Translating Alloy Specifications to the Point-free Style

Nuno Macedo
Alcino Cunha

HASlab
Universidade do Minho
Braga, Portugal

June 28th, 2011

Alloy

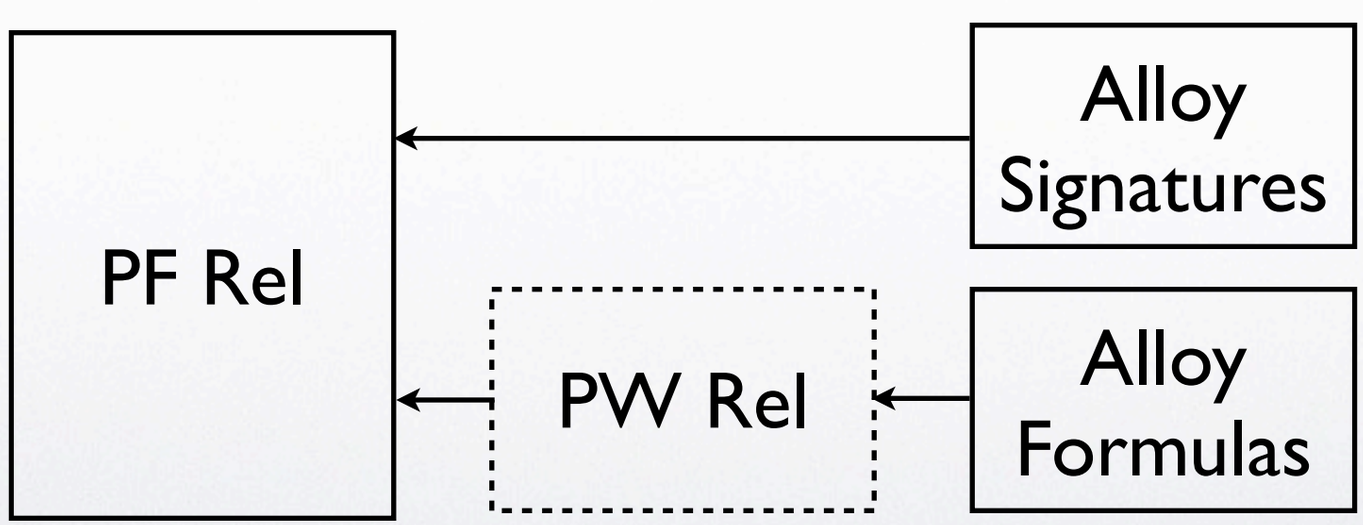
- Lightweight modeling language;
- Simple language, based on simple mathematical notations;
- Characteristics of object modeling;
- Automatic bounded verification.

Motivation

- *Alloy* provides a tool for automatic *bounded* verification (the *Alloy Analyzer*);
- Sometimes however, *unbounded* verification is necessary;
- Alloy's logic is a *relational*, so relational frameworks are natural choices;
- The *point-free* (PF) style provides formulas simple enough for manipulation and analysis.

Objectives

- A complete translation of Alloy models to a PF relational framework is proposed.



Challenges

Alloy's logic gives rise to some challenges:

- Obtaining formulas as *simple* as the original ones;
- Representing and combining relations of *any arity*;
- Dealing with the very *loose type system* of Alloy;
- Representing the *signature*'s hierarchy and properties;

Relational calculus provides a solution for all of these.

Formula Translation

$$\textbf{all}\ \mathrm{a,b} : \mathrm{A} \mid (\textbf{some}\ \mathrm{c} : \mathrm{C} \mid \mathrm{c} = \mathrm{r}\cdot\mathrm{a}\ \&\&\ \mathrm{c} = \mathrm{r}\cdot\mathrm{b}) \Rightarrow \mathrm{a} = \mathrm{b}$$

$$\downarrow$$

$$\langle \forall a, b \in A :: \langle \exists c \in C :: a\, R\, c \wedge b\, R\, c \rangle \Rightarrow a = b \rangle$$

$$\downarrow$$

$$\top \subseteq \overline{((\top \cdot (\pi_1 \cdot \pi_2 \cap R \cdot \pi_2) \cap \top \cdot (\pi_2 \cdot \pi_2 \cap R \cdot \pi_2)) \cdot id\nabla\top \cap \overline{\top \cdot (\pi_1 \cap \pi_2)}) \cdot id\nabla\top \cdot \top}$$

$$\vdots$$

$$r \cdot r^{\circ} \subseteq id$$

Example

Alloy model

```
abstract sig Person {}
sig Student, Professor extends Person {}
sig Course {
  lecturer : some Professor,
  depends : set Course
}
sig University {
  enrolled : set Student,
  courses  : Student -> Course
}
pred inv[u : University] {
  (u.courses).Course in u.enrolled
  all s : Student |
     (s.(u.courses)).*depends in s.(u.courses)
}
pred enroll[u, u' : University, s : Student] {
  u'.enrolled = u.enrolled + s
  u'.courses = u.courses
}
assert {
  all u,u':University,s:Student |
     inv[u] and enroll[u,u',s] => inv[u']
}
```

FA model

Signature facts

$$id = \Phi_{Person} \cup \Phi_{Course} \cup \Phi_{University}$$

$$\Phi_{Student} \cup \Phi_{Professor} \subseteq \Phi_{Person} \wedge \Phi_{Student} \cap \Phi_{Professor} = \bot$$

$$lecturer \subseteq \Phi_{Course} \cdot \top \cdot \Phi_{Professor}$$

$$enrolled \subseteq \Phi_{University} \cdot \top \cdot \Phi_{Student}$$

$$courses \subseteq \Phi_{University} \cdot \top \cdot \Phi_{Student} \times \Phi_{Course}$$

$$depends \subseteq \Phi_{Course} \cdot \top \cdot \Phi_{Course}$$

$$id \subseteq lecturer \cdot lecturer^{\circ}$$

Assertion

$$(\Phi_U \times \Phi_U \times \Phi_S) \cap c_1/(e_1 \cdot \pi_1) \cap (c_1 \cdot (\Phi_S \times d^{*\circ}))/c_1$$
$$\cap$$
$$c_1/c_2 \cap c_2/c_1 \cap e_1/e_2 \cap e_2 \cdot \pi_2 \cdot \pi_2 \cap e_2/(e_1 \cup id_3)$$
$$\subseteq$$
$$c_2/(e_2 \cdot \pi_1) \cap (c_2 \cdot (\Phi_S \times d^{*\circ}))/c_2$$

Conclusions

- *Complete* and *automatic* translation of Alloy models;
- Due to the simplicity, it is suitable for *manual* verification;
- *Automatic* verification is also possible, e. g., Prover9 automatically verified the previous example;

Translating Alloy Specifications to the Point-free Style

Nuno Macedo
Alcino Cunha

HASlab
Universidade do Minho
Braga, Portugal

June 28th, 2011