

Gentle Introduction to the Web-Interface for SPIN

Jesús Mauricio Chimento

Getting started

examples - cheat sheet Web-interface for Spin sources - about

```
promela_e3.pml
```

```
1 bool b;
2
3 proctype TRUE() {
4   if
5     :: b -> printf("1\n")
6     :: true -> printf("2\n")
7   fi
8 }
9
10 proctype ELSE() {
11   if
12     :: b -> printf("1\n")
13     :: else -> printf("2\n")
14   fi
15 }
16
17 init {
18   run TRUE();
19   run ELSE();
20 }
```

Global variables Local variables Statements
 Receive events Send events Verbose

Stop after steps

Mode:
 Weak fairness Run generated trail
 Don't check for invalid endstates
Name of LTL formula:

Minimal mode

```
spin -g -l /tmp/spinfolder_odxL6X/pmlfileurlIvJ

1
1
3 processes created
```

Getting started

examples - cheat sheet Web-interface for Spin sources - about

SPIN Box

<input checked="" type="checkbox"/> Global variables	<input checked="" type="checkbox"/> Local variables	<input type="checkbox"/> Statements
<input type="checkbox"/> Receive events	<input type="checkbox"/> Send events	<input type="checkbox"/> Verbose
<input type="checkbox"/> Stop after <input type="text" value="250"/> steps		<input type="button" value="Interactive"/>
<input type="button" value="Run"/>		
Mode: <input type="text" value="Safety"/>		
<input type="checkbox"/> Weak fairness	<input checked="" type="checkbox"/> Run generated trail	
<input type="checkbox"/> Don't check for invalid endstates		
Name of LTL formula: <input type="text"/>		<input type="button" value="Verify"/>
<input type="checkbox"/> Minimal mode		

```
promela_e3.pml +
1 bool b;
2
3 proctype TRUE() {
4   if
5     :: b -> printf("1\n")
6     :: true -> printf("2\n")
7   fi
8 }
9
10 proctype ELSE() {
11   if
12     :: b -> printf("1\n")
13     :: else -> printf("2\n")
14   fi
15 }
16
17 init {
18   run TRUE();
19   run ELSE();
20 }
```

Coding Box

```
spin -g -l /tmp/spinfolder_odxL6X/pmlfileurlIvJ
1
1
3 processes created
```

Output Box

Getting started

examples - cheat sheet Web-interface for Spin sources - about

```
promela_e3.pml
```

```
1 bool b;
2
3 proctype TRUE() {
4   if
5     :: b -> printf("1\n")
6     :: true -> printf("2\n")
7   fi
8 }
9
10 proctype ELSE() {
11   if
12     :: b -> printf("1\n")
13     :: else -> printf("2\n")
14   fi
15 }
16
17 init {
18   run TRUE();
19   run ELSE();
20 }
```

Global variables Local variables Statements
 Receive events Send events Verbose

Stop after steps

Mode:
 Weak fairness Run generated trail
 Don't check for invalid endstates
Name of LTL formula:

Minimal mode

```
spin -g -l /tmp/spinfolder_odxL6X/pmlfileurlIvJ

1
1
3 processes created
```

Getting started

examples - cheat sheet Web-interface for Spin sources - about

promela_e3.pml

```
1 bool b;
2
3 proctype TRUE() {
4   if
5     :: b -> printf("1\n")
6     :: true -> printf("2\n")
7   fi
8 }
9
10 proctype ELSE() {
11   if
12     :: b -> printf("1\n")
13     :: else -> printf("2\n")
14   fi
15 }
16
17 init {
18   run TRUE();
19   run ELSE();
20 }
```

SPIN as PROMELA simulator

Global variables Local variables Statements
 Receive events Send events Verbose

Stop after steps

Mode:
 Weak fairness Run generated trail
 Don't check for invalid endstates
Name of LTL formula:

Minimal mode

```
spin -g -l /tmp/spinfolder_odxL6X/pmlfileurlIvJ

1
1
3 processes created
```

Getting started

examples - cheat sheet Web-interface for Spin sources - about

```
promela_e3.pml x +
1 bool b;
2
3 proctype TRUE() {
4   if
5     :: b -> printf("1\n")
6     :: true -> printf("2\n")
7   fi
8 }
9
10 proctype ELSE() {
11   if
12     :: b -> printf("1\n")
13     :: else -> printf("2\n")
14   fi
15 }
16
17 init {
18   run TRUE();
19   run ELSE();
20 }
```

SPIN as PROMELA simulator

SPIN as Model Checker

Global variables Local variables Statements
 Receive events Send events Verbose

Stop after steps Interactive

Run

Mode:

Weak fairness Run generated trail
 Don't check for invalid endstates

Name of LTL formula: Verify

Minimal mode

```
spin -g -l /tmp/spinfolder_odxL6X/pmlfileurlIvJ
1
1
3 processes created
```

SPIN as PROMELA simulator

<input checked="" type="checkbox"/> Global variables	<input checked="" type="checkbox"/> Local variables	<input type="checkbox"/> Statements
<input type="checkbox"/> Receive events	<input type="checkbox"/> Send events	<input type="checkbox"/> Verbose
<input type="checkbox"/> Stop after <input type="text" value="250"/> steps	<input type="button" value="Interactive"/>	
<input type="button" value="Run"/>		
Mode: <input type="text" value="Safety"/>		
<input type="checkbox"/> Weak fairness	<input checked="" type="checkbox"/> Run generated trail	
<input type="checkbox"/> Don't check for invalid endstates		
Name of LTL formula: <input type="text"/>	<input type="button" value="Verify"/>	
<input type="checkbox"/> Minimal mode		

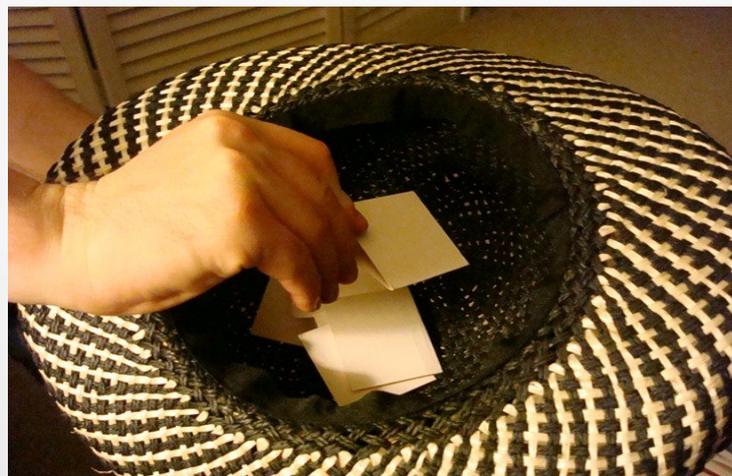
SPIN as PROMELA simulator

<input checked="" type="checkbox"/> Global variables	<input checked="" type="checkbox"/> Local variables	<input type="checkbox"/> Statements
<input type="checkbox"/> Receive events	<input type="checkbox"/> Send events	<input type="checkbox"/> Verbose
<input type="checkbox"/> Stop after <input type="text" value="250"/> steps		<input type="button" value="Interactive"/>
<input type="button" value="Run"/>		

SPIN as PROMELA simulator

<input checked="" type="checkbox"/> Global variables	<input checked="" type="checkbox"/> Local variables	<input type="checkbox"/> Statements
<input type="checkbox"/> Receive events	<input type="checkbox"/> Send events	<input type="checkbox"/> Verbose
<input type="checkbox"/> Stop after <input type="text" value="250"/> steps		<input type="button" value="Interactive"/>
<input type="button" value="Run"/>		

- Run
 - runs a random simulation on your model
 - not aimed to find errors, but helpful to track down problems



SPIN as PROMELA simulator

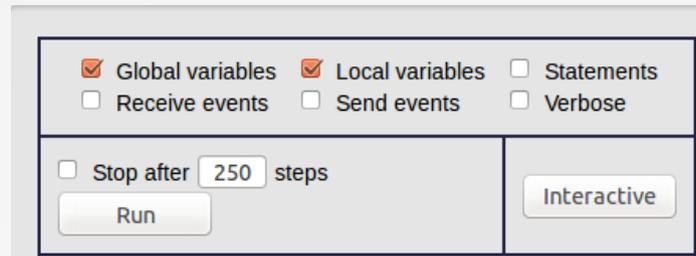
<input checked="" type="checkbox"/> Global variables	<input checked="" type="checkbox"/> Local variables	<input type="checkbox"/> Statements
<input type="checkbox"/> Receive events	<input type="checkbox"/> Send events	<input type="checkbox"/> Verbose
<input type="checkbox"/> Stop after <input type="text" value="250"/> steps		<input type="button" value="Interactive"/>
<input type="button" value="Run"/>		

- Interactive

- runs an interactive simulation on your model
- at each decision point where multiple paths can be taken, the user decides which one to follow



SPIN as PROMELA simulator



The image shows a screenshot of the SPIN simulator's options dialog box. It features a grid of checkboxes for various simulation options. The 'Global variables' and 'Local variables' options are checked, while 'Statements', 'Receive events', 'Send events', and 'Verbose' are unchecked. Below the checkboxes, there is a 'Stop after' field with a text input containing '250' and the unit 'steps'. A 'Run' button is located to the left of the 'Interactive' button.

<input checked="" type="checkbox"/> Global variables	<input checked="" type="checkbox"/> Local variables	<input type="checkbox"/> Statements
<input type="checkbox"/> Receive events	<input type="checkbox"/> Send events	<input type="checkbox"/> Verbose
<input type="checkbox"/> Stop after <input type="text" value="250"/> steps		<input type="button" value="Interactive"/>
<input type="button" value="Run"/>		

- Options

- **Global Variables:** shows the value of the global variables
- **Local Variables:** shows the value of the local variables
- **Statements:** shows step-by-step-like which statements are runned (this option should be selected in order to use the previous ones)
- **Stop After [amount] steps:** Stop the execution of the model after the indicated amount of steps

SPIN as Model Checker

<input checked="" type="checkbox"/> Global variables	<input checked="" type="checkbox"/> Local variables	<input type="checkbox"/> Statements
<input type="checkbox"/> Receive events	<input type="checkbox"/> Send events	<input type="checkbox"/> Verbose
<input type="checkbox"/> Stop after <input type="text" value="250"/> steps	<input type="button" value="Interactive"/>	
<input type="button" value="Run"/>		
Mode: <input type="text" value="Safety"/>		
<input type="checkbox"/> Weak fairness	<input checked="" type="checkbox"/> Run generated trail	
<input type="checkbox"/> Don't check for invalid endstates		
Name of LTL formula: <input type="text"/>	<input type="button" value="Verify"/>	
<input type="checkbox"/> Minimal mode		

SPIN as Model Checker

Mode:

Weak fairness Run generated trail

Don't check for invalid endstates

Name of LTL formula:

SPIN as Model Checker

Mode:

Weak fairness Run generated trail

Don't check for invalid endstates

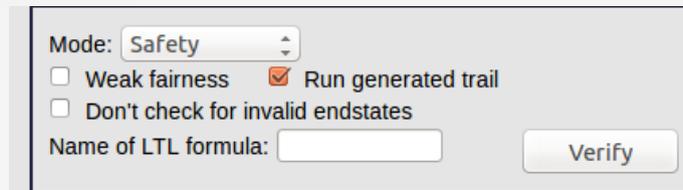
Name of LTL formula:

- **Verify**

- runs a full verification of the model
- if any error is spotted, you'll see a message in the **Output Box**

```
public client static void reFreshCallerDataSource(Object _callerFormDataSource,
boolean _recordExists,
boolean _doReSearch)
{
    if (_recordExists)
    {
        if (_doReSearch)
        {
            _callerFormDataSource.reSearch(true);
        }
        else
        {
            _callerFormDataSource.reRead();
            _callerFormDataSource.reFresh();
            if (_callerFormDataSourceHasMethod(_callerFormDataSource, identifierStr(reReadLines)))
            {
                _callerFormDataSource.reReadLines();
            }
            _callerFormDataSource.active();
        }
    }
}
```

SPIN as Model Checker



The screenshot shows a control panel for the SPIN model checker. It includes a dropdown menu for 'Mode' set to 'Safety', two checkboxes for 'Weak fairness' (unchecked) and 'Run generated trail' (checked), and another checkbox for 'Don't check for invalid endstates' (unchecked). Below these is a text input field for the 'Name of LTL formula:' and a 'Verify' button.

- Options

- **Mode:** specifies which kind of verification is performed (Safety by default)
- **Run generated trail:** runs a guided simulation using the trial file created by the execution of the analyzer
- **Name of LTL formula:** verifies a LTL formula of the model in particular

Conclusions

- Web-Interface introduction
- SPIN as PROMELA simulator
- SPIN as Model Checker