University of Antwerp

Alloy - A lightweight object modelling notation

Maxim Gonnissen

https://groups.csail.mit.edu/sdg/pubs/2002/alloy-journal.pdf

Overview

- Functionality
- Language Definition
- Graphical Syntax
- Based on Z
- Tools
- Extensions
- Conclusion

> **“Alloy is a little language for describing structural properties”**

https://groups.csail.mit.edu/sdg/pubs/2002/alloy-journal.pdf

Functionality

Functionality

- **Describing of structural properties**
- **Analysis**
  - Simulation (generation)
    - Detect overconstraint
  - Checking (counterexample)
    - Detect underconstraint

https://groups.csail.mit.edu/sdg/pubs/2002/alloy-journal.pdf

Language Definition

Language Definition

- **Strongly-typed kernel**
- **Singleton-set scalars**
- **Degenerate relation set encoding**
- **Dot operator**
- **Full language**
  - Declarations
  - Shorthands
  - Paragraphs

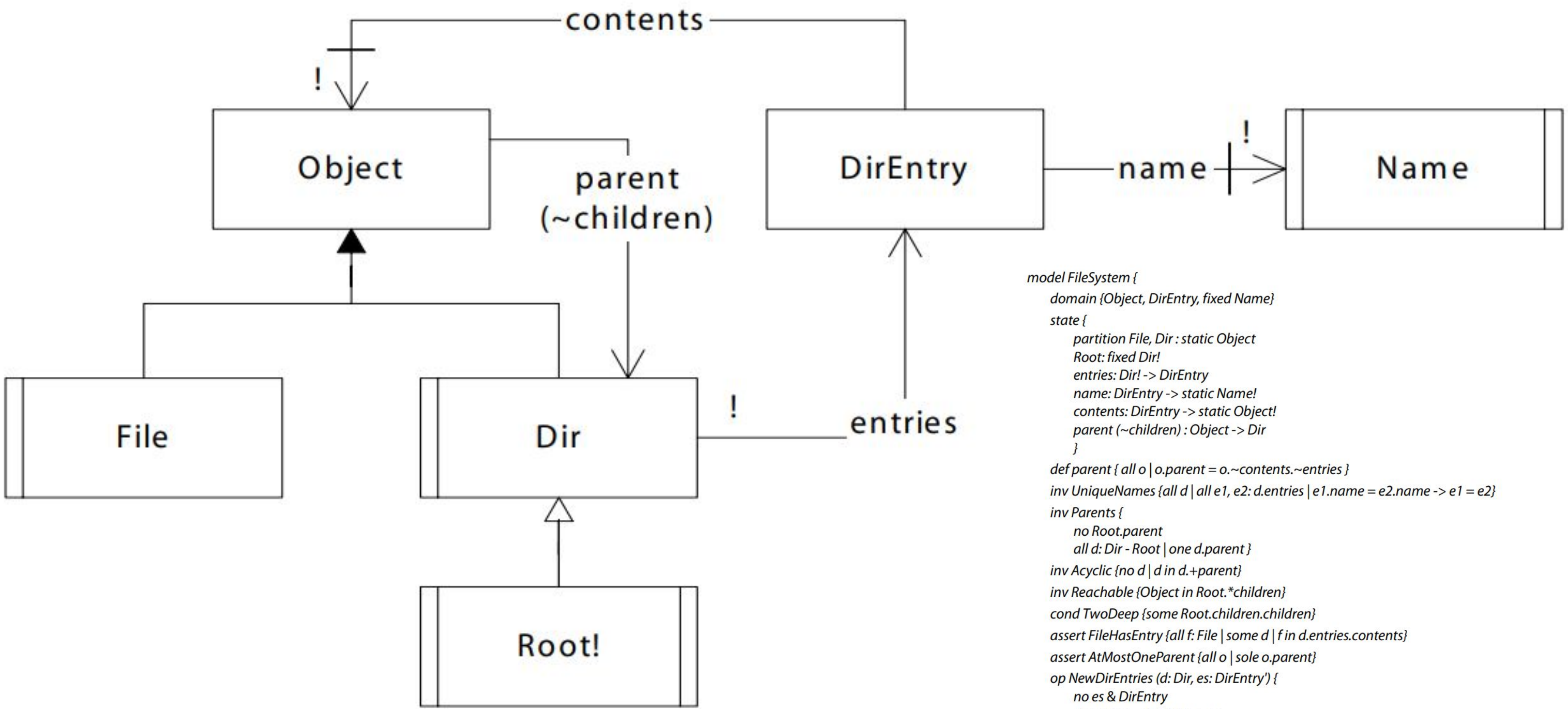
https://groups.csail.mit.edu/sdg/pubs/2002/alloy-journal.pdf

Graphical Syntax

Graphical Syntax

- State
- Constraints
- Comparison to UML

https://groups.csail.mit.edu/sdg/pubs/2002/alloy-journal.pdf



```
model FileSystem {
    domain {Object, DirEntry, fixed Name}
    state {
        partition File, Dir : static Object
        Root: fixed Dir!
        entries: Dir! -> DirEntry
        name: DirEntry -> static Name!
        contents: DirEntry -> static Object!
        parent (~children) : Object -> Dir
        }
    def parent { all o | o.parent = o.~contents.~entries }
    inv UniqueNames {all d | all e1, e2: d.entries | e1.name = e2.name -> e1 = e2}
    inv Parents {
        no Root.parent
        all d: Dir - Root | one d.parent }
    inv Acyclic {no d | d in d.+parent}
    inv Reachable {Object in Root.*children}
    cond TwoDeep {some Root.children.children}
    assert FileHasEntry {all f: File | some d | f in d.entries.contents}
    assert AtMostOneParent {all o | sole o.parent}
    op NewDirEntries (d: Dir, es: DirEntry') {
        no es & DirEntry
        d.entries' = d.entries + es
        all x: Dir - d | x.entries' = x.entries }
    op Create (d: Dir!, o: Object'!, n: Name) {
        n !in d.entries.name
        some e: DirEntry' | NewDirEntries (d, e) && e.contents' = o && e.name' = n }
    assert EntriesCreated {all d: Dir, e: DirEntry' | NewDirEntries (d,e) -> DirEntry' = DirEntry + e}
    assert CreateWorks {all d, o, n | Create (d,o,n) -> o in d.children'}
    }
```

https://groups.csail.mit.edu/sdg/pubs/2002/alloy-journal.pdf

Based on Z

Based on Z

- **Why Z?**
  - Why not OCL?
- **Automatic Analysis**
- **Set-Based Syntax**
- **Mutability**
- **Lexical Issues**

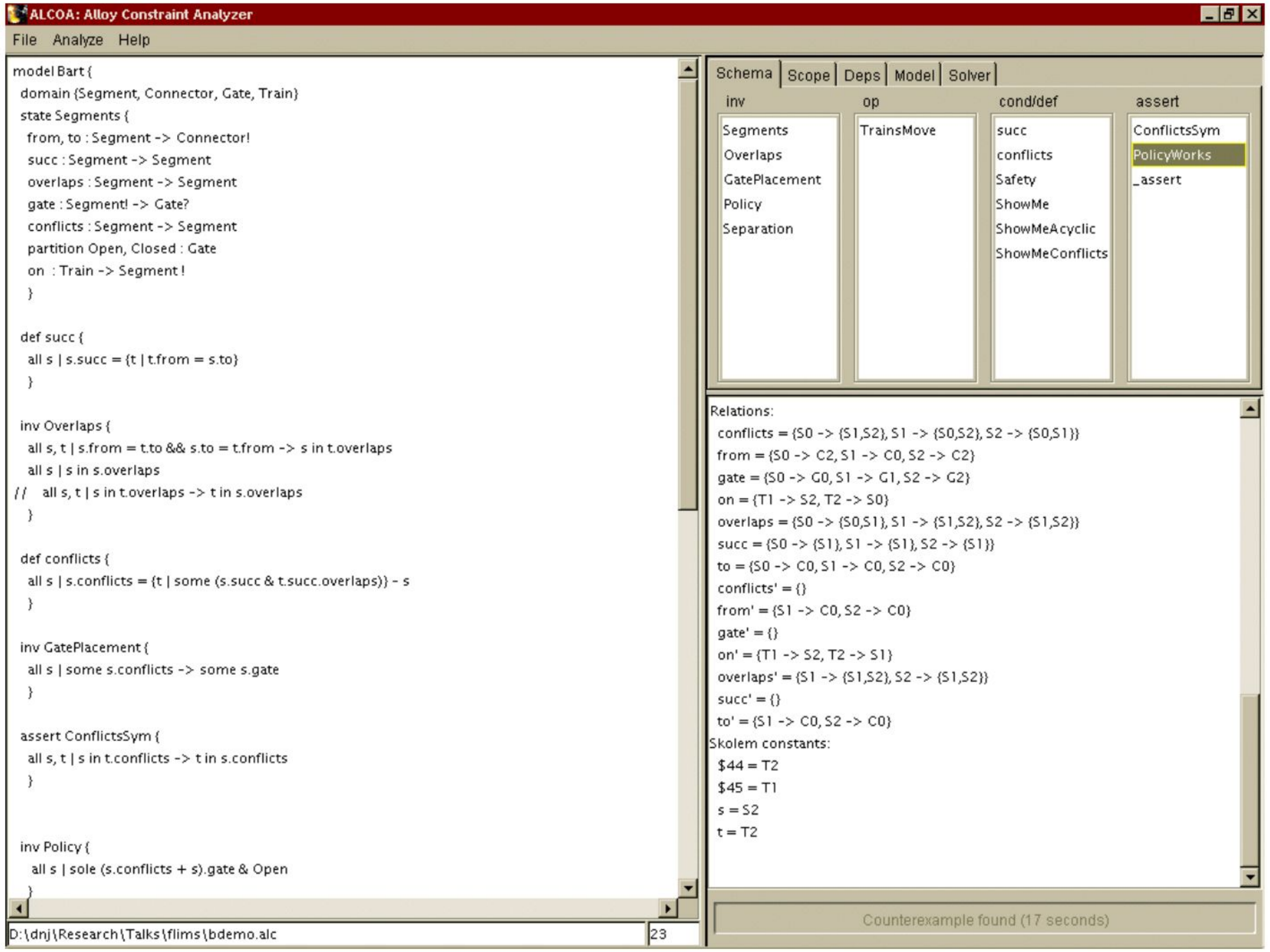
https://groups.csail.mit.edu/sdg/pubs/2002/alloy-journal.pdf

Tools

Tools

- Alcoa: The Alloy Constraint Analyzer
- Alloy Analyzer

Alcoa: The Alloy Constraint Analyzer



```
model Bart {
 domain {Segment, Connector, Gate, Train}
 state Segments {
  from, to : Segment -> Connector!
  succ : Segment -> Segment
  overlaps : Segment -> Segment
  gate : Segment! -> Gate?
  conflicts : Segment -> Segment
  partition Open, Closed : Gate
  on  : Train -> Segment !
  }

 def succ {
  all s | s.succ = {t | t.from = s.to}
  }

 inv Overlaps {
  all s, t | s.from = t.to && s.to = t.from -> s in t.overlaps
  all s | s in s.overlaps
//    all s, t | s in t.overlaps -> t in s.overlaps
  }

 def conflicts {
  all s | s.conflicts = {t | some (s.succ & t.succ.overlaps)} - s
  }

 inv GatePlacement {
  all s | some s.conflicts -> some s.gate
  }

 assert ConflictsSym {
  all s, t | s in t.conflicts -> t in s.conflicts
  }


 inv Policy {
   all s | sole (s.conflicts + s).gate & Open
  }
```

D:\dnj\Research\Talks\flims\bdemo.alc 23

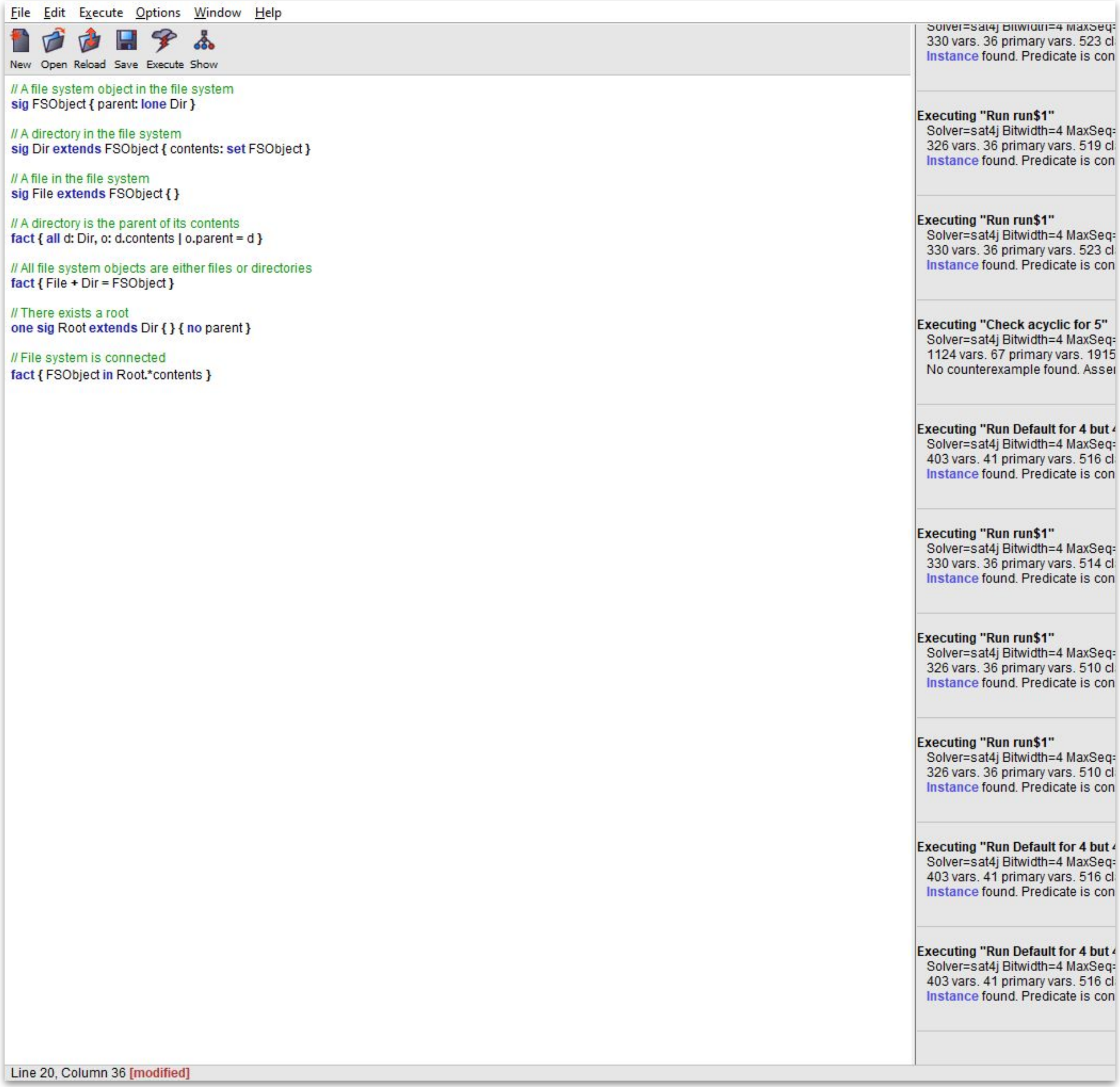
```
Relations:
 conflicts = {S0 -> {S1,S2}, S1 -> {S0,S2}, S2 -> {S0,S1}}
 from = {S0 -> C2, S1 -> C0, S2 -> C2}
 gate = {S0 -> G0, S1 -> G1, S2 -> G2}
 on = {T1 -> S2, T2 -> S0}
 overlaps = {S0 -> {S0,S1}, S1 -> {S1,S2}, S2 -> {S1,S2}}
 succ = {S0 -> {S1}, S1 -> {S1}, S2 -> {S1}}
 to = {S0 -> C0, S1 -> C0, S2 -> C0}
 conflicts' = {}
 from' = {S1 -> C0, S2 -> C0}
 gate' = {}
 on' = {T1 -> S2, T2 -> S1}
 overlaps' = {S1 -> {S1,S2}, S2 -> {S1,S2}}
 succ' = {}
 to' = {S1 -> C0, S2 -> C0}
Skolem constants:
 $44 = T2
 $45 = T1
 s = S2
 t = T2
```

Counterexample found (17 seconds)

- Nitpick
- SAT Solver
- Analysis
  - Checks
  - Generation
- Finite scope

https://groups.csail.mit.edu/sdg/pubs/TR/alcoa-overview.pdf

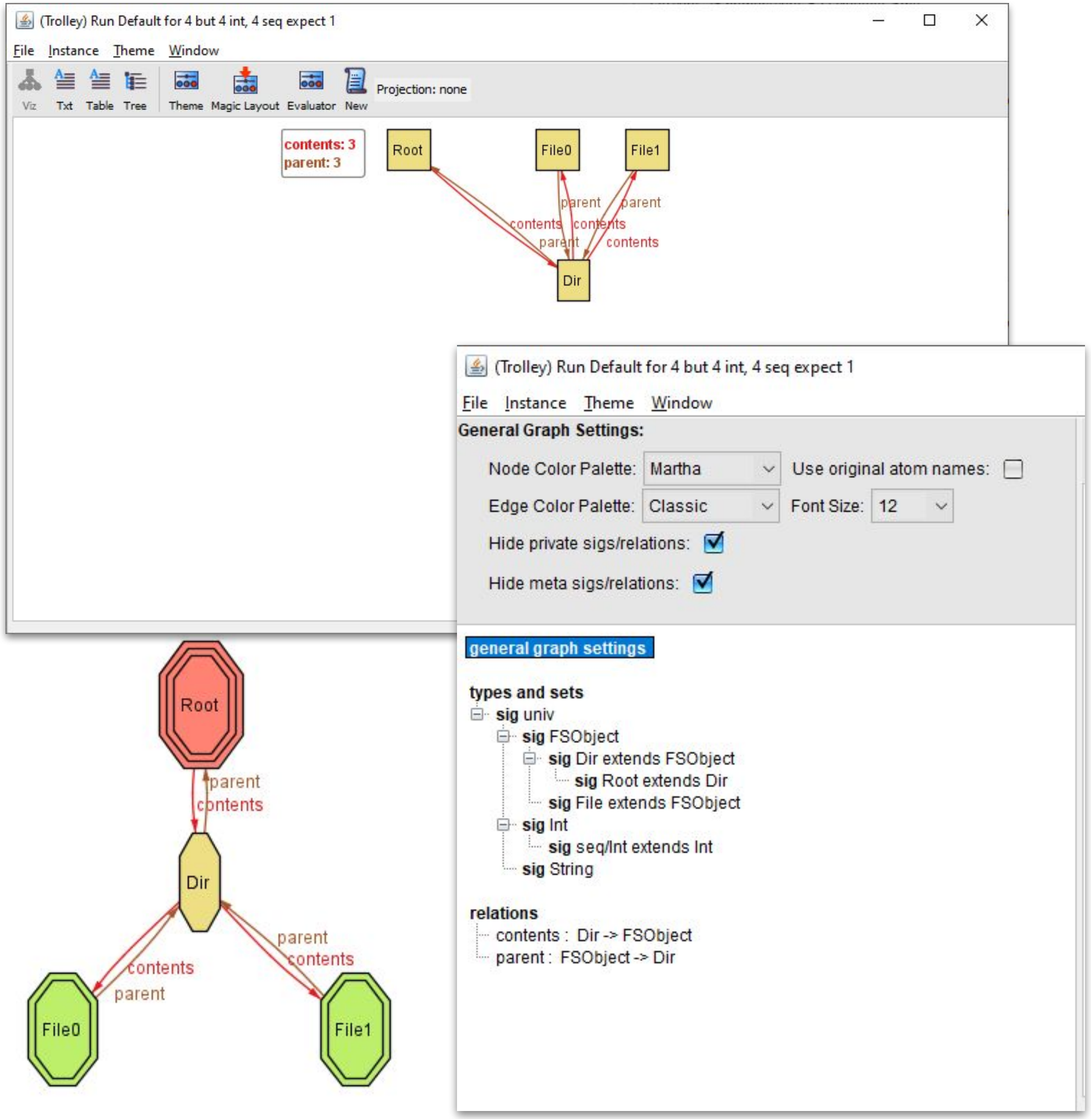
Alloy Analyzer


File Edit Execute Options Window Help
New Open Reload Save Execute Show

```
// A file system object in the file system
sig FSObject { parent: lone Dir }

// A directory in the file system
sig Dir extends FSObject { contents: set FSObject }

// A file in the file system
sig File extends FSObject { }

// A directory is the parent of its contents
fact { all d: Dir, o: d.contents | o.parent = d }

// All file system objects are either files or directories
fact { File + Dir = FSObject }

// There exists a root
one sig Root extends Dir { } { no parent }

// File system is connected
fact { FSObject in Root.*contents }
```

```
  Solver=sat4j Bitwidth=4 MaxSeq=
  330 vars. 36 primary vars. 523 cl
  Instance found. Predicate is con

Executing "Run run$1"
  Solver=sat4j Bitwidth=4 MaxSeq=
  326 vars. 36 primary vars. 519 cl
  Instance found. Predicate is con

Executing "Run run$1"
  Solver=sat4j Bitwidth=4 MaxSeq=
  330 vars. 36 primary vars. 523 cl
  Instance found. Predicate is con

Executing "Check acyclic for 5"
  Solver=sat4j Bitwidth=4 MaxSeq=
  1124 vars. 67 primary vars. 1915
  No counterexample found. Asser

Executing "Run Default for 4 but 4
  Solver=sat4j Bitwidth=4 MaxSeq=
  403 vars. 41 primary vars. 516 cl
  Instance found. Predicate is con

Executing "Run run$1"
  Solver=sat4j Bitwidth=4 MaxSeq=
  330 vars. 36 primary vars. 514 cl
  Instance found. Predicate is con

Executing "Run run$1"
  Solver=sat4j Bitwidth=4 MaxSeq=
  326 vars. 36 primary vars. 510 cl
  Instance found. Predicate is con

Executing "Run run$1"
  Solver=sat4j Bitwidth=4 MaxSeq=
  326 vars. 36 primary vars. 510 cl
  Instance found. Predicate is con

Executing "Run Default for 4 but 4
  Solver=sat4j Bitwidth=4 MaxSeq=
  403 vars. 41 primary vars. 516 cl
  Instance found. Predicate is con

Executing "Run Default for 4 but 4
  Solver=sat4j Bitwidth=4 MaxSeq=
  403 vars. 41 primary vars. 516 cl
  Instance found. Predicate is con
```

Line 20, Column 36 [modified]


(Trolley) Run Default for 4 but 4 int, 4 seq expect 1
File Instance Theme Window
Viz Txt Table Tree Theme Magic Layout Evaluator New
Projection: none
contents: 3
parent: 3
Root
File0
File1
Dir
parent
contents
Root
Dir
File0
File1
parent
contents
(Trolley) Run Default for 4 but 4 int, 4 seq expect 1
File Instance Theme Window
general graph settings

General Graph Settings:

Node Color Palette: Martha Use original atom names: □

Edge Color Palette: Classic Font Size: 12

Hide private sigs/relations: ☑

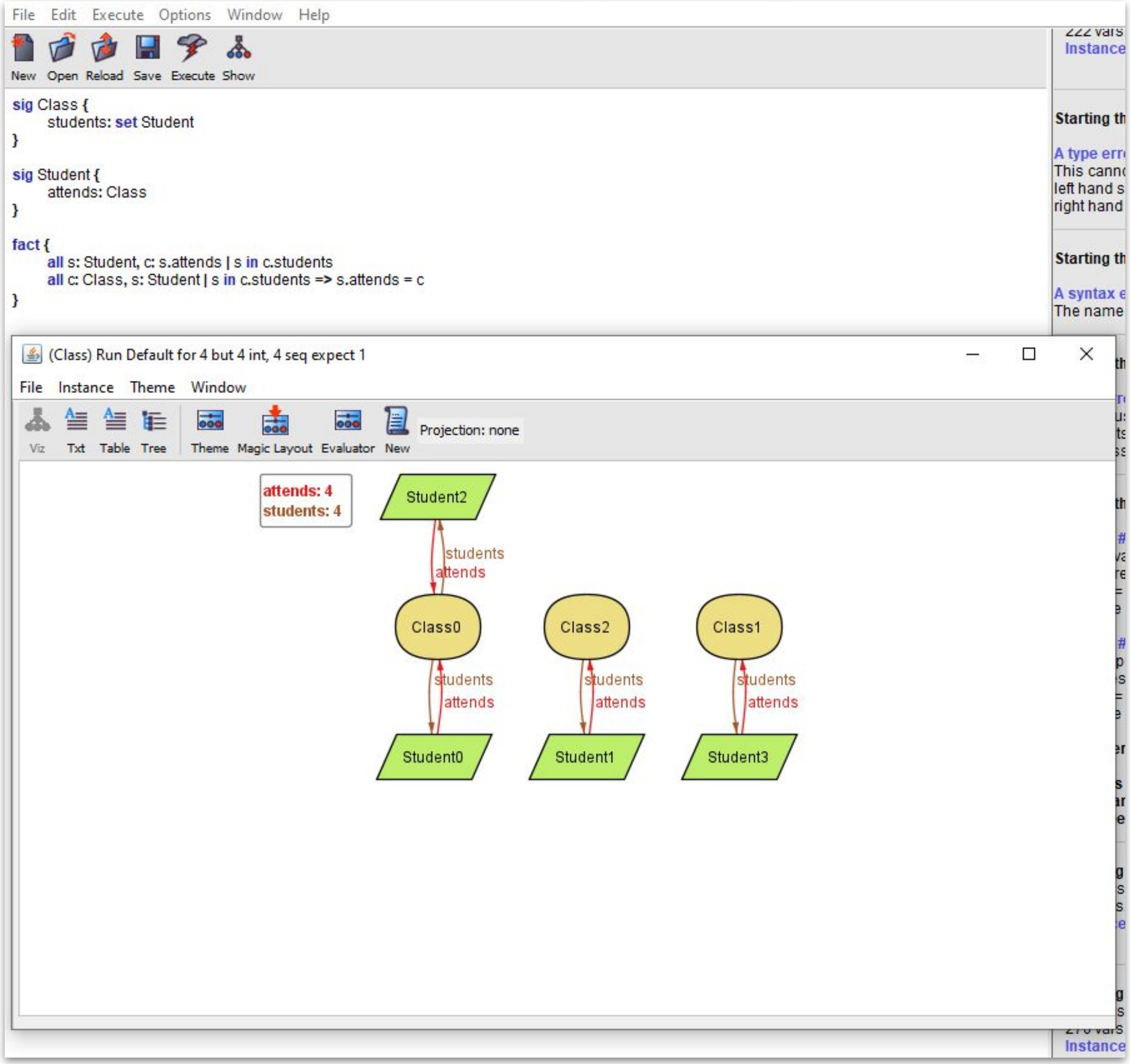
Hide meta sigs/relations: ☑

types and sets
- sig univ
  - sig FSObject
    - sig Dir extends FSObject
      - sig Root extends Dir
    - sig File extends FSObject
  - sig Int
    - sig seq/Int extends Int
  - sig String

relations
- contents : Dir -> FSObject
- parent : FSObject -> Dir

https://alloytools.org/tutorials/online/frame-FS-2.html

Alloy Analyzer



```
sig Class {
	students: set Student
}

sig Student {
	attends: Class
}

fact {
	all s: Student, c: s.attends | s in c.students
	all c: Class, s: Student | s in c.students => s.attends = c
}
```

222 vars
Instance

Starting th

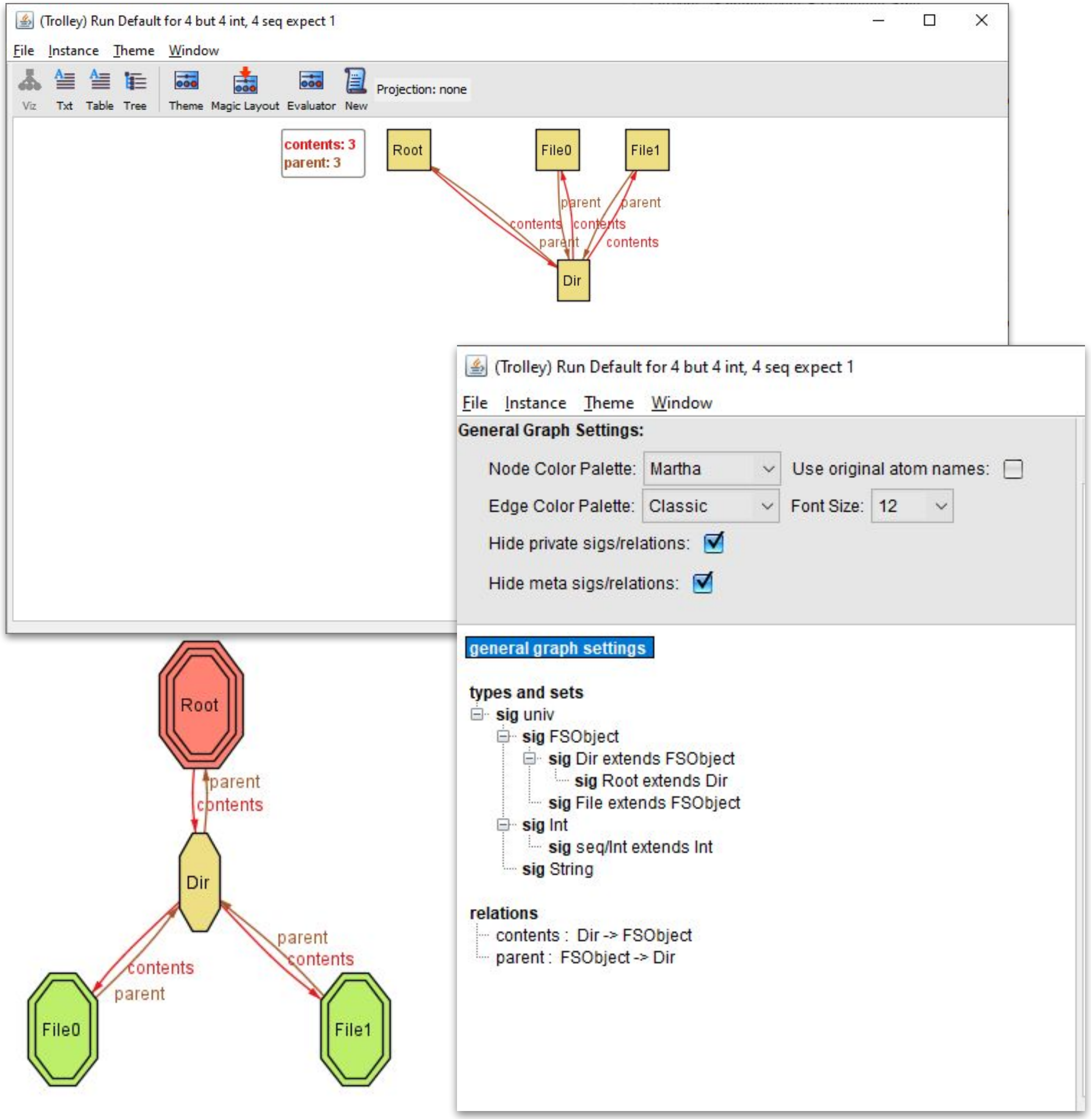
A type err
This cann
left hand s
right hand

Starting th

A syntax e
The name

Projection: none

attends: 4
students: 4



Projection: none

contents: 3
parent: 3

General Graph Settings:

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

types and sets
- sig univ
  - sig FSObject
    - sig Dir extends FSObject
      - sig Root extends Dir
    - sig File extends FSObject
  - sig Int
    - sig seq/Int extends Int
  - sig String

relations
- contents : Dir -> FSObject
- parent : FSObject -> Dir

Extensions

Extensions

- Alloy*
- Electrum
- Aluminum

https://groups.csail.mit.edu/sdg/pubs/2019/alloy-cacm-18-feb-22-2019.pdf

Conclusion