# Translating Alloy specifications to the point-free style

Nuno Macedo

Departamento de Informática
Universidade do Minho
Braga, Portugal

September 28th, 2010

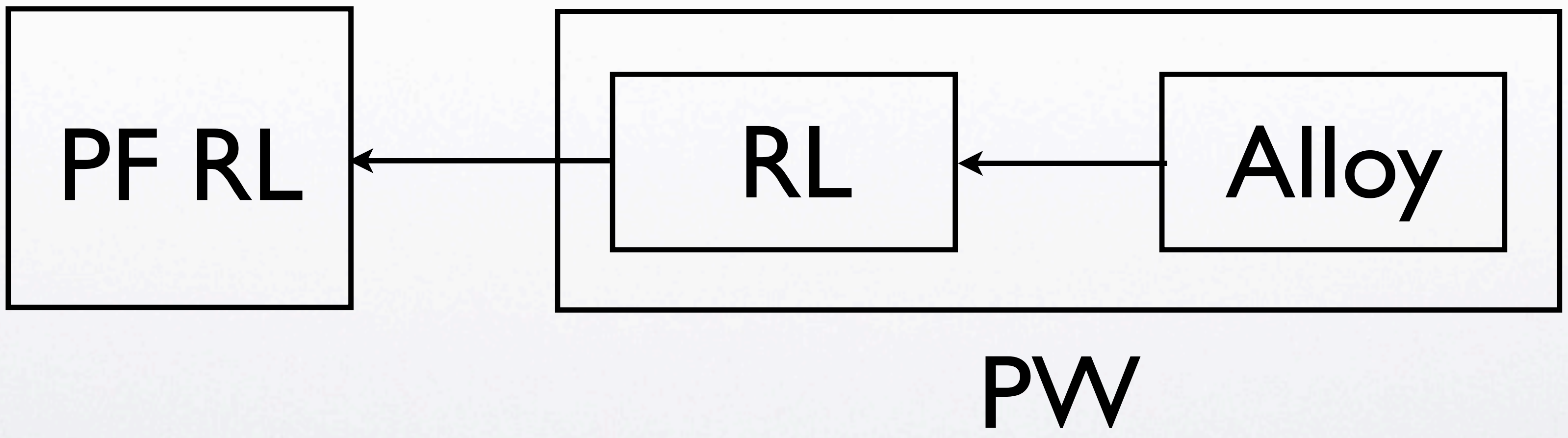
*Prova de dissertação para obtenção do grau de Mestre em Informática*

# Motivation

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

# Objectives

- A translation of Alloy models to a PF relational logic (RL) is proposed.



# Alloy

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

# Calculus of Relations

- *Relational Logic* (RL):
  - First-order logic enhanced with relational operators (composition, meet, join,...)

$$\langle \forall a, b :: a\,R\,b \Rightarrow a\,S\,b \rangle$$

- *Calculus of Relations* (CR):
  - RL without variables

$$R \subseteq S$$

  - Equivalent to RL with 3 quantified variables.

# PF Relational Logic

- Created to overcome the lack of expressiveness of CR;
- Fork Algebras (FA):
  - Introduces a pairs and a new operator *fork:*

$$c\,R\,a \wedge b\,S\,a \equiv (c,b)\,R\nabla S\,a$$

  - Equivalent to RL;
- Categorical CR (CCR):
  - Is able to extend CR and FA with types;
  - Alloy is typed, so it is better suited.

# Alloy to FA translation

- Marcelo Frias, Carlos Pombo and Nazareno Aguirre, *An equational calculus for Alloy*;
- Created to translate (only) Alloy *formulas* to FA;
- Resulting formulas are extremely complex:

**all** a : A | **some** c : C | c **in** r·a

↓

$$\overline{\overline{\overline{\overline{\top \cdot \rho(\langle id, \pi_2 \cdot \phi \rangle)} \cdot \langle \pi_1 \cdot \pi_1, \pi_2 \cdot \pi_1, \pi_2, \top \rangle \cdot \langle \pi_1, \pi_2, \top \rangle} \cdot \langle id, \top \rangle} \cdot \top} = \top$$

$$\phi = id \times \top \cap \overline{\delta(\pi_2 \cdot \pi_1 \cap \pi_2) \cap \overline{\rho(id \times R \cdot \delta(\pi_2 \cdot \pi_1 \cap \pi_2))} \cap id \times \top}$$

# Alloy to CCR

- Two-step translation:
  - Alloy to RL: fully expands Alloy formulas to PF
  - RL to CCR: mechanic PW to PF translation, enhanced with heuristics

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

$$\downarrow$$

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

$$\downarrow$$

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

$$\vdots$$

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

# Type System

- Types are represented by the *sum* of their sub-types;
- Alloy allows the combination of *any types* with the same arity; $\texttt{all}\ a : A, b : B \mid a = b$
- Operations are defined to convert two types to their least common super-type.

# N-ary relations

- N-ary relations are represented as binary relations with the range as tuples:

$$A \leftarrow B \leftarrow C \;\rightsquigarrow\; A \times B \leftarrow C$$

- New operators to manipulate n-ary relations:
  - N-ary composition:

$$(a, b)\, R \bullet S\, c \equiv \langle \exists k :: a\, R\, k \wedge (k, b)\, S\, c \rangle$$

  - Rotate:

$$(a, b)\, \overrightarrow{R}\, c \equiv (c, a)\, R\, b$$

# Multiplicities

- Signature multiplicities:

| Sig A | Property |
|---|---|
| `set` | *true* |
| `some` | $\top_{univ \leftarrow univ} \subseteq \top_{univ \leftarrow A} \cdot \top_{A \leftarrow univ}$ |
| `lone` | $\top_{A \leftarrow A} \subseteq id_A$ |
| `one` | $\top_{univ \leftarrow univ} \subseteq \top_{univ \leftarrow A} \cdot \top_{A \leftarrow univ} \wedge \top_{A \leftarrow A} \subseteq id_A$ |

- Relation multiplicities:

| $i$-th field | Property |
|---|---|
| `set` | *true* |
| `some` | $id \subseteq \overset{n-i}{\overrightarrow{R}} \cdot (\overset{n-i}{\overrightarrow{R}})^\circ$ |
| `lone` | $(\overset{n-i}{\overrightarrow{R}})^\circ \cdot \overset{n-i}{\overrightarrow{R}} \subseteq id$ |
| `one` | $id \subseteq \overset{n-i}{\overrightarrow{R}} \cdot (\overset{n-i}{\overrightarrow{R}})^\circ \wedge (\overset{n-i}{\overrightarrow{R}})^\circ \cdot \overset{n-i}{\overrightarrow{R}} \subseteq id$ |

# Example

- Alloy model

```
sig System {
  store: Path → lone File,
  open: Handle → lone OpenInfo
}
sig Path {
  dir: one Path
}
sig Root extends Path { }
sig OpenInfo{
  path: one Path
}
sig File {}
sig Handle {}
abstract sig Type {}
one sig Reg extends Type {}
one sig Dir extends Type {}

assert ri_ok{
  all s : System | all h: Handle, o: h·(s·open) |
            some f: File | f in (o·path)·(s·store)
}
```

- CCR model

$$Type \cong Dir + Reg$$
$$Path \cong Path' + Root$$
$$univ \cong System + Path' + Root + OpenInfo + File + Dir + Reg + Handle$$

Types

$$store :: System \times (Path' + Rool) \leftarrow File$$
$$open :: System \times Handle \leftarrow OpenInfo$$
$$dir :: (Path' + Root) \leftarrow (Path' + Root)$$
$$path :: OpenInfo \leftarrow (Path' + Root)$$

$$\top_{univ \leftarrow univ} \subseteq \top_{univ \leftarrow Dir} \cdot \top_{Dir \leftarrow univ} \wedge \top_{Dir \leftarrow Dir} \subseteq id_{Dir}$$
$$\top_{univ \leftarrow univ} \subseteq \top_{univ \leftarrow Reg} \cdot \top_{Reg \leftarrow univ} \wedge \top_{Reg \leftarrow Reg} \subseteq id_{Reg}$$

Multiplicity

$$store^\circ \cdot store \subseteq id$$
$$open^\circ \cdot table \subseteq id$$
$$dir^\circ \cdot dir \subseteq id \wedge id \subseteq dir \cdot dir^\circ$$
$$path^\circ \cdot path \subseteq id \wedge id \subseteq path \cdot path^\circ$$

Assert $\langle \forall\, s, h, o : (s, h)\ table\ o : \langle \exists f, i :: o\ path\ i \wedge (s, i)\ store\ f \rangle \rangle$

$$\top \cdot (\pi_1 \cap table \cdot \pi_2) \subseteq \top \cdot (\pi_2 \times id \cap path \bullet \overset{2}{\overrightarrow{store}} \cdot \pi_1 \cdot \pi_1 \cdot \pi_1) \cdot \langle id, \top \rangle$$

# Conclusions

- Defined and implemented a highly improved version of the Alloy to PF RL translation;
- Translates not only the formulas but also the signature declarations;
- New operators to deal with n-ary relations, with interesting properties;
- Due to the simplicity, it is suitable for manual or assisted verification.

# Translating Alloy specifications to the point-free style


Nuno Macedo

Departamento de Informática
Universidade do Minho
Braga, Portugal


September 28th, 2010

*Prova de dissertação para obtenção do grau de Mestre em Informática*