

This short text describes the contents of each directory and the relationships between them.

DataExamples

Most of the files in this directory contain AMN models, which are examples of DES generally drawn from papers in literature on the domain. They are used by AMN files of other directories as input data. They also contains the outputs of some tests.

ExamplesPaper

This directory includes AMN files related to subjects introduced in each section of the paper. The name of each file refers to a particular section. Most of the files are self-contained (they, however, require the files for the definitions of theories), except:

- the file `section4_3_trains.mch` that refers to the directory `Trains`;
- the file `section5_1_5_2.mch` that refers to the directory `DataExamples`;
- the files `section5_3_ex4_MC_app.mch` and `section5_3_ex4_SMCP_i.mch` that refer to the directory `Trains`;
- the file `section5_4_find_ex6_tunnels.mch` that refers to the directory `Trains`;
- the file `section7_1_DEDS_12_3_H_Fig8b_11bPartition.mch` that refers to the directory `DataExamples`.

The output of the execution is included at the end of each file.

Maze (should not be modified)

This directory contains all AMN models related to the maze examples. They are used by AMN files of other directories as input data.

The subdirectory `Maze-json` contains JSON files used in product of automata (see the program `op-composition.rkt`) in the subdirectory `ModularProductionSystem/RacketOperations`. The file `mazes_induced.LB.mch` also appeared in this subdirectory. It generates parts of the legal behaviors of mazes, which need to be manually finalized. The final files are named `LBmaze_LB-[5,8]-rooms.mch` in the upper level of this directory.

ModularProductionSystem

This directory contains all files for the example of the modular production system (MPS) whose the results appeared in Tables 5 and 6. It is subdivided into three subdirectories:

- `ElementaryComponent` (should not be modified) which includes: i) all AMN models of elementary components; ii) a file to verify properties (e.g., controllability, nonblocking) and generate SFBC functions in the state-based formulation of the theory; and iii) two files to verify properties (e.g., controllability, relative closure, consistency of marking, observer property, control coincidence, and nonblocking of the interface) and generate the state realization of the supervisor in the language-based formulation of the theory.
- `CompositeComponent` (should not be modified) which includes: i) all AMN models of composite components; ii) a file to compute the supremal controllable and coreachable predicate, generate SFBC functions, verify the nonconflicting of control in the state-based formulation of the theory; iii) four files to verify properties (e.g., consistency of marking, observer property, control coincidence, and nonblocking of the interface) in the language-based formulation of the theory; and iv) a file to verify that the stations of the MPS are locally modular.
- `RacketOperations` (should not be used, except for new DESs) which contains JSON files used in instantiation of interfaces of elementary and composite components (see the RACKET program `op-instantiation.rkt`) and product of component automata (see the RACKET program `op-composition.rkt`). The final files are in the *CompositeComponent* subdirector.

An example of instantiation by RACKET:

```
Enter a directory name for the composite component: hook_e
Enter a directory name for the component library: LibraryOfComponentInterfaces
Enter a file name for a reusable component (without extension): jack111
Name: (jack111)
Atomic states: (Je Jie Jir Jr)
States: ((Je) (Jie) (Jir) (Jr))
Initial state: (Jr)
Marked states: ((Jr))
Events: (eoe eor extend retract)
Controllable events: (extend retract)
Transitions: (((Jr) extend (Jie)) ((Jie) eoe (Je)) ((Je) retract (Jir)) ((Jir) eor (Jr)))
Enter a file name for instantiation (without extension): jack111-winch
Name: (jack111-winch)
States: ((Je We) (Jie Wie) (Jir Wir) (Jr Wr))
Events: ((eoe W_eod) (eor W_eoa) (extend W_gd) (retract W_gu))

Enter a file name for the instantiated component (without extension): winch
End of processing.
```

An example of composition (product) of automata by RACKET:

```
Enter the number of reusable components: 2
Enter a directory name for the composite component: ../../Maze/Maze-json
Enter a file name for a reusable component (without extension): cat5rooms
Enter a file name for a reusable component (without extension): mouse5rooms
Computing a product of labeled transition systems ...
Number of states: 25
Number of events: 14
Number of transitions: 70

Creating the B machine for the new component from the data...

Enter a file name for the B machine (without extension): cat_mouse_5rooms
End of processing.
```

Tables1_2_3_4_7

This directory contains all AMN files to conduct experiments whose results appear in Tables 1 to 4 and 7. They refer to AMN models in directories Maze, Trains, and DataExamples (only used by the files in the directory Table7).

In the directory Table2, the file tab2_trains_sym.mch requires to assign the value of N in the table to constants bound_G_LB and bound_G_FB. These bounds are respectively defined in files LBtrains_FB_[8,10,12]_2 and LBtrains_LB_[1,2]_[8,10,12]_2. The outputs of experiments are in the files whose names are prefixed by output.

In the directory Table3, the AMN files consider iterative and recursive versions for computing the supremal subpredicate. The execution times in Table 3 correspond to the iterative version. The outputs are in the files whose names are prefixed by output.

Tests

This directory contains AMN test files for testing specific aspects of the theories. They are not directly related with the paper.

Theories

This directory contains the AMN files for some fragments of the supervisory control theory. It is the main directory of the project since it is referred by the most AMN files of the other directories.

Trains

This directory contains all AMN models related to the train examples. They are used by AMN files of other directories as input data.

The subdirectory `Trains-json` contains JSON files used in instantiation of an automaton (see the RACKET program `op-instantiation.rkt`) and product of automata (see the RACKET program `op-composition.rkt`) in the subdirectory *ModularProductionSystem/RacketOperations*. The file `trains_induced_LB.mch` also appeared in this subdirectory. It generates parts of the legal behaviors of trains, which need to be manually finalized. The final files are named `LBtrains_LB_[1,2]_[4,6,8,10,12]_2.mch` in the upper level of this directory. Such an operation requires to select between the predicates `Qpred1` and `Qpred2` in the files `SBtrains_[4,6,8,10,12]_2`.