Alloy
Under the Hood

Alcino Cunha

December 14, 2017

Syntax: formulas

$$
\begin{array}{rcl}
\mathit{form} & = & \texttt{some}\ \mathit{expr} \\
& | & \mathit{expr}\ \texttt{in}\ \mathit{expr} \\
& | & \texttt{not}\ \mathit{form} \\
& | & \mathit{form}\ \texttt{and}\ \mathit{form} \\
& | & \texttt{some}\ \mathit{var}\ \texttt{:}\ \mathit{expr}\ [\texttt{,}\ \mathit{var}\ \texttt{:}\ \mathit{expr}]^{*} \mid \mathit{form} \\
& | & \texttt{eq[}\mathit{expr}\texttt{,}\mathit{expr}\texttt{]} \\
& | & \texttt{gte[}\mathit{expr}\texttt{,}\mathit{expr}\texttt{]}
\end{array}
$$

Syntax: expressions

$$
\begin{array}{rcl}
\mathit{expr} & = & \mathit{var} \\
& | & \mathit{rel} \\
& | & \texttt{none} \\
& | & \texttt{\~{}}\mathit{expr} \\
& | & \texttt{\^{}}\mathit{expr} \\
& | & \mathit{expr}\ \texttt{+}\ \mathit{expr} \\
& | & \mathit{expr}\ \texttt{-}\ \mathit{expr} \\
& | & \mathit{expr}\ \texttt{\&}\ \mathit{expr} \\
& | & \mathit{expr}\ \texttt{.}\ \mathit{expr} \\
& | & \mathit{expr}\ \texttt{->}\ \mathit{expr} \\
& | & \{\ \mathit{var}\ \texttt{:}\ \mathit{expr}\ [\texttt{,}\ \mathit{var}\ \texttt{:}\ \mathit{expr}]^*\ \texttt{|}\ \mathit{form}\ \} \\
& | & \mathit{int} \\
& | & \texttt{int}[\mathit{expr}] \\
& | & \texttt{plus}[\mathit{expr}\texttt{,}\mathit{expr}]
\end{array}
$$

Semantics

$$
\begin{aligned}
[\![\cdot]\!]. &: \mathit{form} \times \mathit{binding} \to \mathit{bool} \\
[\![\cdot]\!]. &: \mathit{expr} \times \mathit{binding} \to \mathit{relation}
\end{aligned}
$$

$$
\begin{aligned}
\mathit{atom} &= \mathbb{Z} \mid \mathit{id} \\
\mathit{tuple} &= \langle [\mathit{atom}\ [, \mathit{atom}]^*] \rangle \\
\mathit{relation} &= \mathcal{P}(\mathit{tuple}) \\
\mathit{binding} &= \mathit{var} \cup \mathit{rel} \mapsto \mathit{relation}
\end{aligned}
$$

$$
\begin{aligned}
\mapsto &: (\mathit{var} \cup \mathit{rel}) \times \mathit{relation} \to \mathit{binding} \\
\oplus &: \mathit{binding} \times \mathit{binding} \to \mathit{binding}
\end{aligned}
$$

Semantics

$$\begin{aligned} \sim &: \mathit{tuple} \to \mathit{tuple} \\ \sim\langle t_1, \ldots, t_{|t|} \rangle &= \langle t_{|t|}, \ldots, t_1 \rangle \end{aligned}$$

$$\begin{aligned} \texttt{->} &: \mathit{tuple} \times \mathit{tuple} \to \mathit{tuple} \\ \langle t_1, \ldots, t_{|t|} \rangle \texttt{->} \langle u_1, \ldots, u_{|u|} \rangle &= \langle t_1, \ldots, t_{|t|}, u_1, \ldots, u_{|u|} \rangle \end{aligned}$$

$$\begin{aligned} . &: \mathit{tuple} \times \mathit{tuple} \to \mathit{tuple} \\ \langle t_1, \ldots, t_{|t|} \rangle . \langle u_1, \ldots, u_{|u|} \rangle &= \begin{cases} \langle t_1, \ldots, t_{|t|-1}, u_2, \ldots, u_{|u|} \rangle & \text{if } t_{|t|} = u_1 \\ \langle\rangle & \text{otherwise} \end{cases} \end{aligned}$$

Semantics: formulas

$$
\begin{aligned}
[\![\texttt{some } \Phi]\!]_\Gamma &= [\![\Phi]\!]_\Gamma \neq \varnothing \\
[\![\Phi \texttt{ in } \Psi]\!]_\Gamma &= [\![\Phi]\!]_\Gamma \subseteq [\![\Psi]\!]_\Gamma \\
[\![\texttt{not } \phi]\!]_\Gamma &= \neg [\![\phi]\!]_\Gamma \\
[\![\phi \texttt{ and } \psi]\!]_\Gamma &= [\![\phi]\!]_\Gamma \wedge [\![\psi]\!]_\Gamma \\
[\![\texttt{some } x\texttt{:}\Phi \mid \phi]\!]_\Gamma &= \bigvee \{ [\![\phi]\!]_{\Gamma \oplus x \mapsto \{t\}} \mid t \in [\![\Phi]\!]_\Gamma \}
\end{aligned}
$$

Semantics: expressions

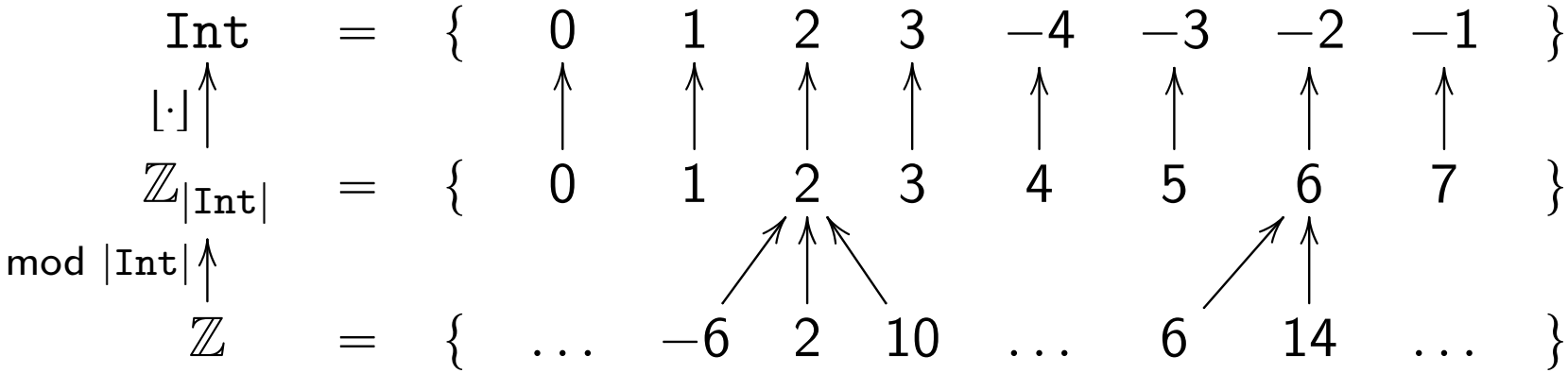
$$
\begin{aligned}
[\![x]\!]_\Gamma &= \Gamma(x) \\
[\![r]\!]_\Gamma &= \Gamma(r) \\
[\![\texttt{none}]\!]_\Gamma &= \varnothing \\
[\![\texttt{\~{}}\Phi]\!]_\Gamma &= \{\texttt{\~{}}t \mid t \in [\![\Phi]\!]_\Gamma\} \\
[\![\texttt{\^{}}\Phi]\!]_\Gamma &= [\![\Phi]\!]_\Gamma \cup [\![\Phi\texttt{.}\Phi]\!]_\Gamma \cup [\![\Phi\texttt{.}\Phi\texttt{.}\Phi]\!]_\Gamma \cup \ldots \\
[\![\Phi \texttt{ + } \Psi]\!]_\Gamma &= [\![\Phi]\!]_\Gamma \cup [\![\Psi]\!]_\Gamma \\
[\![\Phi \texttt{ - } \Psi]\!]_\Gamma &= [\![\Phi]\!]_\Gamma \setminus [\![\Psi]\!]_\Gamma \\
[\![\Phi \texttt{ \& } \Psi]\!]_\Gamma &= [\![\Phi]\!]_\Gamma \cap [\![\Psi]\!]_\Gamma \\
[\![\Phi \texttt{ . } \Psi]\!]_\Gamma &= \{t.u \mid t \in [\![\Phi]\!]_\Gamma \wedge u \in [\![\Psi]\!]_\Gamma \wedge t.u \neq \langle\rangle\} \\
[\![\Phi \texttt{ -> } \Psi]\!]_\Gamma &= \{t\texttt{->}u \mid t \in [\![\Phi]\!]_\Gamma \wedge u \in [\![\Psi]\!]_\Gamma\} \\
[\![\{x\texttt{:}\Phi \mid \phi\}]\!]_\Gamma &= \{t \mid t \in [\![\Phi]\!]_\Gamma \wedge [\![\phi]\!]_{\Gamma \oplus x \mapsto \{t\}}\}
\end{aligned}
$$

Semantics: integers with wrap-around arithmetic

- Given a scope n for `Int` we have

$$\texttt{Int} = \{-2^{n-1} \ldots 2^{n-1} - 1\}$$

- Arithmetic operations may overflow.
- To make them closed under `Int` Alloy default semantics is wrap-around.
- Equivalent to modular arithmetic:



Semantics: integers with wrap-around arithmetic

$$
\begin{aligned}
[\![i]\!]_\Gamma &= \{\lfloor i \mod |\mathtt{Int}| \rfloor\} \\
[\![\mathtt{int}[\Phi]]\!]_\Gamma &= \{\lfloor \textstyle\sum_{i \in [\![\Phi]\!]_\Gamma} i \mod |\mathtt{Int}| \rfloor\}
\end{aligned}
$$

$$
\begin{aligned}
[\![\mathtt{plus}[\Phi,\Psi]]\!]_\Gamma &= \{\lfloor [\![\mathtt{int}[\Phi]]\!]_\Gamma + [\![\mathtt{int}[\Psi]]\!]_\Gamma \mod |\mathtt{Int}| \rfloor\} \\
[\![\mathtt{eq}[\Phi,\Psi]]\!]_\Gamma &= [\![\mathtt{int}[\Phi]]\!]_\Gamma = [\![\mathtt{int}[\Psi]]\!]_\Gamma \\
[\![\mathtt{gte}[\Phi,\Psi]]\!]_\Gamma &= [\![\mathtt{int}[\Phi]]\!]_\Gamma \geqslant [\![\mathtt{int}[\Psi]]\!]_\Gamma
\end{aligned}
$$

Semantics: integers with forbid overflows

- How to avoid spurious counter-examples if one wants to forbid overflows?

```
check {
  all x : Int | x.plus[1].gte[x]
} for 3 Int
```

- Mask instances where overflows occur – filter out instances whose formula is undefined.
- Impact on the overal semantics since one must resort to three-valued logic and change the semantics of quantifiers.

Semantics: integers with forbid overflows

$$[\![\texttt{some } x\texttt{:Int} \mid \phi]\!]_\Gamma = \bigvee\{[\![\phi]\!]_{\Gamma\oplus x\mapsto\{i\}} \neq \bot \wedge [\![\phi]\!]_{\Gamma\oplus x\mapsto\{i\}} \mid i \in \texttt{Int}\}$$
$$[\![\texttt{all } x\texttt{:Int} \mid \phi]\!]_\Gamma = \bigwedge\{[\![\phi]\!]_{\Gamma\oplus x\mapsto\{i\}} = \bot \vee [\![\phi]\!]_{\Gamma\oplus x\mapsto\{i\}} \mid i \in \texttt{Int}\}$$

$$[\![\phi \texttt{ and } \psi]\!]_\Gamma \neq \bot \equiv ([\![\phi]\!]_\Gamma \wedge_3 [\![\psi]\!]_\Gamma) \neq \bot$$

$$[\![i]\!]_\Gamma \neq \bot \equiv 2^{n-1} \leqslant i < 2^{n-1}$$
$$[\![\texttt{plus[}\Phi\texttt{,}\Psi\texttt{]} \neq \bot]\!] \equiv [\![\texttt{int[}\Phi\texttt{]}]\!]_\Gamma \neq \bot \wedge [\![\texttt{int[}\Psi\texttt{]}]\!]_\Gamma \neq \bot \wedge$$
$$2^{n-1} \leqslant [\![\texttt{int[}\Phi\texttt{]}]\!]_\Gamma + [\![\texttt{int[}\Psi\texttt{]}]\!]_\Gamma < 2^{n-1}$$

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.

$$
\begin{aligned}
\texttt{Name} &: \{\langle \texttt{Name} \rangle\} \\
\texttt{Object} &: \{\langle \texttt{Link} \rangle, \langle \texttt{File} \rangle, \langle \texttt{Root} \rangle, \langle \texttt{\$Dir} \rangle\} \\
\texttt{Contents}_{\texttt{File}} &: \{\langle \texttt{File}, \texttt{Block} \rangle\} \\
\texttt{to} &: \{\langle \texttt{Link}, \texttt{Link} \rangle, \langle \texttt{Link}, \texttt{File} \rangle, \langle \texttt{Link}, \texttt{Root} \rangle, \langle \texttt{Link}, \texttt{\$Dir} \rangle\}
\end{aligned}
$$

Bounding types

- Expression Φ has bounding type A given binding Γ:

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

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

$$\frac{\Gamma \vdash \Phi : A}{[\![\Phi]\!]_\lambda \subseteq [\![A]\!]_\lambda}$$

Bounding types: expressions

$$\frac{}{\Gamma \vdash x : \Gamma(x)} \qquad \frac{}{\Gamma \vdash r : \Gamma(r)} \qquad \frac{}{\Gamma \vdash \texttt{none} : \varnothing}$$

$$\frac{\Gamma \vdash \Phi : A}{\Gamma \vdash \texttt{\~{}}\Phi : \texttt{\~{}}A} \qquad \frac{\Gamma \vdash \Phi : A}{\Gamma \vdash \texttt{\^{}}\Phi : \texttt{\^{}}A} \qquad \frac{\Gamma \vdash \Phi : A}{\Gamma \vdash \Phi \texttt{ - } \Psi : A}$$

$$\frac{\Gamma \vdash \Phi : A \qquad \Gamma \vdash \Psi : B}{\Gamma \vdash \Phi \texttt{ + } \Psi : A \texttt{ + } B} \qquad \frac{\Gamma \vdash \Phi : A \qquad \Gamma \vdash \Psi : B}{\Gamma \vdash \Phi \texttt{ \& } \Psi : A \texttt{ \& } B}$$

$$\frac{\Gamma \vdash \Phi : A \qquad \Gamma \vdash \Psi : B}{\Gamma \vdash \Phi \texttt{ . } \Psi : A \texttt{ . } B} \qquad \frac{\Gamma \vdash \Phi : A \qquad \Gamma \vdash \Psi : B}{\Gamma \vdash \Phi \texttt{ -> } \Psi : A \texttt{ -> } B}$$

$$\frac{\Gamma \vdash \Phi : A \qquad \Gamma \oplus x \mapsto A \vdash \phi}{\Gamma \vdash \{x : \Phi \mid \phi\} : A}$$

Bounding types: formulas

$$\frac{\Gamma \vdash \Phi : A}{\Gamma \vdash \texttt{some}\ \Phi} \qquad \frac{\Gamma \vdash \Phi : A \qquad \Gamma \vdash \Psi : B}{\Gamma \vdash \Phi\ \texttt{in}\ \Psi}$$

$$\frac{\Gamma \vdash \phi}{\Gamma \vdash \texttt{not}\ \phi} \qquad \frac{\Gamma \vdash \phi \qquad \Gamma \vdash \psi}{\Gamma \vdash \phi\ \texttt{and}\ \psi}$$

$$\frac{\Gamma \vdash \Phi : A \qquad \Gamma \oplus x \mapsto A \vdash \phi}{\Gamma \vdash \texttt{some}\ x\texttt{:}\Phi\ \texttt{|}\ \phi}$$

Bounding types: example

$\Gamma \vdash \texttt{contents}_{\texttt{D}} : \{\langle \texttt{D}, \texttt{L}\rangle, \langle \texttt{D}, \texttt{F}\rangle, \langle \texttt{D}, \texttt{R}\rangle, \langle \texttt{D}, \texttt{D}\rangle, \langle \texttt{R}, \texttt{L}\rangle, \langle \texttt{R}, \texttt{F}\rangle, \langle \texttt{R}, \texttt{R}\rangle, \langle \texttt{R}, \texttt{D}\rangle\}$

$\Gamma \vdash \texttt{contents}_{\texttt{F}} : \{\langle \texttt{F}, \texttt{B}\rangle\}$

$\Gamma \vdash \texttt{contents} : \{\langle \texttt{D}, \texttt{L}\rangle, \langle \texttt{D}, \texttt{F}\rangle, \langle \texttt{D}, \texttt{R}\rangle, \langle \texttt{D}, \texttt{D}\rangle, \langle \texttt{R}, \texttt{L}\rangle, \langle \texttt{R}, \texttt{F}\rangle, \langle \texttt{R}, \texttt{R}\rangle, \langle \texttt{R}, \texttt{D}\rangle, \langle \texttt{F}, \texttt{B}\rangle\}$

$\Gamma \vdash \texttt{Root} : \{\langle \texttt{R}\rangle\}$

$\Gamma \vdash \texttt{Root.contents} : \{\langle \texttt{L}\rangle, \langle \texttt{F}\rangle, \langle \texttt{R}\rangle, \langle \texttt{D}\rangle\}$

$\Gamma \vdash \texttt{Root + Root.contents} : \{\langle \texttt{L}\rangle, \langle \texttt{F}\rangle, \langle \texttt{R}\rangle, \langle \texttt{D}\rangle\}$

$\Gamma \vdash \texttt{to} : \{\langle \texttt{L}, \texttt{L}\rangle, \langle \texttt{L}, \texttt{F}\rangle, \langle \texttt{L}, \texttt{R}\rangle, \langle \texttt{L}, \texttt{D}\rangle\}$

$\Gamma \vdash \texttt{(Root + Root.contents).to} : \{\langle \texttt{L}\rangle, \langle \texttt{F}\rangle, \langle \texttt{R}\rangle, \langle \texttt{D}\rangle\}$

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

Relevance types

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

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

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

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

- 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 τ.

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

Relevance types: expressions

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

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

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

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

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

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

Relevance types: expressions

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

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

$$\frac{\Gamma \vdash \Phi : A \qquad \Gamma \vdash \Psi : B \qquad \Gamma \vdash \mathcal{C} \downarrow \Phi \text{ -> } \Psi : C}{\Gamma \vdash \mathcal{C}[\bullet \text{ -> } \Psi] \downarrow \Phi : \{t \mid t \in A \wedge u \in B \wedge t\text{->}u \in C\}}$$

$$\frac{\Gamma \vdash \Phi : A \qquad \Gamma \vdash \Psi : B \qquad \Gamma \vdash \mathcal{C} \downarrow \Phi \text{ -> } \Psi : C}{\Gamma \vdash \mathcal{C}[\Phi \text{ -> } \bullet] \downarrow \Psi : \{u \mid t \in A \wedge u \in B \wedge t\text{->}u \in C\}}$$

Relevance types: expressions

$$\frac{\Gamma \vdash \Phi : A \qquad \Gamma \vdash \mathcal{C} \downarrow \tilde{}\Phi : B}{\Gamma \vdash \mathcal{C}[\tilde{}\bullet] \downarrow \Phi : \tilde{}B}$$

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

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

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

Relevance types: formulas

$$\frac{\Gamma \vdash \Phi : A \qquad \Gamma \vdash \mathcal{C} \downarrow \texttt{some}\ \Phi}{\Gamma \vdash \mathcal{C}[\texttt{some}\ \bullet] \downarrow \Phi : A}$$

$$\frac{\Gamma \vdash \Phi : A \qquad \Gamma \vdash \Psi : B \qquad \Gamma \vdash \mathcal{C} \downarrow \Phi\ \texttt{in}\ \Psi}{\Gamma \vdash \mathcal{C}[\bullet\ \texttt{in}\ \Psi] \downarrow \Phi : A}$$

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

Relevance types: formulas

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

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

$$\frac{\Gamma \vdash \Phi : A \qquad \Gamma \vdash \mathcal{C} \downarrow \texttt{some}\ x\texttt{:}\Phi\ \texttt{|}\ \phi}{\Gamma \vdash \mathcal{C}[\texttt{some}\ x\texttt{:}\ \bullet\ \texttt{|}\ \phi] \downarrow \Phi : A}$$

$$\frac{\Gamma \vdash \Phi : A \qquad \Gamma \vdash \mathcal{C} \downarrow \texttt{some}\ x\texttt{:}\Phi\ \texttt{|}\ \phi}{\Gamma \oplus x \mapsto A \vdash \mathcal{C}[\texttt{some}\ x\texttt{:}\Phi\ \texttt{|}\ \bullet] \downarrow \phi}$$

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

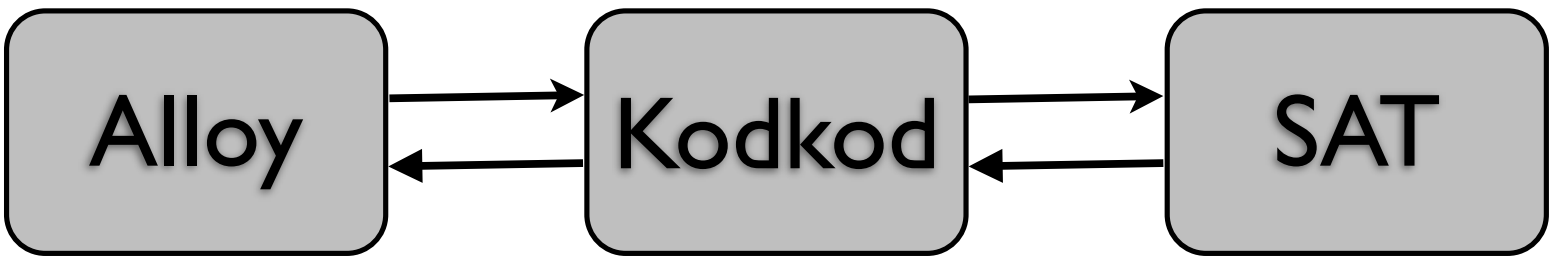
Relevance types: example

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

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
```

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

$$\varnothing \subseteq \texttt{Man}_1 \subseteq \{\langle A_1\rangle, \langle A_2\rangle\}$$
$$\varnothing \subseteq \texttt{\$Woman}_1 \subseteq \{\langle A_1\rangle, \langle A_2\rangle\}$$
$$\{\langle A_3\rangle\} \subseteq \texttt{Eve}_1 \subseteq \{\langle A_3\rangle\}$$
$$\varnothing \subseteq \texttt{wife}_2 \subseteq \{\langle A_1, A_1\rangle, \langle A_1, A_2\rangle, \langle A_1, A_3\rangle, \langle A_2, A_1\rangle, \langle A_2, A_2\rangle, \langle A_2, A_3\rangle\}$$

```
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.

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

- 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

$$\texttt{wife.Eve in Man+\$Woman+Eve}$$

$$\begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} \\ w_{2,1} & w_{2,2} & w_{2,3} \\ \bot & \bot & \bot \end{bmatrix} . \begin{bmatrix} \bot \\ \bot \\ \top \end{bmatrix} \texttt{ in } \begin{bmatrix} M_1 \\ M_2 \\ \bot \end{bmatrix} + \begin{bmatrix} W_1 \\ W_2 \\ \bot \end{bmatrix} + \begin{bmatrix} \bot \\ \bot \\ \top \end{bmatrix}$$

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

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

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.

$$\texttt{some adam : Man | adam.wife = Eve}$$
$$\equiv$$
$$\{\langle A_1 \rangle\}\texttt{.wife = Eve or } \{\langle A_2 \rangle\}\texttt{.wife = Eve}$$

- 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.

$$\texttt{some adam : Man | adam.wife = Eve}$$
$$\cong$$
$$\varnothing \subseteq \texttt{\$adam}_1 \subseteq \{\langle A_1 \rangle, \langle A_2 \rangle\}$$
$$\texttt{one \$adam and \$adam.wife = Eve}$$

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

Skolemization

$$\texttt{all m : Man | some w : Woman | m.wife = w}$$
$$\cong$$
$$\varnothing \subseteq \texttt{w}_2 \subseteq \{\langle A_1, A_1\rangle, \langle A_1, A_2\rangle, \langle A_1, A_3\rangle, \langle A_2, A_1\rangle, \langle A_2, A_2\rangle, \langle A_2, A_3\rangle\}$$
$$\texttt{all m : Man | one m.\$w and and m.wife = m.\$w}$$

$$\texttt{some husband : Woman -> Man | husband = ~wife}$$
$$\cong$$
$$\varnothing \subseteq \texttt{\$husband}_2 \subseteq \{\langle A_1, A_1\rangle, \langle A_1, A_2\rangle, \langle A_2, A_1\rangle, \langle A_2, A_2\rangle, \langle A_3, A_1\rangle, \langle A_3, A_2\rangle\}$$
$$\texttt{\$husband = wife}$$

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.