Class 10

Learning Objectives

- Understand the basic concepts behind workflow nets,
- Be able to model basic processes using workflow nets

Textbook, Section 2.3

This section introduces more formally the notion of petri nets that was informally developed in the previous reading. Do not be distracted by all the formalism, it merely makes more precise what the section informally says. It is not necessary that you remember any of the formalisms. The main idea is that a Petri net consists of places, transitions, and arcs. (Note that the textbook merely says that "places represent intermediate states that may exist during the operation of a process" (page 51). In other words, they do not necessarily represent physical objects.) Definition 1 adds that no place can also be transition and that arcs go from place to transition or transition to place only. The idea of tokens on places is made more precise by the concept of a *marking ("state")*. A *marking* is simply a list of how many tokens are on each place. The enabling and firing rules are described on page 52 (and in Definition 5.2) and shown graphically in Figure 2.14. An important idea is that of *reachability* of a marking. A marking M2 can be reached from a marking M1 if there is any series of transitions that fire, beginning with marking M1 and ultimately leading to M2 (not necessarily in a single step/with a single transition).

Finally, definitions 3 to 6 are important properties of business processes. For example *liveness* means that there are no deadlocks in a process. Imagine a process where activity "Credit Check" has to wait for activity "Address Check" to finish (because it needs a customer's address), but "Address Check" is at the same time waiting for "Credit Check" (e.g. because it provides ownership information about the address). Hence, the process is deadlocked and cannot continue. Liveness is simply the absence of such deadlocks. Obviously, it is nice if we could guarantee that our business processes are free of deadlocks.

Similarly, boundedness is nice to have. If the places represent in-boxes or queues of business items (forms, applications, etc.) to be processed, it would be nice to have an upper limit on the number of items that can be in these in-boxes. This ensures for example that the process is capable of dealing with a certain volume of business items and the wait list does not grow infinitely.

Notice that both liveness and boundedness cannot be decided for a Petri net in general, but only for a Petri Net *and a given marking*.

Subsection 2.3.2 introduces workflow nets, which are really a special kind of Petri net with the two additional properties or constraints informally given on page 54 and formalized in Definition 7: It has a single start place ("source") and a single ending place ("sink").

A desirable property of workflows is *soundness*. Soundness means that, when starting with a single token in the source place, the process will always (no matter what choices are made when transitions are fired) end up with a single token in the sink place (with no tokens left anywhere else), which indicates process completion. In other words, no matter what choices are made, the process will always finish.

Do not read Section 2.3.3

Review questions and exercises:

Using the WoPeD software, do the following:

- Construct a sinple Workflow net that is not safe (Definition 8). Identify why it is not safe. Verify this by means of the token game.
- Chapter 2, Exercise 2 (Instead of YAWL, use Petri nets and Workflow nets). Using the semantic analysis of WoPeD, make sure that you have modelled a workflow net and the workflow net is sound.

  > *You will find that this exercise is both tricky and ambiguous. Identify what is hard (or impossible) to do with Petri and Workflow nets. What would be needed to make it easier? What is ambiguous about the exercise? How many different ways can you model this? Are they all equivalent?*

- Chapter 2, Exercise 4 (Instead of YAWL, use Petri nets and Workflow nets). Using the semantic analysis of WoPeD, make sure that you have modelled a workflow net and the workflow net is sound.

  > *You will find that this exercise is both tricky and ambiguous. Identify what is hard (or impossible) to do with Petri and Workflow nets. What would be needed to make it easier? What is ambiguous about the exercise? How many different ways can you model this? Are they all equivalent?*