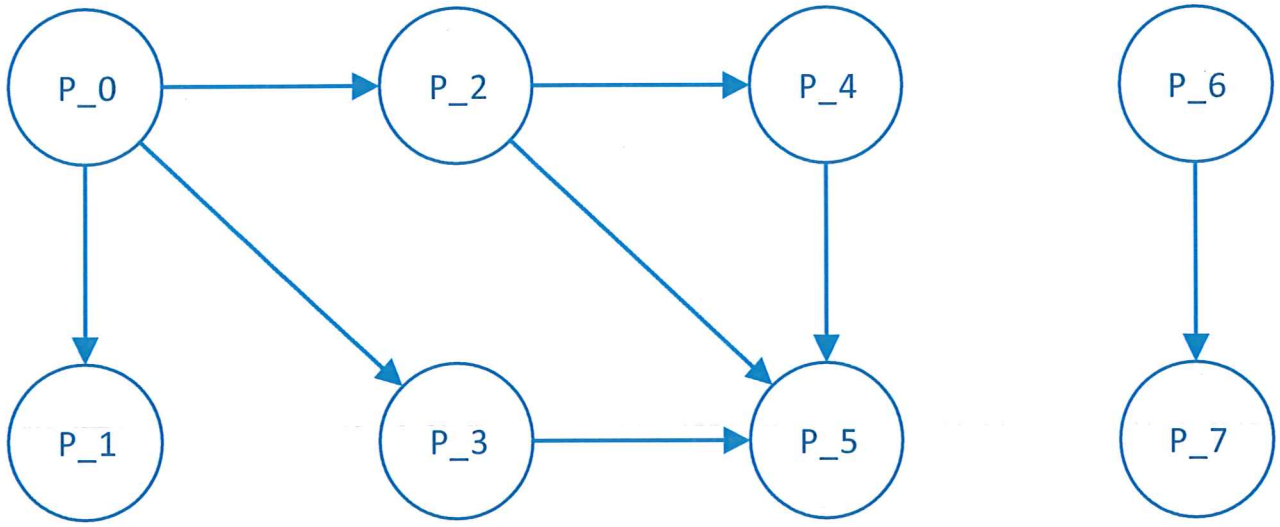
A university has a very busy, and dynamic booking system for its rooms. To support this a computer system is being designed. The idea is that anyone will be able to read the current status of a room, but only 1 client can change the status of a room at any given time. However, you should not change the status of a room while another client is reading the status.

Please design a monitor in pseudo-code to support this functionality. An additional requirement, if someone tries to change the status of a room they should be put in a queue until it becomes available.

Sub-question 4b

What follows is a diagram of a set of processing nodes. Each process should execute once. Synchronization will be achieved using semaphores. Please state the number of semaphores required, initial setup of the semaphores and any limits you wish to place on them. Please provide simple pseudocode for each process. Please note that the arrows mean that the later process should not begin until ALL the earlier connected processes have completed.

P_0 can be thought of as the starting process.



Parallella processer G1F, IT404G. 2025-01-15

Sub-question 4c

Channels usually only have a send and receive operations. For this problem we also allow a *has_messages* operation that will tell you if there are waiting messages or not.

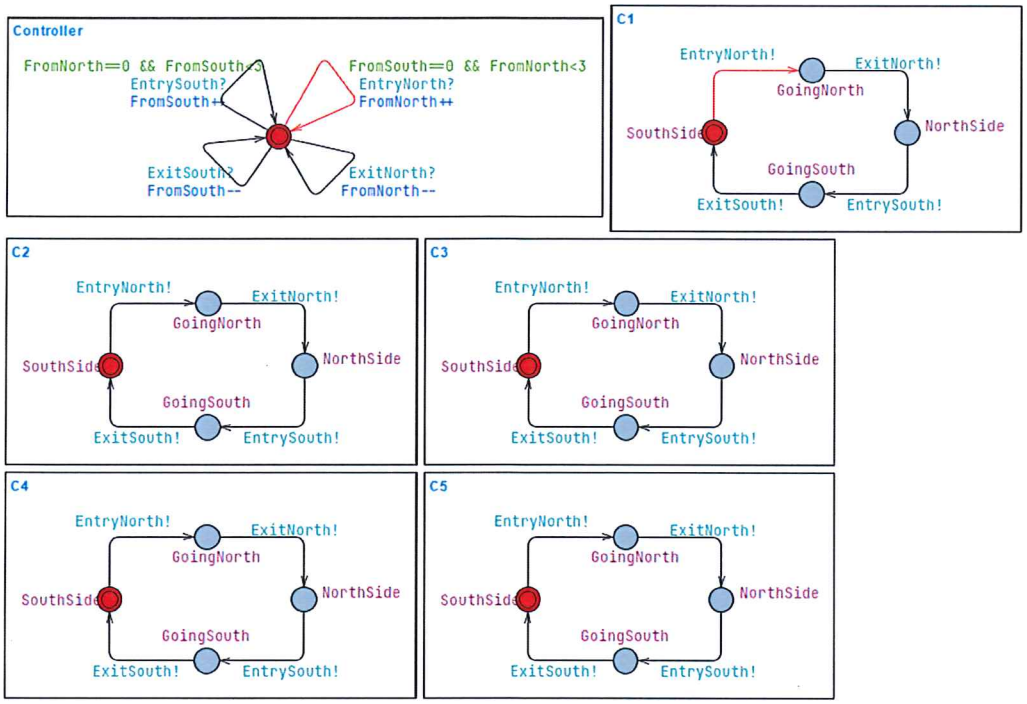
The party problem was introduced in the lectures. In the party problem there is a bowl of punch with a maximum volume (3 drinks in this example). There are 2 kinds of process, the host and the guest. The host can refill the bowl, and there is only 1. The guest mingles and drinks. If the bowl runs out the guest has to ask the host to refill the bowl.

Please write a pseudo-code implementation of this problem. You should provide the code for the host and 2 guests. The bowl can hold 3 drinks (more than that would be an error). Use a-synchronous channels for all communication.

Parallella processer G1F, IT404G. 2025-01-15

Main question 5

Examination criterion: Modellera och verifiera egenskaper hos program med parallella processer, inklusive progression, frånvaro av låsning och ömsesidig uteslutning



Consider the above model for the old bridge problem. There are five cars that cross the bridge when moving between the north and south sides. **Note** that the system uses the inbuilt synchronous messing passing mechanism in uppaal. This means that a sender is blocked until someone is ready to receive and vice versa. Send and receive then happens simultaneously.

Sub-question 5a

Explain the role of the controller. I.e., what it does and why. Give an elaborate example for one of the cars.

Sub-question 5b

Formulate an uppaal query to verify that C1 and C2 cannot move in opposite direction.

Sub-question 5c

Formulate an uppaal query to verify that all cars cannot cross the bridge at the same time.