

# Alloy

## Under the Hood

Alcino Cunha

November 29, 2012

# Syntax: formulas

$$\begin{aligned} \text{form} &= \text{some } \text{expr} \\ &| \text{expr in expr} \\ &| \text{not form} \\ &| \text{form and form} \\ &| \text{some } \text{var} : \text{expr} [\text{, } \text{var} : \text{expr}]^* | \text{form} \end{aligned}$$

# Syntax: expressions

$$\begin{aligned} \text{expr} = & \text{var} \\ & | \text{rel} \\ & | \text{none} \\ & | \sim \text{expr} \\ & | \hat{\text{expr}} \\ & | \text{expr} + \text{expr} \\ & | \text{expr} - \text{expr} \\ & | \text{expr} \& \text{expr} \\ & | \text{expr} . \text{expr} \\ & | \text{expr} \rightarrow \text{expr} \\ & | \{ \text{var} : \text{expr} [, \text{var} : \text{expr}]^* | \text{form} \} \end{aligned}$$

# Semantics

$$[[\cdot]] : \text{form} \times \text{binding} \rightarrow \text{bool}$$
$$[[\cdot]] : \text{expr} \times \text{binding} \rightarrow \text{relation}$$
$$\text{tuple} = \langle [atom [, atom]^*] \rangle$$
$$\text{relation} = \mathcal{P}(\text{tuple})$$
$$\text{binding} = \text{var} \cup \text{rel} \mapsto \text{relation}$$
$$\mapsto : (\text{var} \cup \text{rel}) \times \text{relation} \rightarrow \text{binding}$$
$$\oplus : \text{binding} \times \text{binding} \rightarrow \text{binding}$$

## Semantics

 $\sim : \text{tuple} \rightarrow \text{tuple}$ 

$$\sim \langle t_1, \dots, t_{|t|} \rangle = \langle t_{|t|}, \dots, t_1 \rangle$$

 $\rightarrow : \text{tuple} \times \text{tuple} \rightarrow \text{tuple}$ 

$$\langle t_1, \dots, t_{|t|} \rangle \rightarrow \langle u_1, \dots, u_{|u|} \rangle = \langle t_1, \dots, t_{|t|}, u_1, \dots, u_{|u|} \rangle$$

 $\cdot : \text{tuple} \times \text{tuple} \rightarrow \text{tuple}$ 

$$\langle t_1, \dots, t_{|t|} \rangle \cdot \langle u_1, \dots, u_{|u|} \rangle = \begin{cases} \langle t_1, \dots, t_{|t|-1}, u_2, \dots, u_{|u|} \rangle & \text{if } t_{|t|} = u_1 \\ \diamond & \text{otherwise} \end{cases}$$

# Semantics: formulas

$$\llbracket \text{some } \Phi \rrbracket_{\Gamma} = \llbracket \Phi \rrbracket_{\Gamma} \neq \emptyset$$

$$\llbracket \Phi \text{ in } \Psi \rrbracket_{\Gamma} = \llbracket \Phi \rrbracket_{\Gamma} \subseteq \llbracket \Psi \rrbracket_{\Gamma}$$

$$\llbracket \text{not } \phi \rrbracket_{\Gamma} = \neg \llbracket \phi \rrbracket_{\Gamma}$$

$$\llbracket \phi \text{ and } \psi \rrbracket_{\Gamma} = \llbracket \phi \rrbracket_{\Gamma} \wedge \llbracket \psi \rrbracket_{\Gamma}$$

$$\llbracket \text{some } x:\Phi \mid \phi \rrbracket_{\Gamma} = \bigvee \{ \llbracket \phi \rrbracket_{\Gamma \oplus x \mapsto \{t\}} \mid t \in \llbracket \Phi \rrbracket_{\Gamma} \}$$

# Semantics: expressions

$$\llbracket x \rrbracket_{\Gamma} = \Gamma(x)$$

$$\llbracket r \rrbracket_{\Gamma} = \Gamma(r)$$

$$\llbracket \text{none} \rrbracket_{\Gamma} = \emptyset$$

$$\llbracket \sim\Phi \rrbracket_{\Gamma} = \{\sim t \mid t \in \llbracket \Phi \rrbracket_{\Gamma}\}$$

$$\llbracket \hat{\sim}\Phi \rrbracket_{\Gamma} = \llbracket \Phi \rrbracket_{\Gamma} \cup \llbracket \Phi.\Phi \rrbracket_{\Gamma} \cup \llbracket \Phi.\Phi.\Phi \rrbracket_{\Gamma} \cup \dots$$

$$\llbracket \Phi + \Psi \rrbracket_{\Gamma} = \llbracket \Phi \rrbracket_{\Gamma} \cup \llbracket \Psi \rrbracket_{\Gamma}$$

$$\llbracket \Phi - \Psi \rrbracket_{\Gamma} = \llbracket \Phi \rrbracket_{\Gamma} \setminus \llbracket \Psi \rrbracket_{\Gamma}$$

$$\llbracket \Phi \& \Psi \rrbracket_{\Gamma} = \llbracket \Phi \rrbracket_{\Gamma} \cap \llbracket \Psi \rrbracket_{\Gamma}$$

$$\llbracket \Phi . \Psi \rrbracket_{\Gamma} = \{t.u \mid t \in \llbracket \Phi \rrbracket_{\Gamma} \wedge u \in \llbracket \Psi \rrbracket_{\Gamma} \wedge t.u \neq \diamond\}$$

$$\llbracket \Phi \rightarrow \Psi \rrbracket_{\Gamma} = \{t \rightarrow u \mid t \in \llbracket \Phi \rrbracket_{\Gamma} \wedge u \in \llbracket \Psi \rrbracket_{\Gamma}\}$$

$$\llbracket \{x:\Phi \mid \phi\} \rrbracket_{\Gamma} = \{t \mid t \in \llbracket \Phi \rrbracket_{\Gamma} \wedge \llbracket \phi \rrbracket_{\Gamma \oplus x \mapsto \{t\}}\}$$

# Design principles

- Detect irrelevant expressions (those that could be replaced by `none`).
- Low burden (no casts required, support for overloading).
- Syntactic robustness (subject reduction – there is an error in  $\Phi.(\Psi+\Upsilon)$  iff there is an error in  $\Phi.\Psi+\Phi.\Upsilon$ ).
- Semantic independence (types are just warnings).
- Soundness (no false alarms).
- No completeness (not all irrelevant expressions will be detected).



# Which expressions are irrelevant?

```
sig Name, Block {}
abstract sig Object { name : Name }
sig Dir extends Object { contents : set Object }
sig File extends Object { contents : set Block }
sig Link extends Object { to : Object }
one sig Root extends Dir {}
fact {
  all o : Object | some o.name
  all b : Block | some b.name
  Root.contents in Dir
  all o : Object | some o.contents
  all o : Object | some o.(File <: contents)
  all d : Dir | d not in d.^contents.to
  no (Root.to + Root.contents.to)
  no (Root + Root.contents).to
}
```

# How?

- Semantic types: “*everything is a relation*”, including types.
- The goal is to compute an upper bound for expressions using abstract interpretation.
- Type inference proceeds in two phases:
  - *Bounding types* first compute a rough approximation (bottom-up procedure).
  - *Relevance types* then refine that approximation taking context into account (top-down procedure).
- Overloading resolution comes (almost) for free: at most one relation with the same name can be relevant.

# What is a type?

- Types are unions of products of *atomic types* (disjunctive normal form).
- Atomic types are:
  - Signatures that are not supertypes (`Name`, `Block`, `File`, `Link`, `Root`).
  - Reminder types of non-abstract supertypes (`$Dir`).
- This canonical representation avoids subtype comparisons.
- Relational operators can be used to compute with types.

```
Name : {⟨Name⟩}
Object : {⟨Link⟩, ⟨File⟩, ⟨Root⟩, ⟨$Dir⟩}
ContentsFile : {⟨File, Block⟩}
to : {⟨Link, Link⟩, ⟨Link, File⟩, ⟨Link, Root⟩, ⟨Link, $Dir⟩}
```

# Bounding types

- Expression  $\Phi$  has bounding type  $A$  given binding  $\Gamma$ :

$$\Gamma \vdash \Phi : A$$

- Bounding types are upper bounds for the value of expressions:

$$\frac{\Gamma \vdash \Phi : A}{\llbracket \Phi \rrbracket_{\Lambda} \subseteq \llbracket A \rrbracket_{\Lambda}}$$

## Bounding types: expressions

$$\begin{array}{c}
 \overline{\Gamma \vdash x : \Gamma(x)} \quad \overline{\Gamma \vdash r : \Gamma(r)} \quad \overline{\Gamma \vdash \text{none} : \emptyset} \\
 \overline{\Gamma \vdash \phi : A} \quad \overline{\Gamma \vdash \phi : A} \quad \overline{\Gamma \vdash \phi : A} \\
 \overline{\Gamma \vdash \sim\phi : \sim A} \quad \overline{\Gamma \vdash \wedge\phi : \wedge A} \quad \overline{\Gamma \vdash \phi - \psi : A} \\
 \frac{\Gamma \vdash \phi : A \quad \Gamma \vdash \psi : B}{\Gamma \vdash \phi + \psi : A + B} \quad \frac{\Gamma \vdash \phi : A \quad \Gamma \vdash \psi : B}{\Gamma \vdash \phi \& \psi : A \& B} \\
 \frac{\Gamma \vdash \phi : A \quad \Gamma \vdash \psi : B}{\Gamma \vdash \phi . \psi : A . B} \quad \frac{\Gamma \vdash \phi : A \quad \Gamma \vdash \psi : B}{\Gamma \vdash \phi \rightarrow \psi : A \rightarrow B} \\
 \frac{\Gamma \vdash \phi : A \quad \Gamma \oplus x \mapsto A \vdash \phi}{\Gamma \vdash \{x : \Phi \mid \phi\} : A}
 \end{array}$$

## Bounding types: formulas

$$\frac{\Gamma \vdash \Phi : A}{\Gamma \vdash \text{some } \Phi} \quad \frac{\Gamma \vdash \Phi : A \quad \Gamma \vdash \Psi : B}{\Gamma \vdash \Phi \text{ in } \Psi}$$
$$\frac{\Gamma \vdash \phi}{\Gamma \vdash \text{not } \phi} \quad \frac{\Gamma \vdash \phi \quad \Gamma \vdash \psi}{\Gamma \vdash \phi \text{ and } \psi}$$
$$\frac{\Gamma \vdash \Phi : A \quad \Gamma \oplus x \mapsto A \vdash \phi}{\Gamma \vdash \text{some } x : \Phi \mid \phi}$$

# Bounding types: example

$$\begin{aligned} \Gamma \vdash \text{contents}_D &: \{\langle D, L \rangle, \langle D, F \rangle, \langle D, R \rangle, \langle D, D \rangle, \langle R, L \rangle, \langle R, F \rangle, \langle R, R \rangle, \langle R, D \rangle\} \\ \Gamma \vdash \text{contents}_F &: \{\langle F, B \rangle\} \\ \Gamma \vdash \text{contents} &: \{\langle D, L \rangle, \langle D, F \rangle, \langle D, R \rangle, \langle D, D \rangle, \langle R, L \rangle, \langle R, F \rangle, \langle R, R \rangle, \langle R, D \rangle, \langle F, B \rangle\} \\ \Gamma \vdash \text{Root} &: \{\langle R \rangle\} \\ \Gamma \vdash \text{Root.contents} &: \{\langle L \rangle, \langle F \rangle, \langle R \rangle, \langle D \rangle\} \\ \Gamma \vdash \text{Root} + \text{Root.contents} &: \{\langle L \rangle, \langle F \rangle, \langle R \rangle, \langle D \rangle\} \\ \Gamma \vdash \text{to} &: \{\langle L, L \rangle, \langle L, F \rangle, \langle L, R \rangle, \langle L, D \rangle\} \\ \Gamma \vdash (\text{Root} + \text{Root.contents}).\text{to} &: \{\langle L \rangle, \langle F \rangle, \langle R \rangle, \langle D \rangle\} \\ \Gamma \vdash \text{no} (\text{Root} + \text{Root.contents}).\text{to} & \end{aligned}$$

# Relevance types

- Expression  $\Phi$  has relevance type  $A$  within context  $\mathcal{C}$  given binding  $\Gamma$ :

$$\Gamma \vdash \mathcal{C} \downarrow \Phi : A$$

- The same expression in different contexts can have different relevance types.
- The relevance type is always a subset of the bounding type.

$$\frac{\Gamma \vdash \Phi : A \quad \Gamma \vdash \mathcal{C} \downarrow \Phi : B}{B \subseteq A}$$

- A context  $\mathcal{C}$  is term with an hole, denoted by  $\bullet$ .  $\mathcal{C}[\tau]$  denotes the term that results from replacing the hole in  $\mathcal{C}$  with  $\tau$ .

$$(\text{Root}.\bullet \text{ in Dir})[\text{contents}] \equiv \text{Root}.\text{contents in Dir}$$



## Relevance types: expressions

$$\frac{\Gamma \vdash \Phi : A \quad \Gamma \vdash \Psi : B \quad \Gamma \vdash C \downarrow \Phi + \Psi : C}{\Gamma \vdash C[\bullet + \Psi] \downarrow \Phi : A \& C}$$

$$\frac{\Gamma \vdash \Phi : A \quad \Gamma \vdash \Psi : B \quad \Gamma \vdash C \downarrow \Phi + \Psi : C}{\Gamma \vdash C[\Phi + \bullet] \downarrow \Psi : B \& C}$$

$$\frac{\Gamma \vdash \Phi : A \quad \Gamma \vdash \Psi : B \quad \Gamma \vdash C \downarrow \Phi - \Psi : C}{\Gamma \vdash C[\bullet - \Psi] \downarrow \Phi : C}$$

$$\frac{\Gamma \vdash \Phi : A \quad \Gamma \vdash \Psi : B \quad \Gamma \vdash C \downarrow \Phi - \Psi : C}{\Gamma \vdash C[\Phi - \bullet] \downarrow \Psi : B \& C}$$

$$\frac{\Gamma \vdash \Phi : A \quad \Gamma \vdash \Psi : B \quad \Gamma \vdash C \downarrow \Phi \& \Psi : C}{\Gamma \vdash C[\bullet \& \Psi] \downarrow \Phi : A \& C}$$

$$\frac{\Gamma \vdash \Phi : A \quad \Gamma \vdash \Psi : B \quad \Gamma \vdash C \downarrow \Phi \& \Psi : C}{\Gamma \vdash C[\Phi \& \bullet] \downarrow \Psi : B \& C}$$

## Relevance types: expressions

$$\frac{\Gamma \vdash \Phi : A \quad \Gamma \vdash \Psi : B \quad \Gamma \vdash C \downarrow \Phi . \Psi : C}{\Gamma \vdash C[\bullet . \Psi] \downarrow \Phi : \{t \mid t \in A \wedge u \in B \wedge t.u \in C\}}$$

$$\frac{\Gamma \vdash \Phi : A \quad \Gamma \vdash \Psi : B \quad \Gamma \vdash C \downarrow \Phi . \Psi : C}{\Gamma \vdash C[\Phi . \bullet] \downarrow \Psi : \{u \mid t \in A \wedge u \in B \wedge t.u \in C\}}$$

$$\frac{\Gamma \vdash \Phi : A \quad \Gamma \vdash \Psi : B \quad \Gamma \vdash C \downarrow \Phi \rightarrow \Psi : C}{\Gamma \vdash C[\bullet \rightarrow \Psi] \downarrow \Phi : \{t \mid t \in A \wedge u \in B \wedge t \rightarrow u \in C\}}$$

$$\frac{\Gamma \vdash \Phi : A \quad \Gamma \vdash \Psi : B \quad \Gamma \vdash C \downarrow \Phi \rightarrow \Psi : C}{\Gamma \vdash C[\Phi \rightarrow \bullet] \downarrow \Psi : \{u \mid t \in A \wedge u \in B \wedge t \rightarrow u \in C\}}$$

## Relevance types: expressions

$$\frac{\Gamma \vdash \Phi : A \quad \Gamma \vdash C \downarrow \sim\Phi : B}{\Gamma \vdash C[\sim\bullet] \downarrow \Phi : \sim B}$$

$$\frac{\Gamma \vdash \Phi : A \quad \Gamma \vdash C \downarrow \sim\Phi : B}{\Gamma \vdash C[\sim\bullet] \downarrow \Phi : \{t \mid t \in A \wedge u \in B \wedge \langle u_1, t_1 \rangle \in *A \wedge \langle t_2, u_2 \rangle \in *A\}}$$

$$\frac{\Gamma \vdash \Phi : A \quad \Gamma \vdash C \downarrow \{x:\Phi \mid \phi\} : B}{\Gamma \vdash C[\{x:\bullet \mid \phi\}] \downarrow \Phi : B}$$

$$\frac{\Gamma \vdash \Phi : A \quad \Gamma \vdash C \downarrow \{x:\Phi \mid \phi\} : B}{\Gamma \oplus x \mapsto A \vdash C[\{x:\Phi \mid \bullet\}] \downarrow \phi}$$

## Relevance types: formulas

$$\frac{\Gamma \vdash \Phi : A \quad \Gamma \vdash \mathcal{C} \downarrow \text{some } \Phi}{\Gamma \vdash \mathcal{C}[\text{some } \bullet] \downarrow \Phi : A}$$
$$\frac{\Gamma \vdash \Phi : A \quad \Gamma \vdash \Psi : B \quad \Gamma \vdash \mathcal{C} \downarrow \Phi \text{ in } \Psi}{\Gamma \vdash \mathcal{C}[\bullet \text{ in } \Psi] \downarrow \Phi : A}$$
$$\frac{\Gamma \vdash \Phi : A \quad \Gamma \vdash \Psi : B \quad \Gamma \vdash \mathcal{C} \downarrow \Phi \text{ in } \Psi}{\Gamma \vdash \mathcal{C}[\Phi \text{ in } \bullet] \downarrow \Psi : A \& B}$$

## Relevance types: formulas

$$\frac{\Gamma \vdash \mathcal{C} \downarrow \text{not } \phi}{\Gamma \vdash \mathcal{C}[\text{not } \bullet] \downarrow \phi}$$

$$\frac{\Gamma \vdash \mathcal{C} \downarrow \phi \text{ and } \psi}{\Gamma \vdash \mathcal{C}[\bullet \text{ and } \psi] \downarrow \phi} \quad \frac{\Gamma \vdash \mathcal{C} \downarrow \phi \text{ and } \psi}{\Gamma \vdash \mathcal{C}[\phi \text{ and } \bullet] \downarrow \psi}$$

$$\frac{\Gamma \vdash \Phi : A \quad \Gamma \vdash \mathcal{C} \downarrow \text{some } x : \Phi \mid \phi}{\Gamma \vdash \mathcal{C}[\text{some } x : \bullet \mid \phi] \downarrow \Phi : A}$$

$$\frac{\Gamma \vdash \Phi : A \quad \Gamma \vdash \mathcal{C} \downarrow \text{some } x : \Phi \mid \phi}{\Gamma \oplus x \mapsto A \vdash \mathcal{C}[\text{some } x : \Phi \mid \bullet] \downarrow \phi}$$

$$\frac{\Gamma \vdash \phi}{\Gamma \vdash \bullet \downarrow \phi}$$

# Bounding types: example

$$\Gamma \vdash \text{no } (\text{Root} + \text{Root}.\text{contents}).\text{to}$$

$$\Gamma \vdash \bullet \downarrow \text{no } (\text{Root} + \text{Root}.\text{contents}).\text{to}$$

$$\Gamma \vdash \text{no } \bullet \downarrow (\text{Root} + \text{Root}.\text{contents}).\text{to} : \{\langle L \rangle, \langle F \rangle, \langle R \rangle, \langle D \rangle\}$$

$$\Gamma \vdash \text{no } (\text{Root} + \text{Root}.\text{contents}).\bullet \downarrow \text{to} : \{\langle L, L \rangle, \langle L, F \rangle, \langle L, R \rangle, \langle L, D \rangle\}$$

$$\Gamma \vdash \text{no } \bullet.\text{to} \downarrow \text{Root} + \text{Root}.\text{contents} : \{\langle L \rangle\}$$

$$\Gamma \vdash \text{no } (\bullet + \text{Root}.\text{contents}).\text{to} \downarrow \text{Root} : \emptyset$$

$$\Gamma \vdash \text{no } (\text{Root} + \bullet).\text{to} \downarrow \text{Root}.\text{contents} : \{\langle L \rangle\}$$

$$\Gamma \vdash \text{no } (\text{Root} + \bullet.\text{contents}).\text{to} \downarrow \text{Root} : \{\langle R \rangle\}$$

$$\Gamma \vdash \text{no } (\text{Root} + \text{Root}.\bullet).\text{to} \downarrow \text{contents} : \{\langle R, L \rangle\}$$

$$\Gamma \vdash \text{no } (\text{Root} + \text{Root}.\text{to}(\bullet + \text{contents}_F)).\text{to} \downarrow \text{contents}_D : \{\langle R, L \rangle\}$$

$$\Gamma \vdash \text{no } (\text{Root} + \text{Root}.\text{to}(\text{contents}_D + \bullet)).\text{to} \downarrow \text{contents}_F : \emptyset$$

# The real implementation

- Computations are performed on base instead of atomic types (better error messages).
- Empty bounding types are immediately reported as errors.
- Expressions of mixed arity are rejected.
- `none` has arity 1 and special atomic type `none`.
- The current implementation of relevance types is bugged.

# Architecture





# Kodkod

- Kodkod is a relational model finder.
- A *Kodkod problem* consists of a *universe declaration*, *relation declarations*, and a *formula*.
- A relation declaration specifies an arity and (lower and upper) bounds.

# From Alloy to Kodkod

- Only declares relations for atomic types and fields.
- From scope infer universe and bounds for relations.
- The semantics of the structural part is encoded as a formula.
- Asserts are negated (a formula is valid iff its negation is unsat).
- The value of a signature is computed from its representation as atomic types.
- Atoms are renamed by the visualizer.

# From Alloy to Kodkod

```

abstract sig Person {}
sig Man extends Person {wife : lone Woman}
sig Woman extends Person {}
one sig Eve extends Woman {}
run {wife.Eve in Person} for 3

```

$$U = \{A_1, A_2, A_3\}$$

$$\emptyset \subseteq \text{Man}_1 \subseteq \{\langle A_1 \rangle, \langle A_2 \rangle\}$$

$$\emptyset \subseteq \$\text{Woman}_1 \subseteq \{\langle A_1 \rangle, \langle A_2 \rangle\}$$

$$\{\langle A_3 \rangle\} \subseteq \text{Eve}_1 \subseteq \{\langle A_3 \rangle\}$$

$$\emptyset \subseteq \text{wife}_2 \subseteq \{\langle A_1, A_1 \rangle, \langle A_1, A_2 \rangle, \langle A_1, A_3 \rangle, \langle A_2, A_1 \rangle, \langle A_2, A_2 \rangle, \langle A_2, A_3 \rangle\}$$

```

no Man & ($Woman+Eve)
all x:Man | lone x.wife and x.wife in $Woman+Eve
wife.univ in Man
wife.Eve in Man+$Woman+Eve

```

# From Kodkod to SAT

- A relation  $r$  of arity  $k$  can be represented by  $k$ -dimensional matrix  $|\mathcal{U}|^k$  of propositional variables.

$$r[i_1, \dots, i_k] = \begin{cases} \top & \text{if } \langle A_{i_1}, \dots, A_{i_k} \rangle \in r_L \\ r_{i_1, \dots, i_k} & \text{if } \langle A_{i_1}, \dots, A_{i_k} \rangle \in r_U \setminus r_L \\ \perp & \text{otherwise} \end{cases}$$

- Relational operators can then be lifted to matrix operations. For example:
  - Composition is product;
  - Union is sum;
  - Intersection is Hadamard product.
- Formulas yield propositional formulas. In particular inclusion is implication.

## From Kodkod to SAT

wife.Eve in Man+\$Woman+Eve

$$\begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} \\ w_{2,1} & w_{2,2} & w_{2,3} \\ \perp & \perp & \perp \end{bmatrix} \cdot \begin{bmatrix} \perp \\ \perp \\ \top \end{bmatrix} \text{ in } \begin{bmatrix} M_1 \\ M_2 \\ \perp \end{bmatrix} + \begin{bmatrix} W_1 \\ W_2 \\ \perp \end{bmatrix} + \begin{bmatrix} \perp \\ \perp \\ \top \end{bmatrix}$$

$$\begin{bmatrix} w_{1,1} \wedge \perp \vee w_{1,2} \wedge \perp \vee w_{1,3} \wedge \top \\ w_{2,1} \wedge \perp \vee w_{2,2} \wedge \perp \vee w_{2,3} \wedge \top \\ \perp \wedge \perp \vee \perp \wedge \perp \vee \perp \wedge \top \end{bmatrix} \text{ in } \begin{bmatrix} M_1 \vee W_1 \vee \perp \\ M_2 \vee W_2 \vee \perp \\ \perp \vee \perp \vee \top \end{bmatrix}$$

$$(w_{1,3} \Rightarrow M_1 \vee W_1) \wedge (w_{2,3} \Rightarrow M_2 \vee W_2)$$

# Symmetry breaking

- Kodkod performs several optimizations to decrease SAT complexity.
- The most significant is symmetry breaking.
- Since atoms are uninterpreted (almost) any permutation of an instance is also a valid instance.
- A symmetry-breaking formula is conjoined to the problem formula.
- It tries to avoid most symmetries but for efficiency reasons the technique is not complete.

# Skolemization

- Since scope is finite, quantifiers can be handled by expansion.

$$\begin{aligned} & \text{some adam : Man} \mid \text{adam.wife} = \text{Eve} \\ & \quad \equiv \\ & \{\langle A_1 \rangle\}.\text{wife} = \text{Eve} \text{ or } \{\langle A_2 \rangle\}.\text{wife} = \text{Eve} \end{aligned}$$

- Unfortunately, this technique yields no witnesses to existential quantifiers.

# Skolemization

- Skolemization replaces existentially quantified variables by new free variables.
- Free variables are implicitly existentially quantified.
- Generates smaller (equisatisfiable) formulas.

$$\begin{aligned} \text{some adam : Man} \mid \text{adam.wife} = \text{Eve} \\ \cong \\ \emptyset \subseteq \$\text{adam}_1 \subseteq \{\langle A_1 \rangle, \langle A_2 \rangle\} \\ \text{one } \$\text{adam} \text{ and } \$\text{adam.wife} = \text{Eve} \end{aligned}$$

- The skolemized variables are witnesses of the quantifier.
- Skolemization handles higher-order existential quantifications.



# Skolemization

all m : Man | some w : Woman | m.wife = w

$\cong$

$\emptyset \subseteq w_2 \subseteq \{\langle A_1, A_1 \rangle, \langle A_1, A_2 \rangle, \langle A_1, A_3 \rangle, \langle A_2, A_1 \rangle, \langle A_2, A_2 \rangle, \langle A_2, A_3 \rangle\}$

all m : Man | one m.\$w and m.wife = m.\$w

some husband : Woman  $\rightarrow$  Man | husband = ~wife

$\cong$

$\emptyset \subseteq \$\text{husband}_2 \subseteq \{\langle A_1, A_1 \rangle, \langle A_1, A_2 \rangle, \langle A_2, A_1 \rangle, \langle A_2, A_2 \rangle, \langle A_3, A_1 \rangle, \langle A_3, A_2 \rangle\}$

$\$\text{husband} = \text{wife}$

# Bibliography

- Daniel Jackson: *Software Abstractions: Logic, Language, and Analysis*. Revised edition, MIT Press 2012.
- Jonathan Edwards, Daniel Jackson, Emina Torlak: A type system for object models. SIGSOFT FSE 2004: 189-199.
- Emina Torlak, Daniel Jackson: *Kodkod: A Relational Model Finder*. TACAS 2007: 632-647.
- Aleksandar Milicevic, Daniel Jackson: Preventing Arithmetic Overflows in Alloy. ABZ 2012: 108-121.