Concurrency:
State Models & Design Patterns

Practical Session

Week 11

Lab 02

Discussion

Lab 02 - Task 1 - Nested Monitor

What you should have done:

- Removal of synchronization in put() / get()

Issues you encountered:

- Introduction of additional (unnecessary) synchronisation

Lab 02 - Task 2 - Readers and Writers

What you should have done:

- Implementation of WritersPriorityPolicy (additional check if writers are waiting)
- Implementation of FairReadWritePolicy (additional flag for turn)

Issues you encountered:

- Writer priority didn't work (too much permits released, wrong conditions used, ...)
- Fairness was not implemented properly (flawed starting conditions, ...)
- Generic mistakes: wrong value assignments (booleans), variable shadowing, ...

Lab 02 - Task 3 - Single Lane Bridge

What you should have done:

- Implementation of SafeBridge (additional waits / notifies)
- Implementation of FairBridge (additional flag for turn)

Issues you encountered:

- Missed variable changes in while condition
- Deadlocks (cars wait endlessly due to missed notifications)
- Fairness was not implemented properly due to flawed condition checks

Lab 02 - Task 4 - Golf Ball Allocation Monitor

What you should have done:

- Implementation of FairAllocator (introduction of player numbering -> FIFO)

Issues you encountered:

- No time left
- Issues with fairness due to missing implementation
- At most one player added at a time

Exercises 11

Preview

Exercise 11 - Task 1

Answer the following *questions* about Petri nets:

a) List and briefly explain all the elements a petri net consists of.
b) How can nets model concurrency and synchronization?
c) What is the reachability set of a net? How can you compute this set?
d) What kinds of Petri nets can be modeled by finite state processes?
e) What are some simple conditions for guaranteeing that a net is bounded?
f) What could you add to Petri nets to make them Turing-complete?

Exercise 11 - Task 2

Perform some *analysis* on the provided Petri nets:

a) Provide the definition of the Petri net in figure 1.

b) Provide the definition of the Petri net in figure 2.

c) Is the Petri net in Figure 2 bounded? Safe? Conservative? Are all the transitions live?

Exercise 11 - Task 3

Two machines need to interact with a database. The machines can read, write or stay idle. ***Model*** the situation using Petri nets ensuring that the machines cannot write at the same time.

Use the Petri net editor in the web site of the course. Hand-drawn Petri net diagrams are acceptable, ***but make them readable please!***

Exercise 11 - Task 4

Answer the following *questions* about lock objects and threads:

a) How do the classes ReentrantLock and Semaphore support fairness?

   Hint: You may have to look at the Java documentation.

b) What are daemon threads in Java? What is their purpose? How can you create them?