Expert interview – Instructions

Thanks for participating in the evaluation of the Master thesis „Mode Extraction from Blech“.

The goal of this thesis was to extract the implicitly contained states and transitions of the Blech code and visualize them in a graph. For visualization purposes, the visual programming language SCCharts was used. SCCharts use a statechart notation. If you need some advice on how to read Statecharts in general, consult this page. For SCCharts specific information, visit this link.

This folder contains subfolders that cover different scenarios. They contain generated SCCharts (.sctx), abstractly visualizing Blech code and the corresponding Blech code (.blc) that the SCCharts are generated from.

To examine the generated SCCharts, please install a SCCharts-compatible editor such as KIELER or KEITH and import the .sctx-files. Keep in mind, that the visual representation of the sctx-file is of interest and not the code inside the sctx-files.

You might find hierarchical states such as referenced SCCharts, representing run statements and the corresponding called activity. Such hierarchical states can be expanded (and collapsed by double clicking).

This document will take you through the folders, explains what is shown and asks you questions. Feel free to submit your answers as you like and send them to daniel.lucas@stu.uni-kiel.de. I recommend adding comments for each question into the pdf and submit the commented pdf.

Most sctx-files will contain the original code as a comment, so it will be enough to import the sctx-file, view the Blech code that is present as a comment and then look at the visualization. For the bigger programs, the comment would be too enormous. Thus, in the directories 01_proposal and 02_programms, you will need to open the original Blech code in their respective files. It is recommended to use an editor that supports Blech syntax highlighting.

01_proposal

This folder contains the visualization of the code that was present in the proposal of the project. Further, it contains a pdf-file called *proposal-vis*, showing the envisioned intuitive drawing that was given in the project's proposal. First, look at the generated version (sctx-file).

Please answer the following questions:

1) Is this a viable representation of the given Blech code? If not, please explain why.
2) Does the visualization help to understand the stateful nature of the code?

Next, I want you to look at the pdf-file containing the initial intuitive example and compare it to the version you examined before. Please answer the following questions:

3) Which version do you like better and why?
4) Would you like to see the identified differences implemented in the automatic generation in the future?
5) Do you have any other thoughts on the matter? Is there something you would have made different? Are there things that surprised you?

02_programs

This code contains other actual (public) Blech programs. These are multiple programs that are somewhat complicated. Try to inspect them and try to determine if the visualization fits the programs. The original source is here.

Please answer the following questions:

1) Are these viable representations of the given Blech code? If not, please explain why.
2) Does the visualization help to understand the stateful nature of the code?
3) Do you have any other thoughts on the matter? Is there something you would have made different? Are there things that surprised you?

03_abort

This folder considers the illustration and simplification of abort blocks.

Abort.blc.sctx contains the translation including a flattening step of the hierarchy that is visible in *abort_hierarchical.blc.sctx*.

Please answer the following questions:

1) In your opinion, are the two options viable representations of the given Blech code? If not, please explain why.
2) Which of the two options do you prefer? And why? Would your opinion change if the inner graph inside the abort statement were more complicated?
3) Do you have any other thoughts on the matter? Is there something you would have made different? Are there things that surprised you?

04_run

This folder considers the flattening of activities in a run statement.

Run_flattened.blc.sctx shows the flattened hierarchy, whereas *run_hierarchical.blc.sctx* shows the unbroken hierarchy.

Please answer the following questions:

1) In your opinion, are the two options viable representations of the given Blech code? If not, please explain why.
2) Which of the two options do you prefer and why? Would your opinion change if the inner graph inside the abort statement were more complicated?
3) Do you have any other thoughts on the matter? Is there something you would have made different? Are there things that surprised you?

05_weak_abort_pattern

This folder considers the flattening of a certain pattern in cobegin constructs that models a weak abort. A branch that only awaits one condition may terminate another branch that is weak and thus aborted, as soon as the other condition is met.

Weak_abort_pattern_hierarchical.blc.sctx shows the pattern with unbroken hierarchy. *Weak_abort_pattern_flattened.blc.sctx* shows an option where the non-awaiting branch is elevated in the graph and the weak-abort condition is added to it. *Weak_abort_pattern_alt_flattened.blc.sctx* shows an alternative way of representing the weak abort.

Please answer the following questions:

1) In your opinion, are the three options viable representations of the given Blech code? If not, please explain why.
2) Which of the three options do you prefer and why? Would your opinion change if the inner (non-awaiting) graph inside the weak branch were more complicated?
3) Do you have any other thoughts on the matter? Is there something you would have made different? Are there things that surprised you?

06_labels

This folder considers different options of using labels. Labels.blc.sctx considers a version that is focussed on the states that were to be extracted in this project. Labels are merged for one identified state. Labels_EXT.blc.sctx shows a version that is more flexible and allows to label mostly unflattened complex states such as cobegins and its regions.

Please answer the following questions:

1) In your opinion, are the two options viable representations of the given Blech code. If not, please explain why.
2) Which of the two options do you prefer and why?
3) Would you use the extended label functionality?
4) Do you have any other thoughts on the matter? Is there something you would have made different? Are there things that surprised you?