Exploring Scenario Exploration

Nuno Macedo Alcino Cunha Tiago Guimarães

HASLab
HIGH-ASSURANCE SOFTWARE LABORATORY
INESCTEC

Universidade do Minho

FASE 2015
April 15
London, UK

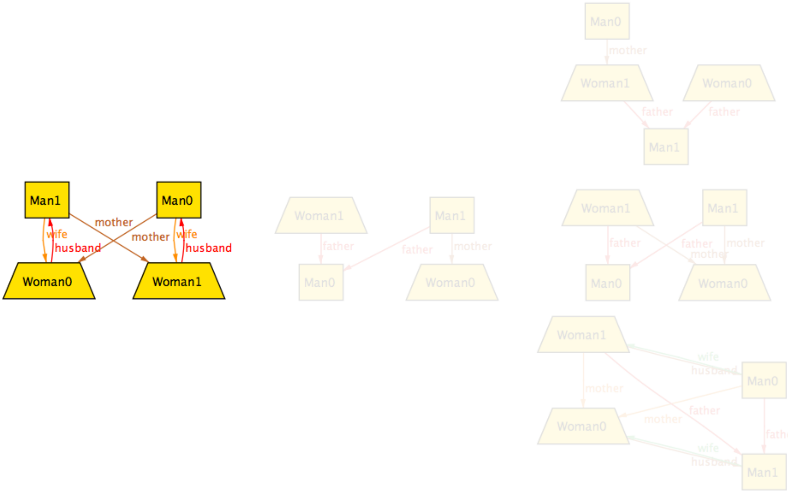
Introduction

- *Model finders* search for models that satisfy a given formula;
- Important at early MDE development stages to quickly generate *scenarios* and validate the specification;
- *Alloy*, and the underlying finder *Kodkod*, have proven suitable for this task;
- Limited usefulness: *no control* on how scenarios are generated:
  - Generate minimal / maximal scenarios;
  - Generate next closest / farthest scenarios;
  - Adapt to specification evolution;
  - ...

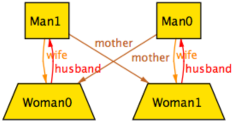
"I'm My Own Grandpa" in Alloy

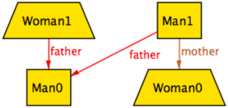
```
abstract sig Person {
  father  : lone Man,
  mother  : lone Woman
}
sig Man extends Person {
  wife    : lone Woman
}
sig Woman extends Person {
  husband : lone Man
}
fact {
  no p:Person | p in p.^(mother+father)     // Biology
  wife = ~husband                           // Terminology
  no (wife+husband) & ^(mother+father)      // SocialConvention
  Person in                                 // NoSolitary
    Person.(mother+father+~mother+~father+wife+husband)
}
run {} for exactly 2 Man, exactly 2 Woman
```

"I'm My Own Grandpa" in Alloy

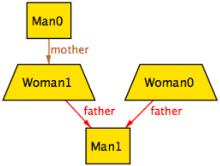


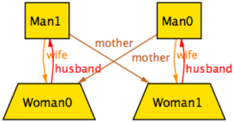
"I'm My Own Grandpa" in Alloy

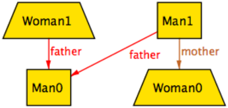




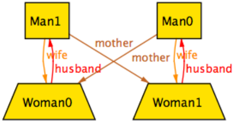
"I'm My Own Grandpa" in Alloy

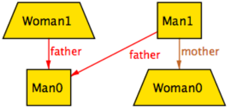


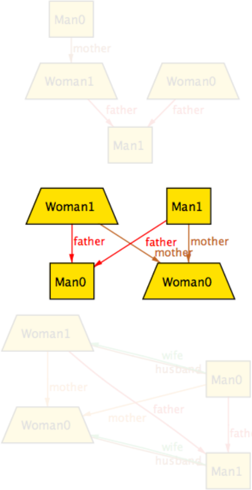




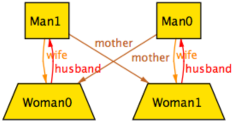
"I'm My Own Grandpa" in Alloy

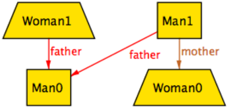


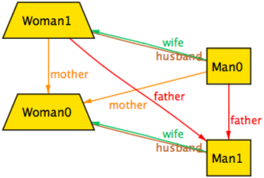




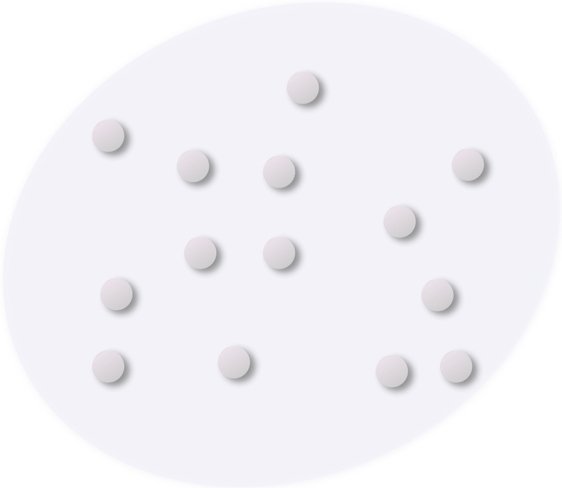
"I'm My Own Grandpa" in Alloy



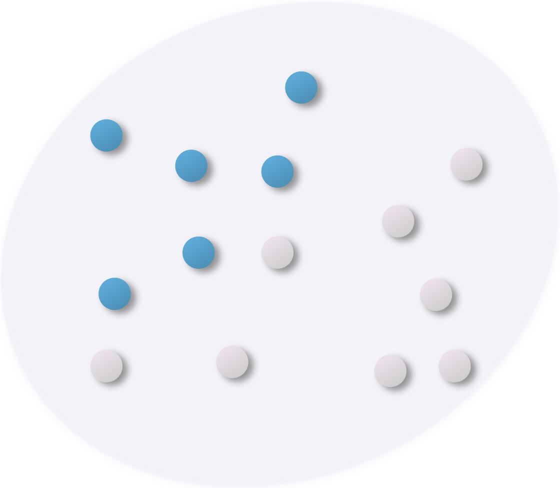




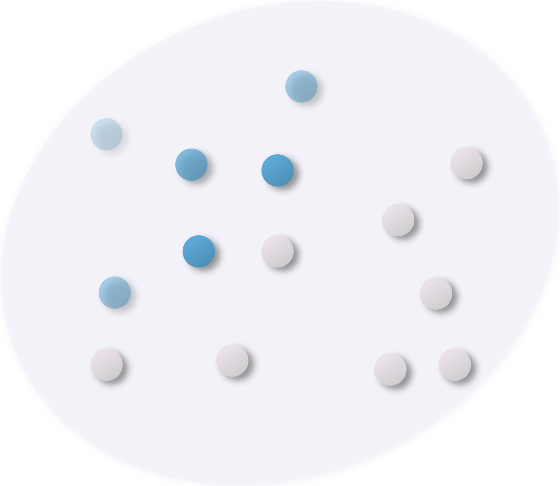
Motivation



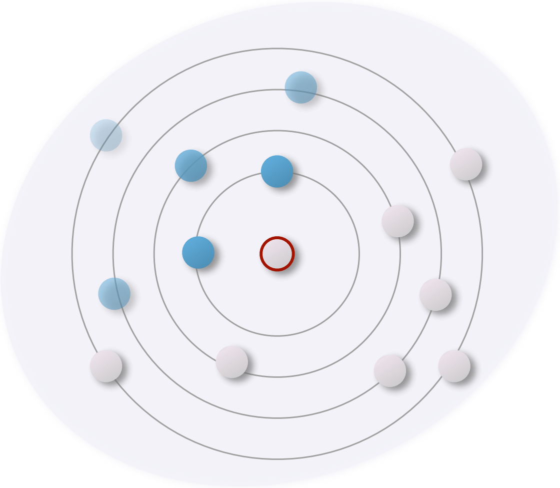
Motivation



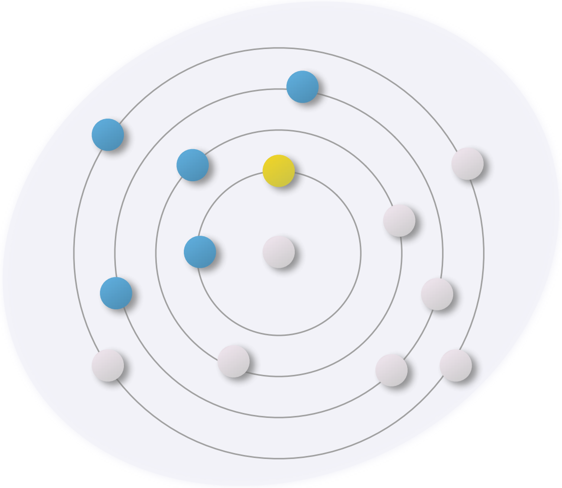
Motivation



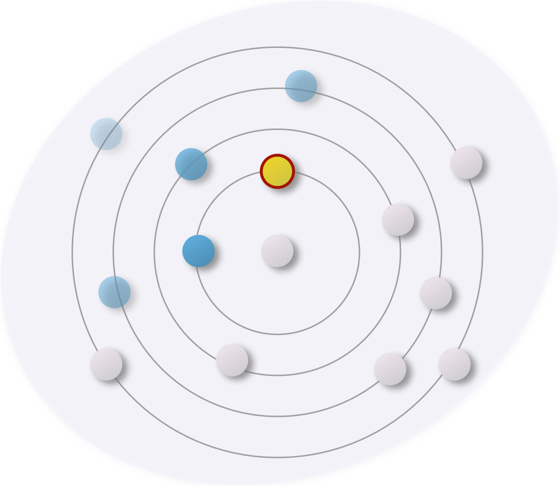
Motivation



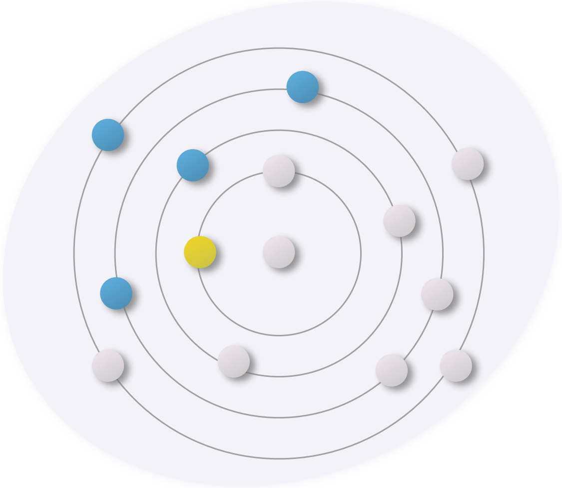
Motivation



Motivation



Motivation



Overview

- We formalize the notion of *weighted target-oriented model finding*;
- Kodkod is extended to support such model finding problems;
- We explore a set of *scenario exploration operations* over them;
- The Alloy Analyzer is extended to support these scenario exploration operations.

Kodkod

- Kodkod = relational logic + partial instances;
- Relational logic:
  - High-level specification language;
  - Favors a navigational style similar to OO;
  - Includes closures to express reachability properties;
- Partial instances:
  - Capture a priori knowledge about the desired outcome;
  - Bound the set of admissible instances (by specifying which elements may or must be present).

Model Finding

- A *model finding problem* $\langle \mathcal{A}, L, U, \phi \rangle \in \mathcal{P}$ consists of:
  - a universe of atoms $\mathcal{A}$, from which tuple sets $\mathcal{T}$ are drawn;
  - lower- and upper-bounds $L, U : \mathcal{R} \to \mathcal{T}$ that define and bound the free relational variables $\mathcal{R}$;
  - a relational formula ϕ over $\mathcal{R}$ variables.
- Model finding returns (arbitrary) bindings $B : \mathcal{R} \to \mathcal{T}$ within L and U that satisfy ϕ.

"I'm My Own Grandpa" in Kodkod

```
{M0,M1,W0,W1}

Man     : ({M0,M1},{M0,M1})
Woman   : ({W0,W1},{W0,W1})
father  : ({},{(M0,M0),(M0,M1),(M1,M0),(M1,M1),
               (W0,M0),(W0,M1),(W1,M0),(W1,M1)})
mother  : ({},{(M0,W0),(M0,W1),(M1,W0),(M1,W1),
               (W0,W0),(W0,W1),(W1,W0),(W1,W1)})
wife    : ({},{(M0,W0),(M1,W0),(M0,W1),(M1,W1)})
husband : ({},{(W0,M0),(W1,M0),(W0,M1),(W1,M1)})

all p:Man | lone p.wife && all p:Woman | lone p.husband
all p:Man+Woman | lone p.father && lone p.mother
all p:Man+Woman | !(p in p.^(mother+father))
wife = ~husband
no ((wife+husband) & ^(mother+father))
Man+Woman in (Man+Woman).(father+mother+~father+~mother+wife+husband)
```

Target-oriented Model Finding

- A *weighted target-oriented model finding problem* $\langle \mathcal{A}, L, U, T, w, \phi \rangle \in \mathcal{P}$ consists of:
  - a regular model finding problem $\langle \mathcal{A}, L, U, \phi \rangle$;
  - targets $T : \mathcal{R} \to \mathcal{T}$ for some of the $\mathcal{R}$ variables within the bounds;
  - weights $w : \mathcal{R} \to \mathbb{N}_0$ for some of the $\mathcal{R}$ variables.
- Model finding returns bindings $B : \mathcal{R} \to \mathcal{T}$ that are solutions of $\langle \mathcal{A}, L, U, \phi \rangle$ and minimize the following distance:

$$\sum_{r \in \operatorname{dom} W} w(r)|B(r) \ominus T(r)| + \sum_{r \in \operatorname{dom} T \setminus \operatorname{dom} W} |B(r) \ominus T(r)|$$

"I'm My Own Grandpa" in Target-oriented Kodkod

```
{M0,M1,W0,W1}

Man     : ({M0,M1},{M0,M1},{M0,M1},3)
Woman   : ({W0,W1},{W0,W1},{W0,W1},3)
father  : ({},{(M1,M0),(W1,M0)},{(M0,M0),(M0,M1),(M1,M0),(M1,M1),
                                 (W0,M0),(W0,M1),(W1,M0),(W1,M1)},3)
mother  : ({},{(M1,W0)},{(M0,W0),(M0,W1),(M1,W0),(M1,W1),
                         (W0,W0),(W0,W1),(W1,W0),(W1,W1)},3)
wife    : ({},{},{(M0,W0),(M1,W0),(M0,W1),(M1,W1)},1)
husband : ({},{},{(W0,M0),(W1,M0),(W0,M1),(W1,M1)},1)

all p:Man | lone p.wife && all p:Woman | lone p.husband
all p:Man+Woman | lone p.father && lone p.mother
all p:Man+Woman | !(p in p.^(mother+father))
wife = ~husband
no ((wife+husband) & ^(mother+father))
Man+Woman in (Man+Woman).(father+mother+~father+~mother+wife+husband)
```

Scenario Exploration

- State transition system with model finding problems as states;
- Generates the first problem:

$$\text{init} : \mathcal{S} \rightarrow \mathcal{P}$$

- Given the previous solution, generates the succeeding problem:

$$\text{next} : \mathcal{P} \times (\mathcal{R} \rightarrow \mathcal{T}) \rightarrow \mathcal{P}$$

Embedding Specifications

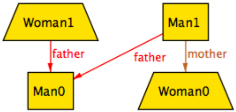
- A specification S is embedded in Kodkod as:

$$[\![\mathsf{S}]\!] = \langle \mathcal{A}_\mathsf{S}, L_\mathsf{S}, U_\mathsf{S}, \{\}, \underline{1}, \phi_\mathsf{S} \rangle$$

- Atoms are reified as relations:

$$\forall A \in \mathcal{A} | A \in \mathcal{R} \wedge L(A) = U(A) = \{A\}$$

- $[B]_=$ tests whether the value of the relations is that of B:

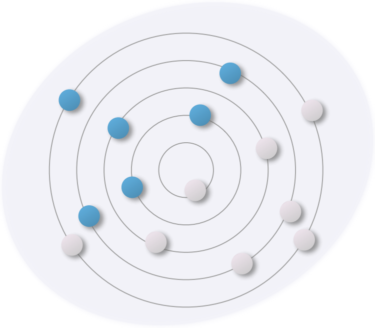


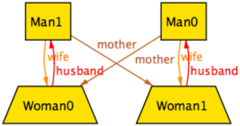

```
M0 + M1             = Man     and
W0 + W1             = Woman   and
M1->M0 + W1->M0     = father  and
M1->W0              = mother  and
none->none          = wife    and
none->none          = husband
```

Regular Generation

- Arbitrary scenarios.

$$\mathrm{init}(S) = \langle \mathcal{A}_S, L_S, U_S, \{\}, \underline{1}, \phi_S \rangle$$

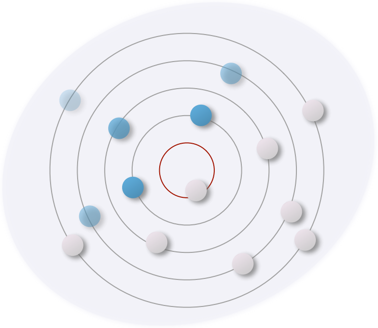




Minimal Generation

- Minimal scenarios.

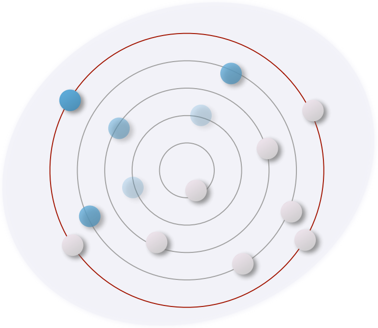
$$\mathsf{init}_{\perp}(\mathsf{S}) = \langle \mathcal{A}_{\mathsf{S}}, L_{\mathsf{S}}, U_{\mathsf{S}}, L_{\mathsf{S}}, \underline{1}, \phi_{\mathsf{S}} \rangle$$

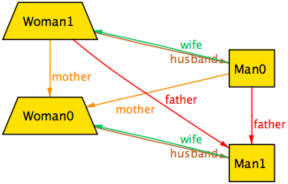


Maximal Scenarios

- Maximal scenarios (high complexity).

$$\text{init}_{\top}(\mathsf{S}) = \langle \mathcal{A}_{\mathsf{S}}, L_{\mathsf{S}}, U_{\mathsf{S}}, U_{\mathsf{S}}, \underline{1}, \phi_{\mathsf{S}} \rangle$$



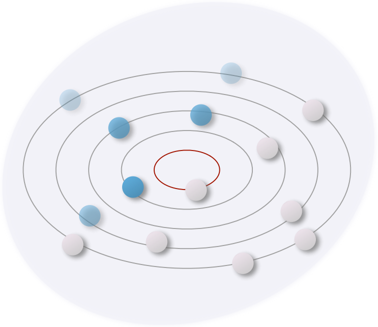


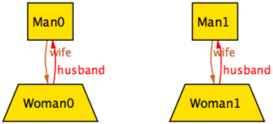
Weighted Generation

- Control the notion of 'minimal' and 'maximal' with w.

$$\mathrm{init}^{w}_{\perp}(\mathsf{S}) = \langle \mathcal{A}_{\mathsf{S}}, L_{\mathsf{S}}, U_{\mathsf{S}}, L_{\mathsf{S}}, w, \phi_{\mathsf{S}} \rangle$$

$$\mathrm{init}^{w}_{\top}(\mathsf{S}) = \langle \mathcal{A}_{\mathsf{S}}, L_{\mathsf{S}}, U_{\mathsf{S}}, U_{\mathsf{S}}, w, \phi_{\mathsf{S}} \rangle$$

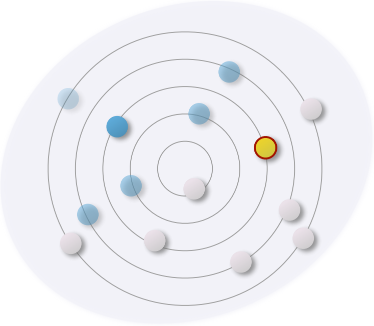


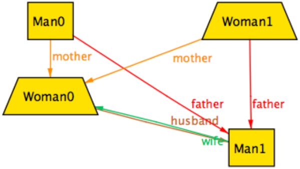


Generation from Instance

- Restart from a previously known instance B.

$$\mathrm{init}_B(\mathsf{S}) = \langle \mathcal{A}_\mathsf{S}, L_\mathsf{S}, U_\mathsf{S}, B, \underline{1}, \phi_\mathsf{S} \rangle$$

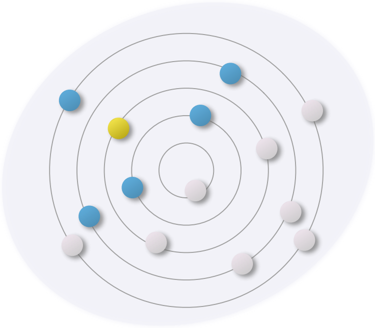


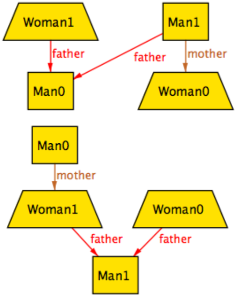


Regular Iteration

- Arbitrary solution.

$$\text{next}(\langle \mathcal{A}, L, U, \_, \_, \phi \rangle, B_0) = \langle \mathcal{A}, L, U, \{\}, \underline{1}, \phi \wedge \neg [B_0]_= \rangle$$

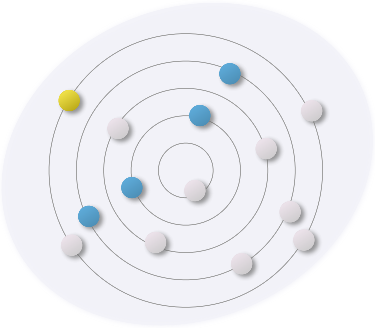


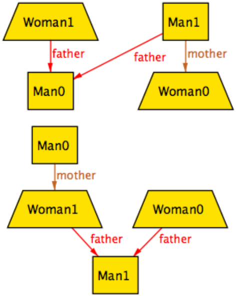


Regular Iteration

- Arbitrary solution.

$$\text{next}(\langle \mathcal{A}, L, U, \_, \_, \phi \rangle, B_0) = \langle \mathcal{A}, L, U, \{\}, \underline{1}, \phi \wedge \neg [B_0]_= \rangle$$

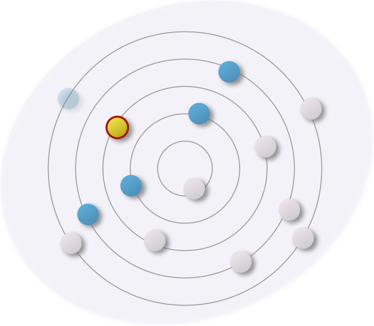


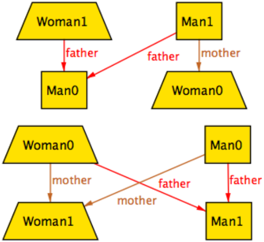


Least-change Iteration

- Generate solutions closest to the current one.

$$\text{next}_{\perp}(\langle \mathcal{A}, L, U, \_, \_, \phi \rangle, B_0) = \langle \mathcal{A}, L, U, B_0, \underline{1}, \phi \wedge \neg [B_0]_= \rangle$$

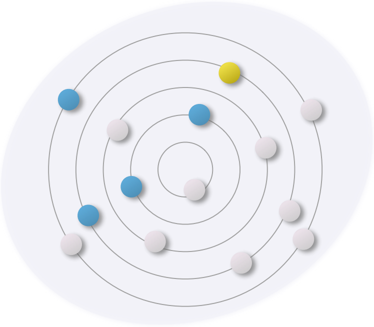


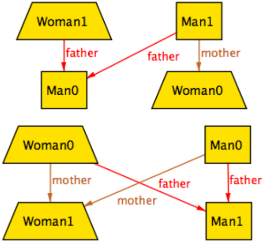


Least-change Iteration

- Generate solutions closest to the current one.

$$\text{next}_{\perp}(\langle \mathcal{A}, L, U, \_, \_, \phi \rangle, B_0) = \langle \mathcal{A}, L, U, B_0, \underline{1}, \phi \wedge \neg [B_0]_= \rangle$$

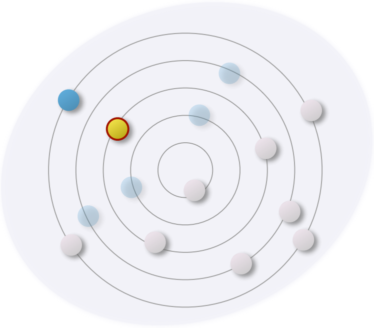


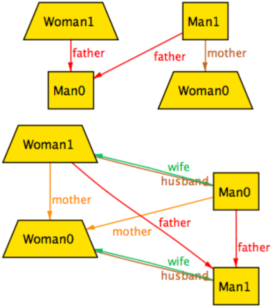


Most-change Iteration

- Generate solutions farthest from the current one.

$$\mathrm{next}_{\top}(\langle \mathcal{A}, L, U, \_, \_, \phi \rangle, B_0) = \langle \mathcal{A}, L, U, \overline{B_0}, \underline{1}, \phi \wedge \neg [B_0]_{=} \rangle$$

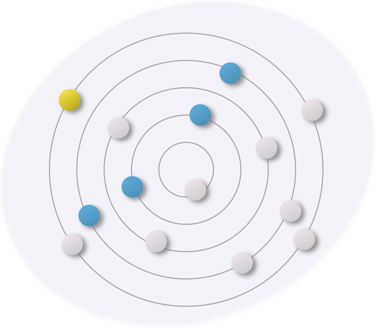


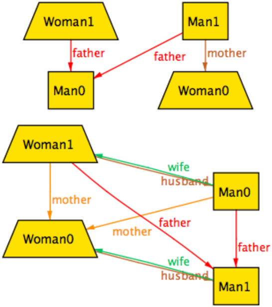


Most-change Iteration

- Generate solutions farthest from the current one.

$$\text{next}_{\top}(\langle \mathcal{A}, L, U, \_, \_, \phi \rangle, B_0) = \langle \mathcal{A}, L, U, \overline{B_0}, \underline{1}, \phi \wedge \neg [B_0]_{=} \rangle$$





Other Iteration Operations

- *Weighted iteration*: control the notion of least- and most-change with w.

$$\mathsf{next}^{w}_{\perp}(\langle \mathcal{A}, L, U, \_, \_, \phi\rangle, B_0) = \langle \mathcal{A}, L, U, B_0, w, \phi \wedge \neg[B_0]_=\rangle$$

$$\mathsf{next}^{w}_{\top}(\langle \mathcal{A}, L, U, \_, \_, \phi\rangle, B_0) = \langle \mathcal{A}, L, U, \overline{B_0}, w, \phi \wedge \neg[B_0]_=\rangle$$

- *Circular iteration*: circulate a fixed preferred solution T.

$$\mathsf{next}_T(\langle \mathcal{A}, L, U, \_, \_, \phi\rangle, B_0) = \langle \mathcal{A}, L, U, T, \underline{1}, \phi \wedge \neg[B_0]_=\rangle$$

- *Extended iteration*: introduce a new constraint ψ.

$$\mathsf{next}_\psi(\langle \mathcal{A}, L, U, \_, \_, \phi\rangle, B_0) = \langle \mathcal{A}, L, U, B_0, \underline{1}, \phi \wedge \psi\rangle$$

Alloy Analyzer Extension

- Kodkod was extended to deploy PMax-SAT solvers with weights;
- The Alloy Analyzer was extended to support Kodkod with weighted target-oriented model finding;
- Implemented support for $\mathsf{init}_\perp$, $\mathsf{init}_\top$, $\mathsf{next}^w_\perp$ and $\mathsf{next}^w_\top$ as proof of concept;
- Seamless integration to the regular Alloy user;
  - Weights are stored in the theme.

Conclusions

- Scenario exploration operations formalized over weighted target-oriented problems;
- New functionalities improve the usefulness of the Alloy Analyzer in scenario exploration;
- Subsumes previously proposed scenario exploration techniques;

- *Extension*: Implement the additional operations in the Analyzer;
- *Usability*: Infer weights from user feedback;
- *Evaluation*: Empirical study on the effectiveness of new scenario exploration operations.