

Softwaretechnik/Software Engineering

<http://swt.informatik.uni-freiburg.de/teaching/SS2018/swtv1>

Exercise Sheet 4

Early submission: Wednesday, 2018-06-20, 12:00 Regular submission: Thursday, 2018-06-21, 12:00

**Exercise 1 – LSC Syntax and Semantics (10/20)**

Consider the live sequence chart in Figure 1.

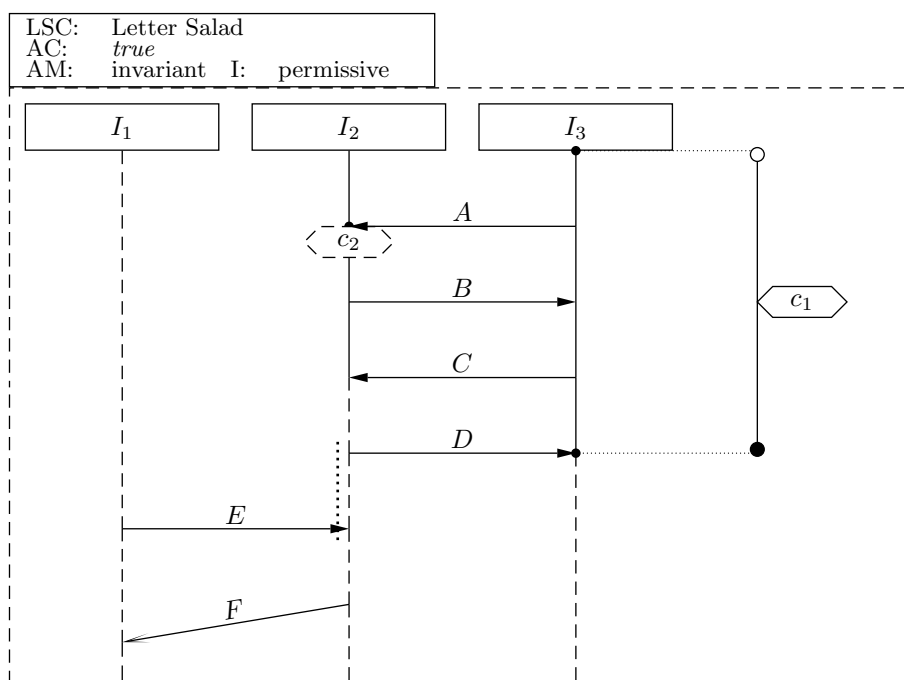


Figure 1: A live sequence chart.

- (i) From the **abstract syntax** of the chart, write down
- the set of locations  $\mathcal{L}$  (also draw the names next to the location on the chart), including their temperature,
  - the partial order relation  $\preceq$  for the locations on instance lines  $I_1$  and  $I_2$ ,
  - one** example element from the simultaneity relation  $\sim$ , and
  - one** example element from the set of messages  $\text{Msg}$ , the set of local invariants  $\text{LocInv}$ , and the set of conditions  $\text{Cond}$ , each including their temperature.

Use the formal notation from the lecture. You can show the temperature with colors. (3)

*Hint: Note that all locations in a coregion obtain the same temperature from the relevant instance line segment adjacent to the whole coregion.*

- (ii) Compute the **Büchi automaton** for the chart body. The strictness conditions  $\psi_{\text{strict}}$  need to be given only for one loop and one progress transition. (4)

Show the steps of your construction: Write down the **cut** for each state and the **fired-set** for each transition.

- (iii) Consider the following computation path  $\pi_1$ .

$$\pi_1 \equiv \sigma_0 \xrightarrow{A_2^{I_2, I_3} \wedge A_1^{I_2, I_3}} \sigma_1 \xrightarrow{B_2^{I_2, I_3} \wedge B_1^{I_2, I_3}} \sigma_2 \xrightarrow{\tau} \sigma_3 \xrightarrow{\tau} \dots$$

where  $\sigma_i \models c_1 \wedge \neg c_2$  for all  $i \geq 0$ .

Intuitively, the first event in the path is sending and receiving the message  $A$ . The second event is sending and receiving the message  $B$ . After that, no event happens ever again. During the whole computation path the condition  $c_1$  holds and the condition  $c_2$  is violated.

- a) Is the path  $\pi_1$  accepted by the chart or does it violate the chart? If it is accepted, does it take a legal exit or not? (1)

Give two further **computation paths**  $\pi_2$  and  $\pi_3$  such that each of the following conditions is satisfied by one of the paths  $\pi_1$ ,  $\pi_2$ , or  $\pi_3$ :

- One of the paths violates the chart.
- One of the paths is accepted and takes a legal exit.
- One of the paths is accepted without taking a legal exit. (2)

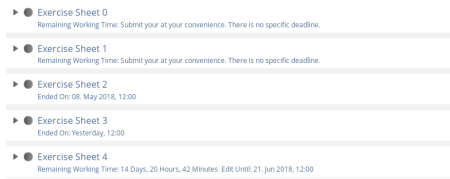
Use your examples to explain in your own words the intuition behind the concepts of acceptance with and without taking a 'legal exit', and of violation of a chart.

## Exercise 2 – (Creating LSCs)

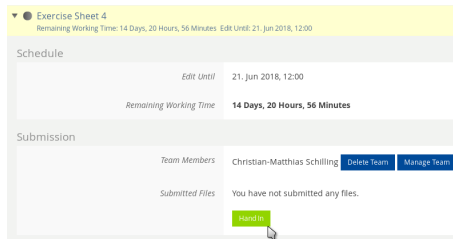
(5/20 Points)

name	exercise submission (preliminaries)
goal	exercise sheet solution uploaded
precondition	student navigated to exercise group screen in ILIAS
postcondition	student has submitted their solutions
actors	student (main actor), tutor
open questions	none
normal case	<ol style="list-style-type: none"> <li>1. student opens the current exercise sheet tab</li> <li>2. Ilias shows 'Hand In' button</li> <li>3. student clicks 'Hand In' button</li> </ol>
exception case 1a	<p>submission deadline exceeded</p> <ol style="list-style-type: none"> <li>2a.1 Ilias does not show 'Hand In' button due to deadline</li> <li>2a.2 student sends solutions to tutor via e-mail and explains why the submission is late</li> </ol>

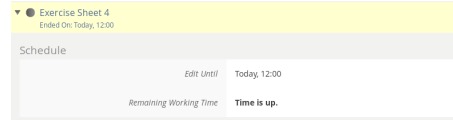
Create **universal Live Sequence Charts (LSC)** to specify our requirement on the Ilias system that, in order to accept the system, it must be possible to submit via 'Hand In' before the set deadline has exceeded, and that it must be possible to mail the submission to the tutor after the deadline has exceeded. The situation is illustrated in Figure 2.



(a) Display before clicking on the tab “Exercise Sheet 4”.



(b) Display after clicking on the tab “Exercise Sheet 4”, within the deadline limit.



(c) Display after clicking on the tab “Exercise Sheet 4”, after the deadline expiration.

Figure 2: Ilias screen example for the “exercise submission (preliminaries)” use case.

Available messages and conditions are shown in Figure 3. For simplicity assume that all messages are instantaneous.

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

### Exercise 3 – Class Diagrams and Object Diagrams (5/20)

Consider the class diagram shown in Figure 4.

- (i) Provide the **abstract syntax** of the diagram, i.e., present the class diagram as a *signature*. (2)
- (ii) The class diagram is supposed to model a clause data structure for Boolean formulae in conjunctive normal form, as it might be found in a SAT solver. Atom objects should link back to the Literal they are used by. In this case, it is undesirable to have any dangling references (functions and methods working on the clause data structure need to ensure this condition).

Source	Destination	Message	Description
Student	Ilias	<i>open_tab</i>	click on exercise in the list of exercises
Ilias	Student	<i>show_submit_form</i>	display exercise screen with ‘Hand In’ button
Ilias	Student	<i>show_DL_info</i>	display exercise screen with deadline info only
Student	Ilias	<i>submit</i>	submit solution via ‘Hand In’
Student	Ilias	<i>mail_tutor</i>	mail submission to tutor

Condition	Description
<i>on_exercises_screen</i>	student is on ‘exercises’ screen in their Ilias group
<i>past_DL</i>	deadline exceeded; current system time is later than deadline

Figure 3: Messages and conditions that may be used for LSC creation.

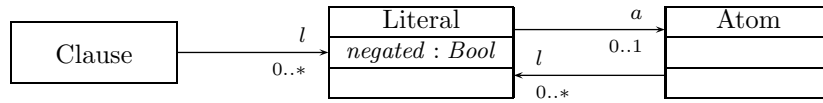


Figure 4: Class diagram for a clause representation.

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

- two instances of Clause,
- three instances of Literal, one of which is negated and one of which is not negated,
- at least one shared instance of Atom.

Give the *Basic System Object Structure* that your system state is based on. (2)

(iii) Provide another system state that illustrates an undesirable use of the data structure. (1)

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