

Testing Mist

Pedro Valero

14/15 C1

Índice general

1	Wednesday, 29 October 2014	2
1.1	I installed misti in my computer	2
1.2	Testing program	2
2	Thursday, 30 October 2014	2
2.1	Analyzing the output	2
2.2	Running all tests	3
3	Friday, 30 October 2014	3
3.1	Collecting results of the tests	3
3.2	Learning something about Petri nets	5
4	Monday, 3 November 2014	5
4.1	Updating python script	5

1. Wednesday, 29 October 2014

1.1. I installed misti in my computer

Following the steps given on the README I executed the followed commands:

1. `./autogent.sh.`

Executed without problems

2. `./configure.`

Executed without problems

3. `make.`

It trhows an error because the command `yacc` was not found.

I installed the package *bison* which includes this command and repeat this step.

Now, it seems to work.

4. `make install`

Although this step is optional I did it to check if all works fine.

Throws a permission denied error.

I executed this command again with superuser privileges and it seem to work.

Note: For some commands it didn't do anything:

- `install-data-am`
- `install-exec-am`
- `install-data-am`

1.2. Testing program

I wrote a python script which takes all the files in the folder given as argument with extension `.spec` and uses mist over them.

Right now the script seems to work but I'm not sure about how do I have to interpret the output to recognise when the targe has been reached or not.

2. Thursday, 30 October 2014

2.1. Analyzing the output

I take a look at <http://www.ulb.ac.be/di/ssd/ggeeraer/eec/tableau/> and I found some differences between the results given there and results obtained in my computer. For example:

- The example **Java** is said to be not safe in the table but when I executed the tool with it the output says:
"EEC concludes safe with the abstraction"
"backward algorithm concludes unsafe"
"TSI concludes safe"
- The example **MOESI** is said to be safe but when I executed it in my computer I obtained:
"Cannot assign non null value to a variable in transition 5"
for any algorithm used.

2.2. Running all tests

After see this I modified my script to take the conclusions given by each algorithm for each test given in the folder 'examples'. After check with some examples that the script seems to work, I let the computer working all night.

3. Friday, 30 October 2014

3.1. Collecting results of the tests

Although the computer has been working all night it didn't end the job. To collect, at least, a first approximation of the tests I forced to end this executions of the tools which takes too long.

The executions that I forced to end by keyboard interrupt are:

- `mist --tsi ../examples/PN/bingham_h150.spec`
- `mist --backward ../examples/PN/bingham_h250.spec`
- `mist --tsi ../examples/PN/bingham_h250.spec`
- `mist --eec ../examples/PN/extendedread-write-smallconsts.spec`
- `mist --tsi ../examples/PN/extendedread-write-smallconsts.spec`
- `mist --eec ../examples/PN/extendedread-write.spec`
- `mist --tsi ../examples/PN/extendedread-write.spec`
- `mist --eec ../examples/PN/fms_attic.spec`
- `mist --eec ../examples/PN/kanban.spec`
- `mist --eec ../examples/PN/mesh3x2.spec`

- `mist --tsi ../examples/PN/mesh3x2.spec`

Apart from this, there are some examples whose output didn't match with the results given in <http://www.ulb.ac.be/di/ssd/ggeeraer/eec/tableau/> or, at least, cause some confusion.

These examples are:

1. There are so many examples that **throws an error**:

"Cannot assign non null value to a variable in transition x"

- MOESI
- last-in-first-served
- german-protocol
- berkeley
- dragon
- futurebus
- illinois

The problem is the same for each algorithm applied

2. Other examples causes a **segmentation fault**:

- fms backward
- kanban backward
- newrtsp backward
- firefly tsi

3. The examples 'german', 'consprod', 'delegatebuffer', 'examplelea', 'transthesis', **should be safe** but the results obtained by the tool are:

```

''EEC concludes unsafe''
''backward algorithm concludes safe''
''TSI concludes unsafe''

```

4. The examples 'simplejavaexample' and 'Java' **should be unsafe** but the results are:

```

''EEC concludes safe''
''backward algorithm concludes unsafe''
''TSI concludes safe''

```

5. The example 'kanban' **should be safe** but the results are:

```

EEC keyboard interrupt (too long)
''backward algorithm concludes unsafe''
''TSI concludes unsafe''

```

3.2. Learning something about Petri nets

I started to look for information about Petri nets to get a better knowledge about what the tool does.

I got the needed information from http://en.wikipedia.org/wiki/Petri_net and <http://condor.depaul.edu/rjohnson/dm6th/petri.pdf>.

After reading it I think that the tool give us some results which are inconsistent with the older ones.

Until Pierre could give me an answer about if there is a problem with the tool or I'm wrong, I'm going to try to understand how does the tool works and what does exactly do.

4. Monday, 3 November 2014

4.1. Updating python script

After Pierre give me an explanation about how to interpret the results given by 'mist' I made a new python script to give us a structured summary about the consistence of the mist's results in relation with the results from the web.

I also collect the official results into a file to be used by the script. The file which contains these results is called "expected_results.txt"