Métodos Formais em Engenharia de Software

Ana Paiva

apaiva@fe.up.pt

Universidade do Porto
Faculdade de Engenharia
FEUP

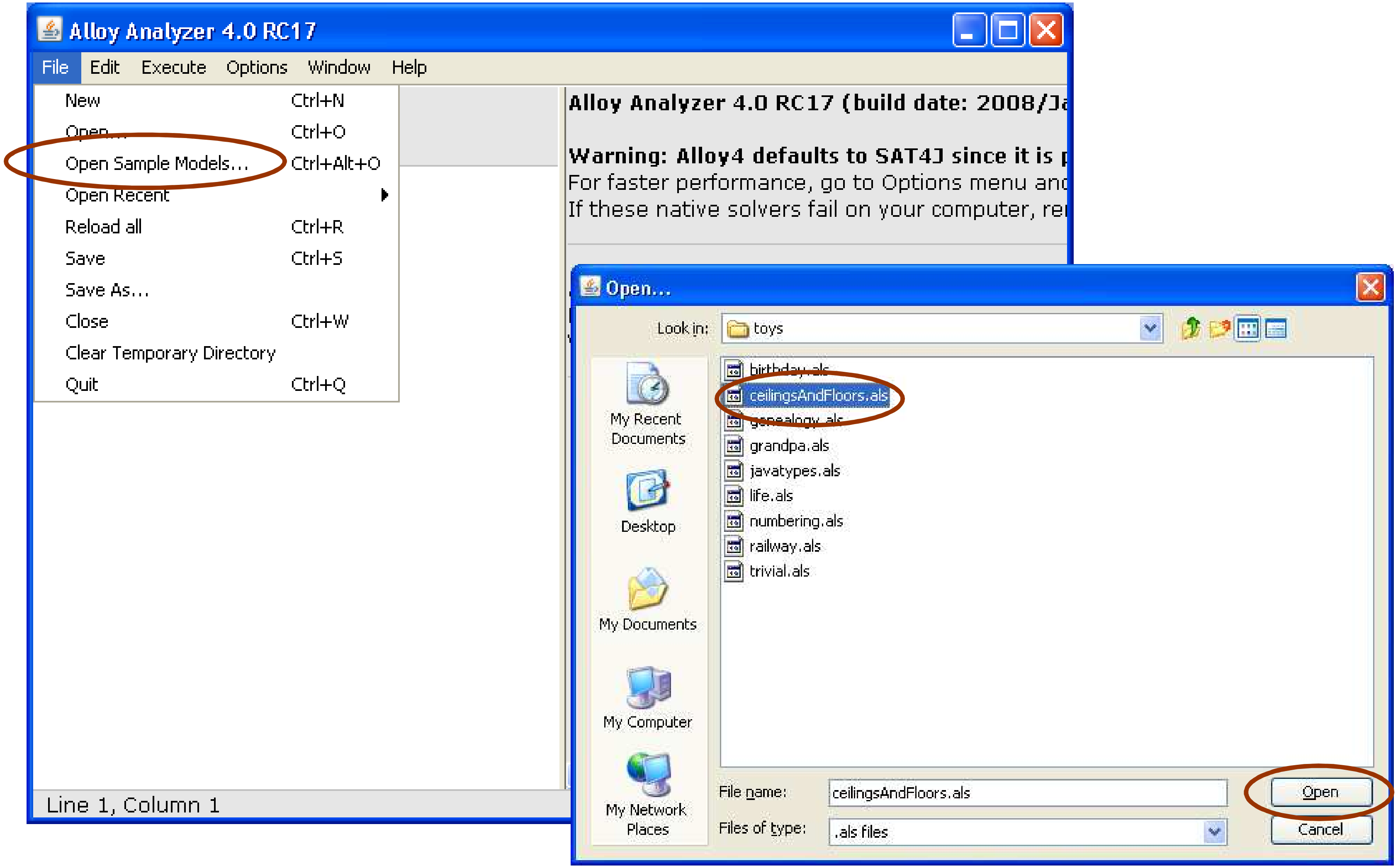
Alloy Analyzer: manual

- Run the tool
- The GUI of the tool
  - Verify properties of the models
  - View the result of the analysis
    - The Viz View
    - The Tree View
    - The XML View
  - Sintax
  - Usefull modules: “buil in”
  - How to use modules

Run Alloy Analyzer

```
java –jar alloy4.jar
```

Open models



Alloy Analyzer 4.0 RC17 (build date: 2008/Ja

Warning: Alloy4 defaults to SAT4J since it is p
For faster performance, go to Options menu and
If these native solvers fail on your computer, re

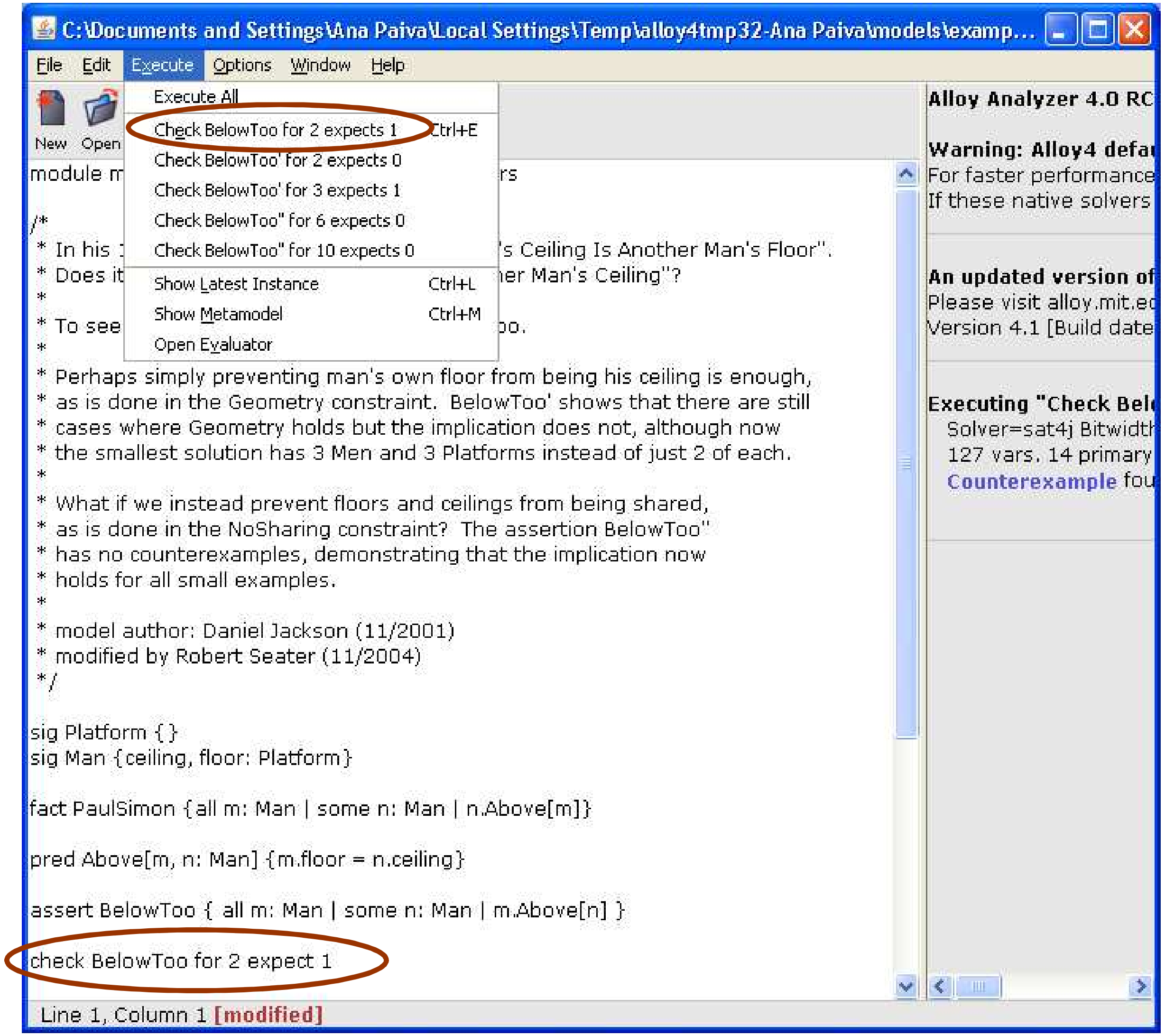
File name: ceilingsAndFloors.als

Files of type: .als files

Analysis of Alloy models

- The run command is used to find solutions that meet the specification and the predicate, while the check command is used to find solutions that meet the specification but violate an assertion.
- To run each of the possible analysis, select the appropriate command from the run menu.
- The menu shows the list of run checks and runs the commands in the model. You can run one command at a time or to run them all at once, all run
- The run button will re-execute the command previously executed. If no command has been executed so far, will run the first command of the model.
- The analysis ends with a solution or indicating that it is possible to find a solution within the state space defined by the limits imposed. If you can find a solution, it can be viewed by selecting the blue hyperlink that appears in the message pane. Or, if the option automatically view within the Options menu is active, then the solution will appear automatically.

Check properties (one by one)

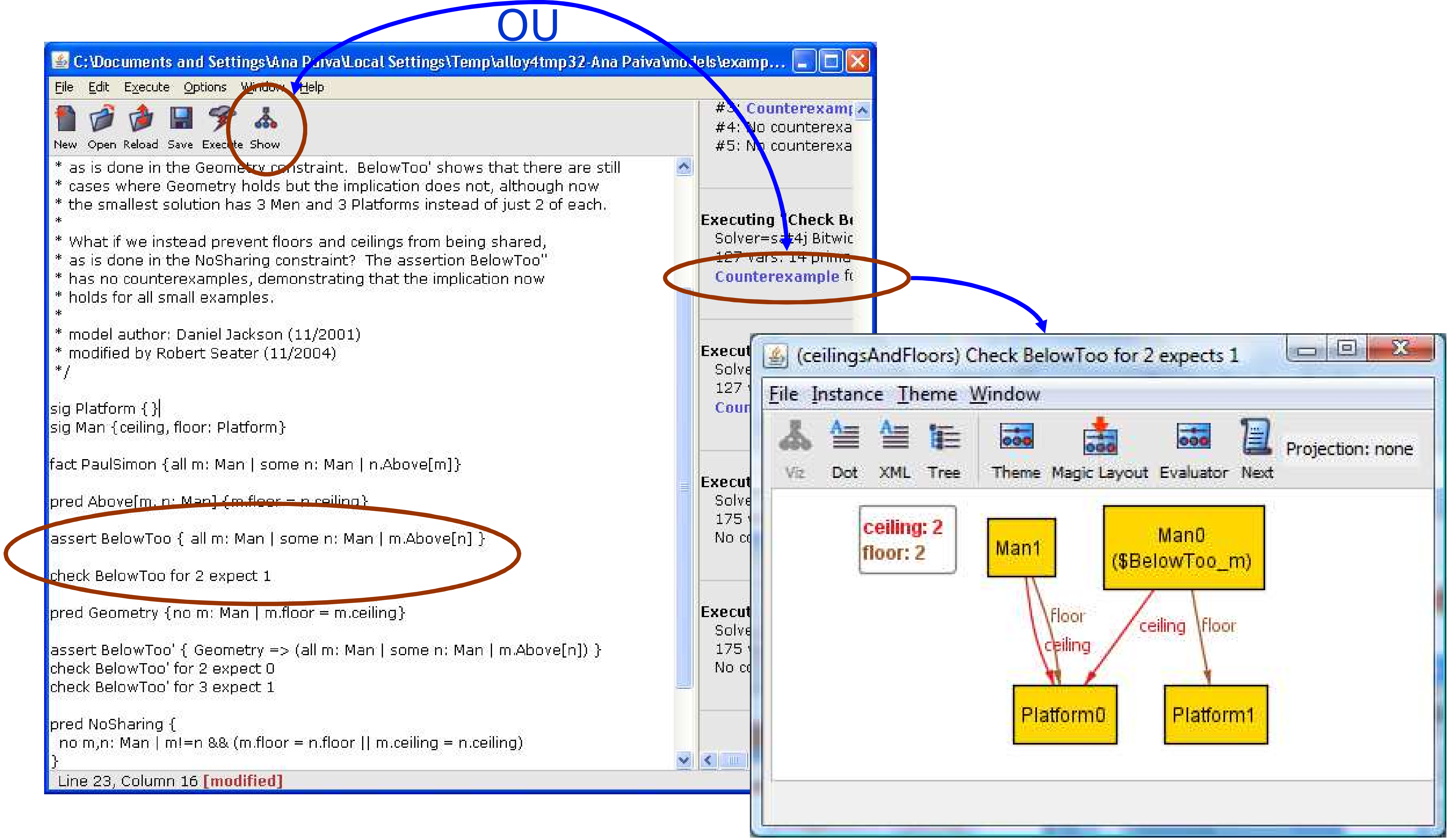

C:\Documents and Settings\Ana Paiva\Local Settings\Temp\alloy4tmp32-Ana Paiva\models\examp...
File
Edit
Execute
Options
Window
Help
New
Open
Execute All
Check BelowToo for 2 expects 1 Ctrl+E
Check BelowToo' for 2 expects 0
Check BelowToo' for 3 expects 1
Check BelowToo'' for 6 expects 0
Check BelowToo'' for 10 expects 0
Show Latest Instance Ctrl+L
Show Metamodel Ctrl+M
Open Evaluator

```
module m                                              rs

/*
 * In his                                             's Ceiling Is Another Man's Floor".
 * Does it                                            er Man's Ceiling"?
 *
 * To see                                             oo.
 *
 * Perhaps simply preventing man's own floor from being his ceiling is enough,
 * as is done in the Geometry constraint.  BelowToo' shows that there are still
 * cases where Geometry holds but the implication does not, although now
 * the smallest solution has 3 Men and 3 Platforms instead of just 2 of each.
 *
 * What if we instead prevent floors and ceilings from being shared,
 * as is done in the NoSharing constraint?  The assertion BelowToo''
 * has no counterexamples, demonstrating that the implication now
 * holds for all small examples.
 *
 * model author: Daniel Jackson (11/2001)
 * modified by Robert Seater (11/2004)
 */

sig Platform {}
sig Man {ceiling, floor: Platform}

fact PaulSimon {all m: Man | some n: Man | n.Above[m]}

pred Above[m, n: Man] {m.floor = n.ceiling}

assert BelowToo { all m: Man | some n: Man | m.Above[n] }

check BelowToo for 2 expect 1
```

```
Alloy Analyzer 4.0 RC

Warning: Alloy4 defau
For faster performance
If these native solvers

An updated version of
Please visit alloy.mit.ed
Version 4.1 [Build date

Executing "Check Belo
  Solver=sat4j Bitwidth
  127 vars. 14 primary
  Counterexample fou
```

Line 1, Column 1 [modified]

See counter-example

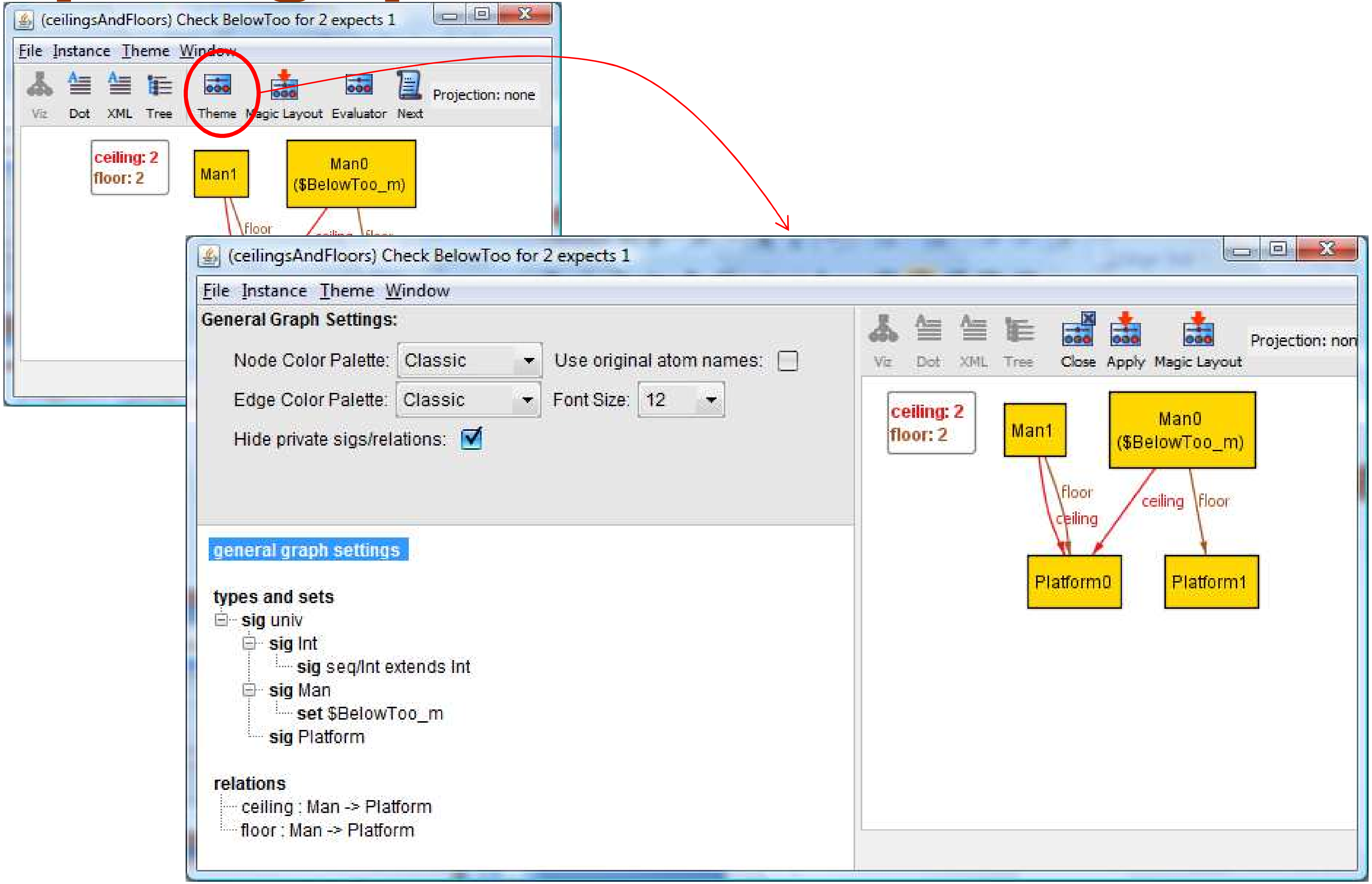

OU
C:\Documents and Settings\Ana Paiva\Local Settings\Temp\alloy4tmp32-Ana Paiva\models\examp...
File Edit Execute Options Window Help
New Open Reload Save Execute Show
(ceilingsAndFloors) Check BelowToo for 2 expects 1
File Instance Theme Window
Viz Dot XML Tree Theme Magic Layout Evaluator Next
Projection: none
ceiling: 2
floor: 2
Man1
Man0
($BelowToo_m)
Platform0
Platform1
floor
ceiling
ceiling
floor

```
 * as is done in the Geometry constraint.  BelowToo' shows that there are still
 * cases where Geometry holds but the implication does not, although now
 * the smallest solution has 3 Men and 3 Platforms instead of just 2 of each.
 *
 * What if we instead prevent floors and ceilings from being shared,
 * as is done in the NoSharing constraint?  The assertion BelowToo''
 * has no counterexamples, demonstrating that the implication now
 * holds for all small examples.
 *
 * model author: Daniel Jackson (11/2001)
 * modified by Robert Seater (11/2004)
 */

sig Platform {}
sig Man {ceiling, floor: Platform}

fact PaulSimon {all m: Man | some n: Man | n.Above[m]}

pred Above[m, n: Man] {m.floor = n.ceiling}

assert BelowToo { all m: Man | some n: Man | m.Above[n] }

check BelowToo for 2 expect 1

pred Geometry {no m: Man | m.floor = m.ceiling}

assert BelowToo' { Geometry => (all m: Man | some n: Man | m.Above[n]) }
check BelowToo' for 2 expect 0
check BelowToo' for 3 expect 1

pred NoSharing {
  no m,n: Man | m!=n && (m.floor = n.floor || m.ceiling = n.ceiling)
}
Line 23, Column 16 [modified]
```

```
#3: Counterexamp
#4: No counterexa
#5: No counterexa

Executing "Check Be
  Solver=sat4j Bitwic
  127 vars. 14 prima
  Counterexample fo

Execut
  Solve
  127
  Coun

Execut
  Solve
  175
  No co

Execut
  Solve
  175
  No co
```

Man0.Floor <> Man1.celling

Update graphical visualization


(ceilingsAndFloors) Check BelowToo for 2 expects 1
File Instance Theme Window
Viz Dot XML Tree Theme Magic Layout Evaluator Next
Projection: none
ceiling: 2
floor: 2
Man1
Man0 ($BelowToo_m)
floor
ceiling
(ceilingsAndFloors) Check BelowToo for 2 expects 1
File Instance Theme Window
Viz Dot XML Tree Close Apply Magic Layout
Projection: non
ceiling: 2
floor: 2
Man1
Man0 ($BelowToo_m)
floor
ceiling
ceiling
floor
Platform0
Platform1

General Graph Settings:

Node Color Palette: Classic
Use original atom names: □
Edge Color Palette: Classic
Font Size: 12
Hide private sigs/relations: ☑

general graph settings

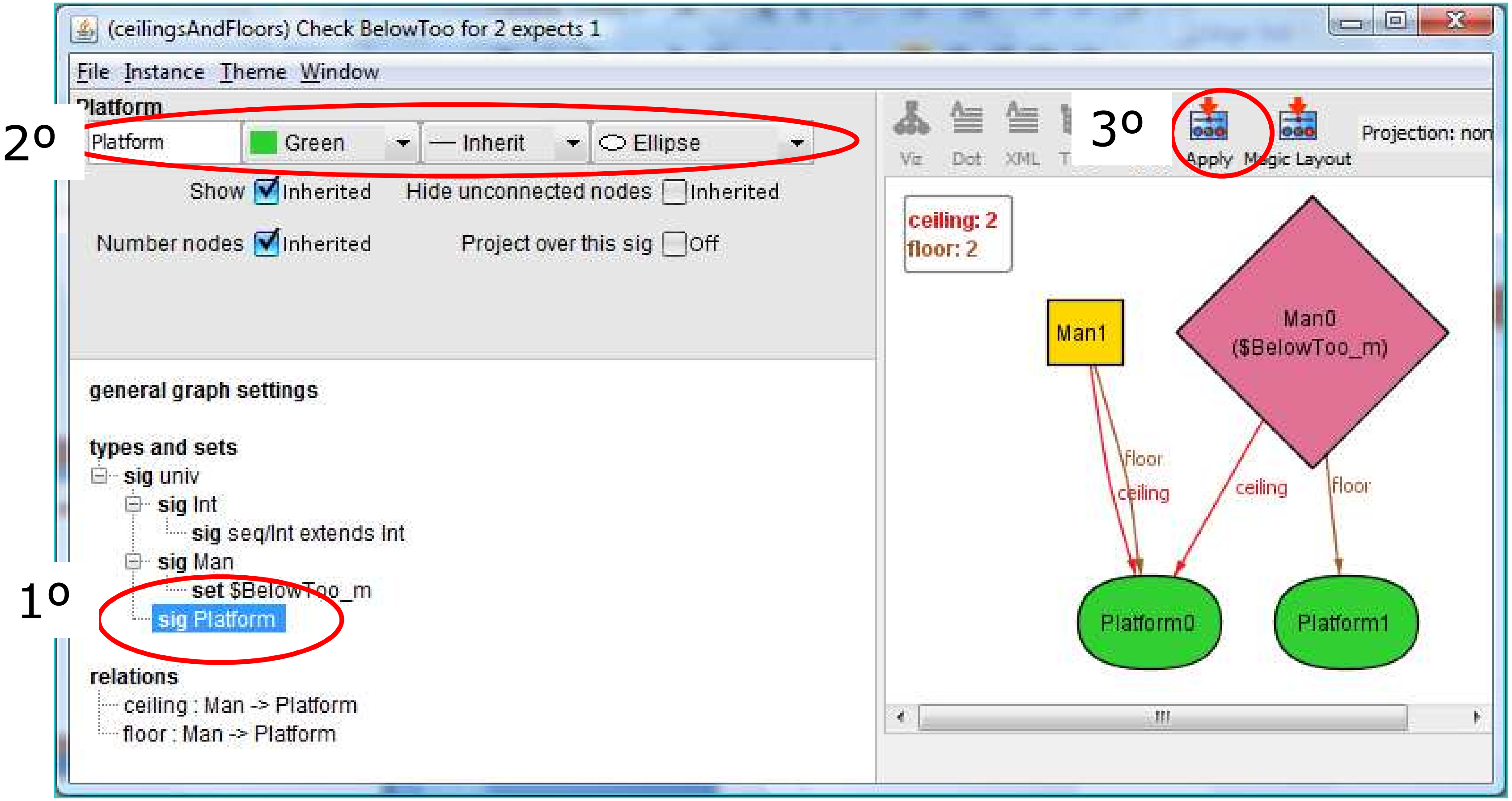
types and sets

```
sig univ
   sig Int
      sig seq/Int extends Int
   sig Man
      set $BelowToo_m
   sig Platform
```

relations

```
ceiling : Man -> Platform
floor : Man -> Platform
```

Update graphical visualization



(ceilingsAndFloors) Check BelowToo for 2 expects 1

File Instance Theme Window

Platform

2º

Platform | 🟩 Green | — Inherit | ○ Ellipse

Show ☑ Inherited   Hide unconnected nodes □ Inherited

Number nodes ☑ Inherited   Project over this sig □ Off

general graph settings

types and sets

```
sig univ
  sig Int
    sig seq/Int extends Int
  sig Man
    set $BelowToo_m
  sig Platform
```

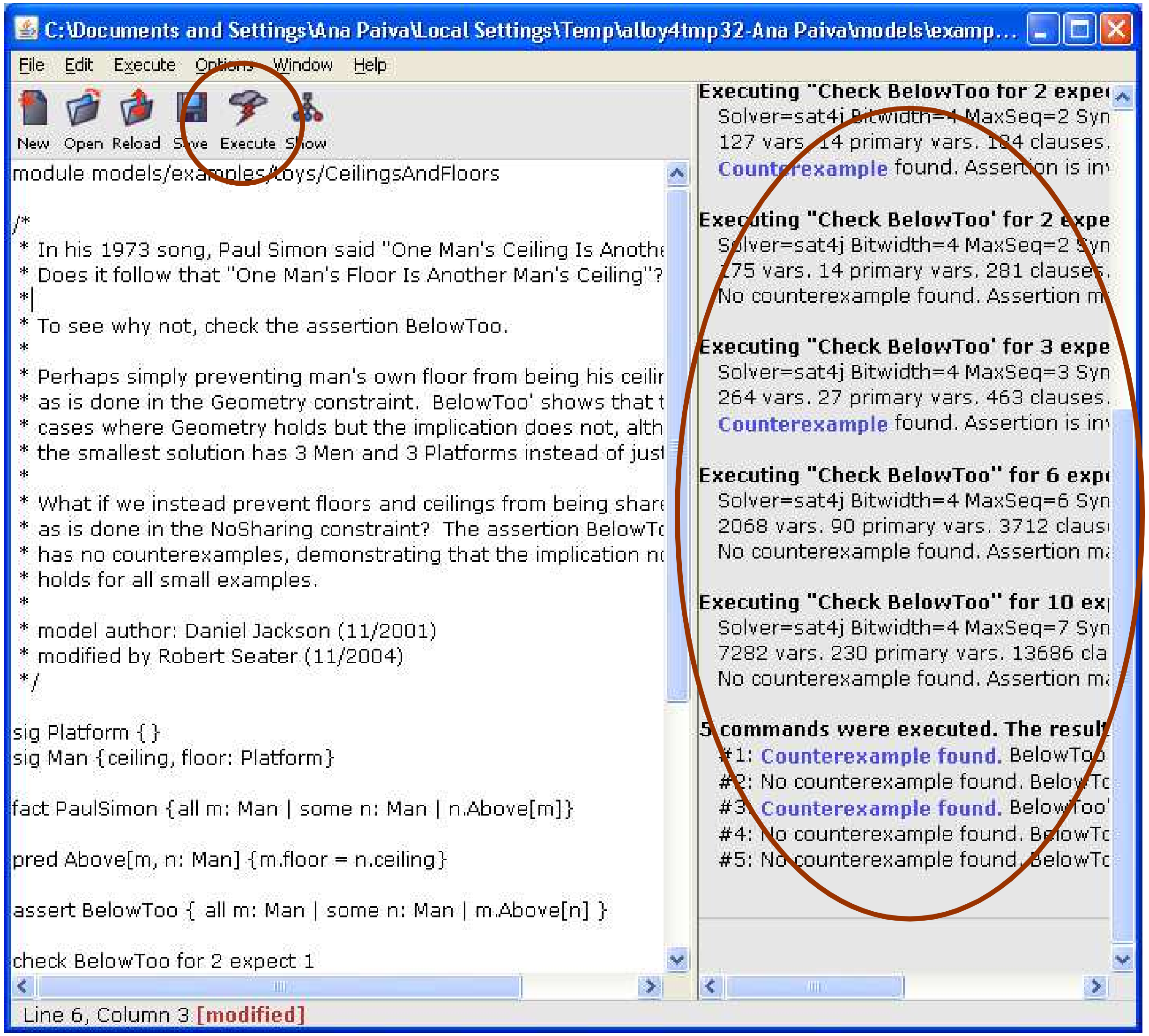
1º

relations

```
ceiling : Man -> Platform
floor : Man -> Platform
```

3º

Check properties (all at once)


C:\Documents and Settings\Ana Paiva\Local Settings\Temp\alloy4tmp32-Ana Paiva\models\examp...
File
Edit
Execute
Options
Window
Help
New
Open
Reload
Save
Execute
Show

```
module models/examples/toys/CeilingsAndFloors

/*
 * In his 1973 song, Paul Simon said "One Man's Ceiling Is Anothe
 * Does it follow that "One Man's Floor Is Another Man's Ceiling"?
 *
 * To see why not, check the assertion BelowToo.
 *
 * Perhaps simply preventing man's own floor from being his ceilir
 * as is done in the Geometry constraint.  BelowToo' shows that t
 * cases where Geometry holds but the implication does not, alth
 * the smallest solution has 3 Men and 3 Platforms instead of just
 *
 * What if we instead prevent floors and ceilings from being share
 * as is done in the NoSharing constraint?  The assertion BelowTo
 * has no counterexamples, demonstrating that the implication no
 * holds for all small examples.
 *
 * model author: Daniel Jackson (11/2001)
 * modified by Robert Seater (11/2004)
 */

sig Platform {}
sig Man {ceiling, floor: Platform}

fact PaulSimon {all m: Man | some n: Man | n.Above[m]}

pred Above[m, n: Man] {m.floor = n.ceiling}

assert BelowToo { all m: Man | some n: Man | m.Above[n] }

check BelowToo for 2 expect 1
```

```
Executing "Check BelowToo for 2 expe
   Solver=sat4j Bitwidth=4 MaxSeq=2 Syn
   127 vars. 14 primary vars. 124 clauses.
   Counterexample found. Assertion is in

Executing "Check BelowToo' for 2 expe
   Solver=sat4j Bitwidth=4 MaxSeq=2 Syn
   175 vars. 14 primary vars. 281 clauses.
   No counterexample found. Assertion m

Executing "Check BelowToo' for 3 expe
   Solver=sat4j Bitwidth=4 MaxSeq=3 Syn
   264 vars. 27 primary vars. 463 clauses.
   Counterexample found. Assertion is in

Executing "Check BelowToo'' for 6 exp
   Solver=sat4j Bitwidth=4 MaxSeq=6 Syn
   2068 vars. 90 primary vars. 3712 claus
   No counterexample found. Assertion m

Executing "Check BelowToo'' for 10 ex
   Solver=sat4j Bitwidth=4 MaxSeq=7 Syn
   7282 vars. 230 primary vars. 13686 cla
   No counterexample found. Assertion m

5 commands were executed. The result
   #1: Counterexample found. BelowToo
   #2: No counterexample found. BelowTo
   #3: Counterexample found. BelowToo'
   #4: No counterexample found. BelowTo
   #5: No counterexample found. BelowTo
```

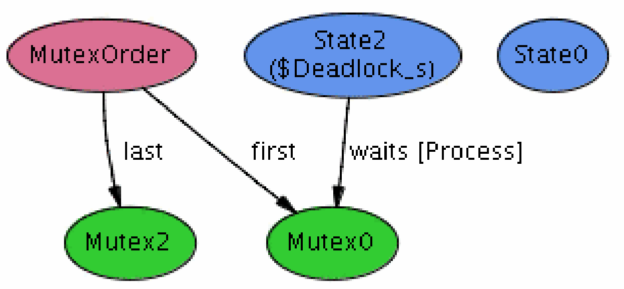
Line 6, Column 3 [modified]

How to see results

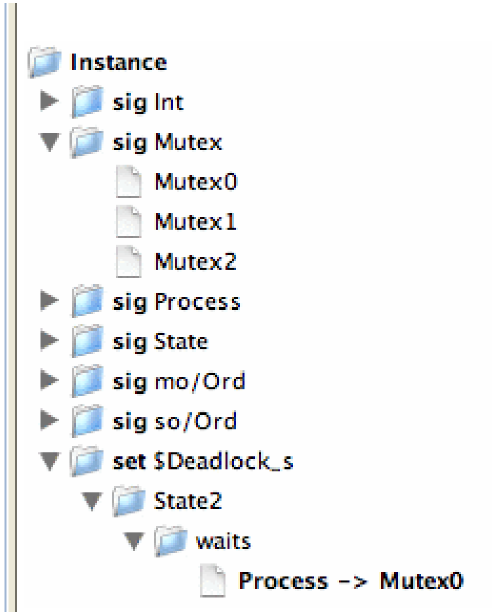
XML

```
<alloy>
<sig name="Name" extends="univ">
   <atom name="Name$0"/>
   <atom name="Name$1"/>
</sig>
<sig name="Date" extends="univ">
   <atom name="Date$0"/>
</sig>
<sig name="BirthdayBook" extends="univ">
   <atom name="BirthdayBook$0"/>
   <atom name="BirthdayBook$1"/>
</sig>
<field name="known">
   <type> <sig name="BirthdayBook"/><sig name="Name"/></type>
   <tuple><atom name="BirthdayBook$1"/><atom name="Name$1"/> </tuple>
</field>
<field name="date">
   <type><sig name="BirthdayBook"/><sig name="Name"/><sig name="Date"/></type>
   <tuple><atom name="BirthdayBook$1"/><atom name="Name$1"/><atom name="Date$0"/></tuple>
</field>
</instance>
</alloy>
```

Graph

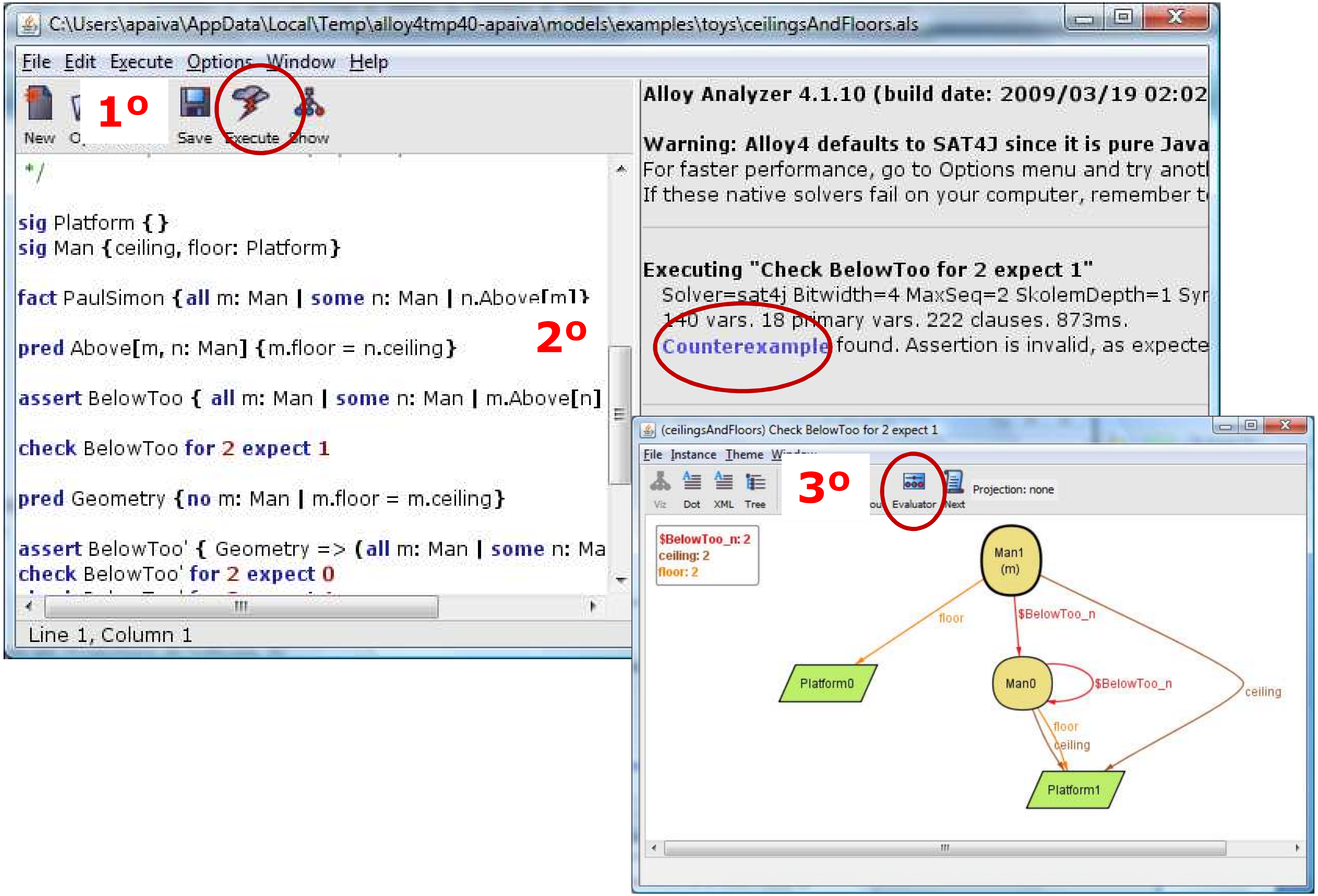


Tree



```
Instance
▶ sig Int
▼ sig Mutex
      Mutex0
      Mutex1
      Mutex2
▶ sig Process
▶ sig State
▶ sig mo/Ord
▶ sig so/Ord
▼ set $Deadlock_s
   ▼ State2
      ▼ waits
            Process -> Mutex0
```

Evaluator


C:\Users\apaiva\AppData\Local\Temp\alloy4tmp40-apaiva\models\examples\toys\ceilingsAndFloors.als
File Edit Execute Options Window Help
New
O
Save
Execute
Show
Line 1, Column 1
(ceilingsAndFloors) Check BelowToo for 2 expect 1
File Instance Theme Window
Viz
Dot
XML
Tree
Evaluator
Next
Projection: none
$BelowToo_n: 2
ceiling: 2
floor: 2
Man1
(m)
floor
$BelowToo_n
Platform0
Man0
$BelowToo_n
ceiling
floor
ceiling
Platform1

1º

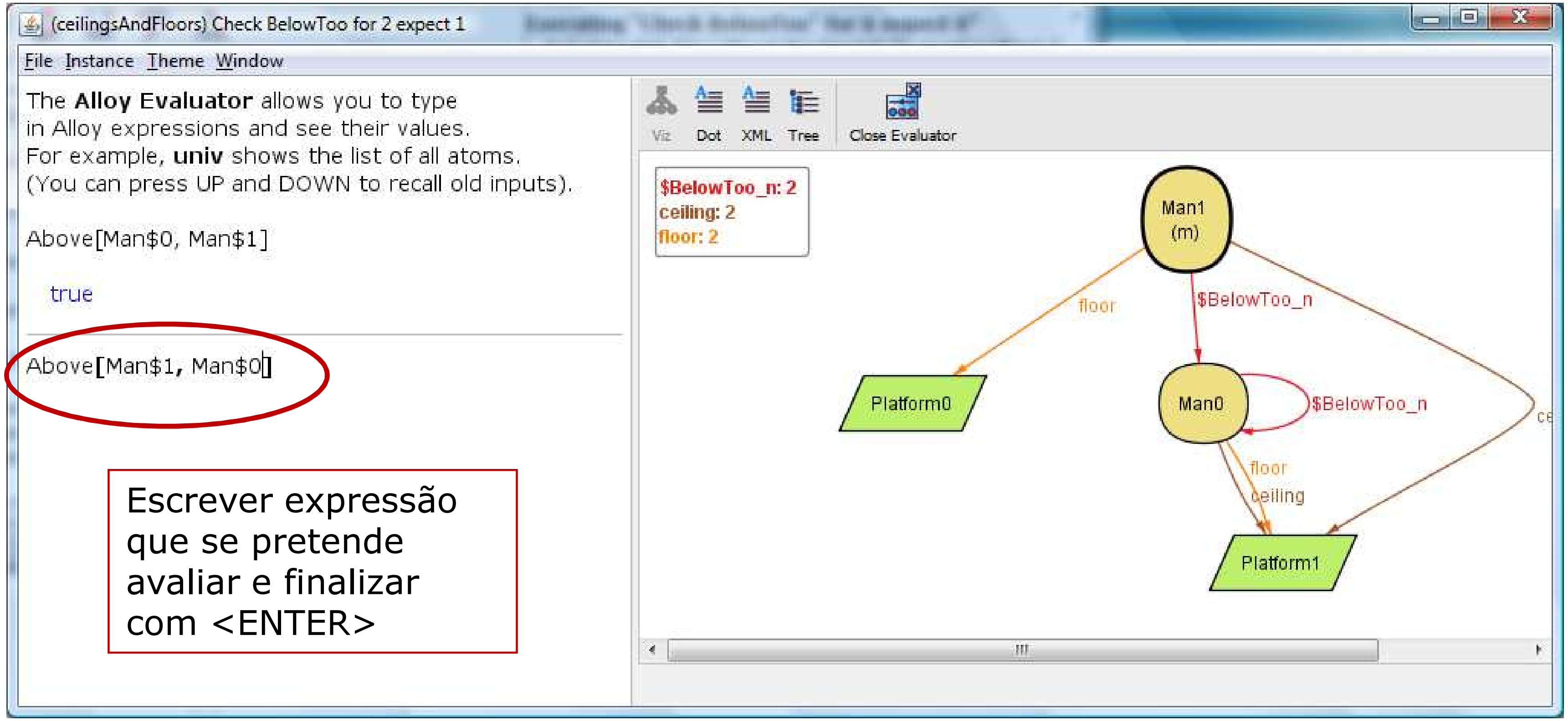
2º

3º

```
*/

sig Platform { }
sig Man {ceiling, floor: Platform}

fact PaulSimon {all m: Man | some n: Man | n.Above[m]}

pred Above[m, n: Man] {m.floor = n.ceiling}

assert BelowToo { all m: Man | some n: Man | m.Above[n]

check BelowToo for 2 expect 1

pred Geometry {no m: Man | m.floor = m.ceiling}

assert BelowToo' { Geometry => (all m: Man | some n: Ma
check BelowToo' for 2 expect 0
```

```
Alloy Analyzer 4.1.10 (build date: 2009/03/19 02:02

Warning: Alloy4 defaults to SAT4J since it is pure Java
For faster performance, go to Options menu and try anot
If these native solvers fail on your computer, remember t

Executing "Check BelowToo for 2 expect 1"
  Solver=sat4j Bitwidth=4 MaxSeq=2 SkolemDepth=1 Sy
  140 vars. 18 primary vars. 222 clauses. 873ms.
  Counterexample found. Assertion is invalid, as expecte
```

Evaluator



The **Alloy Evaluator** allows you to type in Alloy expressions and see their values.
For example, **univ** shows the list of all atoms.
(You can press UP and DOWN to recall old inputs).

```
Above[Man$0, Man$1]

  true

Above[Man$1, Man$0]
```

Escrever expressão que se pretende avaliar e finalizar com <ENTER>