

Softwaretechnik/Software Engineering

<http://swt.informatik.uni-freiburg.de/teaching/SS2017/swtv1>

Exercise Sheet 4

Early submission: Wednesday, 2017-06-28, 12:00 Regular submission: Thursday, 2017-06-29, 12:00

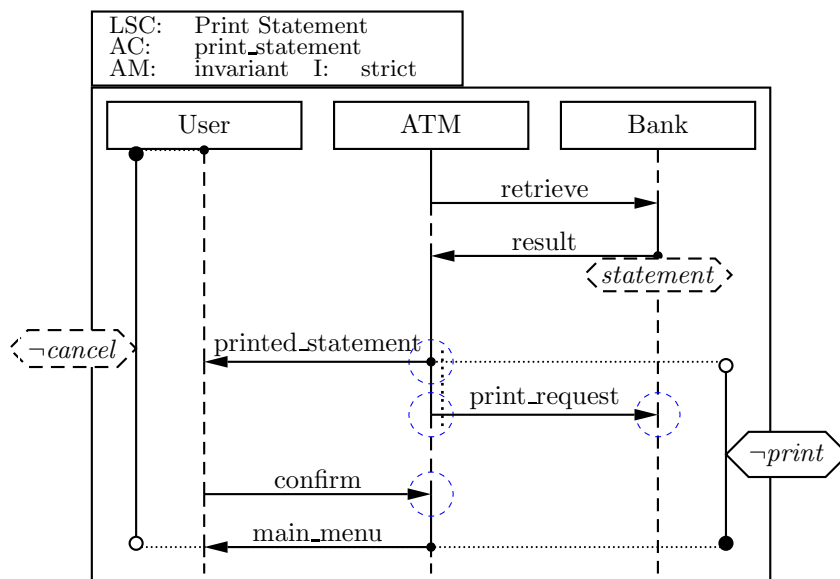


Figure 1: Live Sequence Chart for the “Print Statement” use case.

Exercise 1 – LSC Syntax and Semantics

(10/20)

Consider the Live Sequence Chart (cf. Figure 1) for a “Print Statement” use case of the ATM example from the previous exercise sheet.¹

- (i) From the **abstract syntax** of the chart, write down (using the formal notation from the lecture):
 - a) the set of locations of the chart (also draw the names next to the location on the chart) (note that all locations in a coregion obtain the same temperature from the relevant instance line segment adjacent to the whole coregion),
 - b) the set of instance lines \mathcal{I} of the chart,

¹Some extra information for the curious readers (i.e. not needed to solve the exercise): the LSC specifies the case when the customer wishes to display and print the latest available statement. The ATM initiates retrieval of the statement data from the bank once the user has selected the corresponding option. The ATM displays the statement on the screen and presents the user two options: to confirm the statement and return to the main menu or to print the statement. The statement can be printed only once, and the ATM reports the printing job to the bank. The chart specifies the sequence of messages for the case when the user chooses to print the statement. Observations corresponding to chart elements like message ‘retrieve’ and condition ‘print’ have been established with the customer and documented in a term glossary.

- c) the partial order relation (\preceq) and the simultaneity relation (\sim) for the locations marked in blue,
- d) **an example** of a message from the chart,
- e) **an example** of a local invariant from the chart, including its temperature,
- f) and **an example** of a condition from the chart, including its temperature.

(3)

- (ii) Compute the **Büchi automaton** for the chart body. The strictness conditions need to be given only for one loop, one progress, and one exit transition. (4)

*Show the steps of your construction: write down the **cut** for each state and the **fired sets** for each transition.*

- (iii) For each of the following cases, give one **example computation path** π such that:

- a) π is accepted by the chart without taking a legal exit. (1)
- b) π violates the chart. Point out why your trace violates the chart (what messages are received, which condition or invariant is not satisfied, etc.). (1)
- c) π takes a legal exit, i.e. it exits legally during the traversal of the main chart body. Describe, in your own words and using your example, the intuition behind the concept of 'legal exit'. (1)

Exercise 2 – (Creating LSCs)

(5/20 Points)

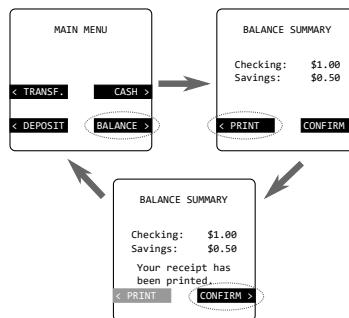


Figure 2: ATM screen example for the “Check Balances” use case.

Create **universal Live Sequence Charts (LSC)** to specify the ATM behavior for the use case from the previous exercise sheet, including the exceptional case “connection to the bank is interrupted when checking the balances”. Remember also the “Cancel” button on the ATM’s keyboard. Create your diagrams such that the cases when the user presses it is another exceptional case (whose LSC need not be written).

Available messages are shown in the following table, for simplicity assume that all messages are instantaneous.

Source	Destination	Message
User	ATM	cancel
User	ATM	confirm
ATM	User	display_balances
ATM	User	display_error
ATM	User	display_menu
ATM	User	print_receipt
User	ATM	print_request
ATM	Bank	retrieve_balances
Bank	ATM	transaction_result

Also, for the LSCs, assume that the user has already selected the corresponding option and the balances screen is being displayed. (This means that the activation condition of the chart is that the *balances* screen is being displayed and thus you do not need a prechart.)

For each LSC, give a brief description of the sequences that you want to cover with it. Also, give a brief description of how the LSCs together conform a more formal specification of the use case.

Exercise 3 – Object and Class Diagrams

(5/20)

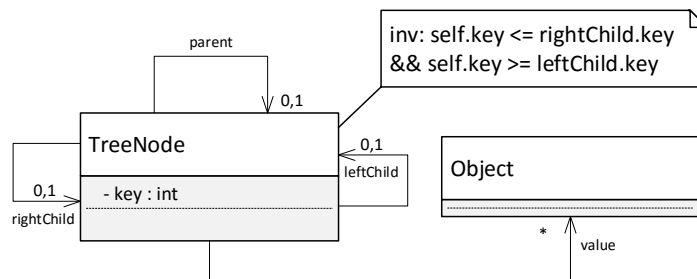


Figure 3: Class diagram for a sorted tree.

Consider the class diagram shown in Figure 3.

- (i) Provide the **abstract syntax** of the diagram, i.e. present the class diagram as a *signature*. (2)

Hint: Multiplicities other than 0..1 are treated as 0.. for the abstract syntax.*

- (ii) The class diagram is supposed to model the well-known binary tree data-structure.

Provide a system state which illustrates the use of the data-structure. To be a useful illustration, it should comprise at least

- three instances of `TreeNode`,
- at least two unconnected binary trees, one of them comprising at least two nodes.
- at least one dangling reference.

(3)

Hint: system states can be represented in different forms. You may choose the form to present your solution.