

THE LOGIC OF ALLOY

CURT CLIFTON

ROSE-HULMAN INSTITUTE OF TECHNOLOGY

HOPE TO LEARN

- PRACTICAL TECHNIQUES
- WHY DISCO MATTERS



FORMAL METHODS—THE BFG OF
DEVELOPMENT TECHNIQUES

CONCERNS

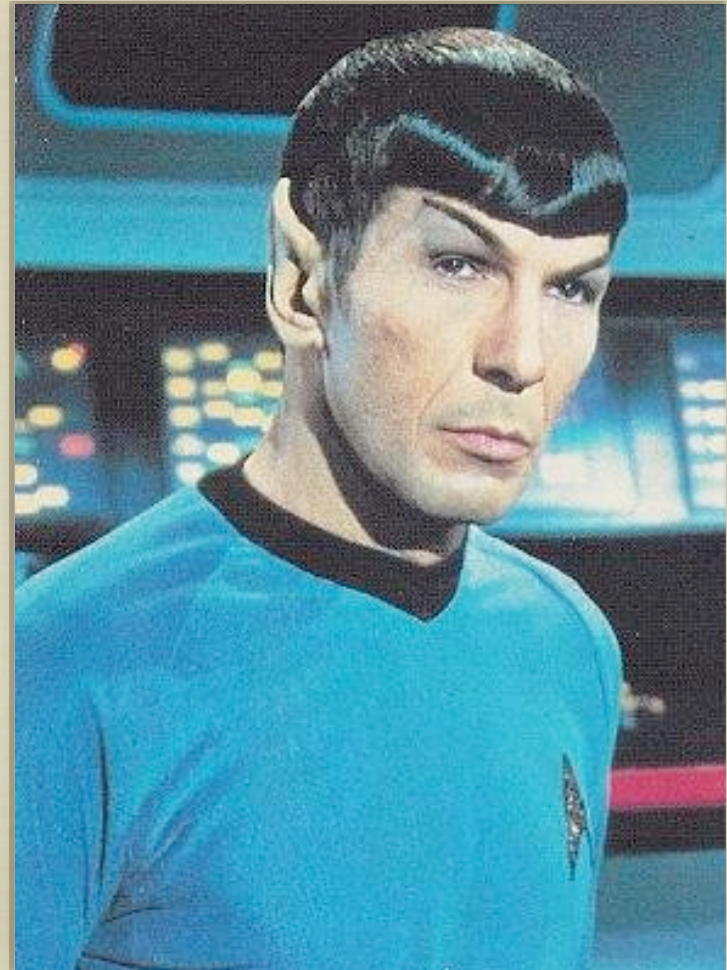
- REMEMBERING DISCO
- DOING PROOFS
- COURSE LOAD (AT LEAST 3 TAKING PLC NOW)
- USEFUL?

LABELS IN THE VISUALIZER

- LOOK AT EXAMPLE MODEL `addressBook1.g.als`
- WHAT IS THE COUNTEREXAMPLE FOR `delUndoesAdd`?

THE LOGIC OF ALLOY

- **EVERYTHING IS EITHER AN ATOM OR A RELATION**
- **RELATIONS ARE SETS OF TUPLES**
- **LIKE TABLES IN RELATIONAL DATABASES**



ATOMS

- INDIVISIBLE
- IMMUTABLE
- UNINTERPRETED
- EXAMPLES:
 - Book\$1, Name\$0, Addr\$2

TO SAY ANYTHING
INTERESTING
ABOUT ATOMS IN
ALLOY WE USE
RELATIONS

RELATIONS

- **A SET OF TUPLES**
- **TERMS:**
 - **SIZE—NUMBER OF TUPLES (OR ROWS)**
 - **ARITY—LENGTH OF TUPLES (OR NO. OF COLS.)**
 - **UNARY MEANS ARITY OF 1**
 - **BINARY MEANS ARITY OF 2**
 - **TERNARY MEANS ARITY OF 3**

CARTOON OF THE DAY

DEAR PHYSICISTS,

YOU KNOW HOW YOU, LIKE,
ALWAYS STEAL ALL OUR IDEAS
AND THEN YOU, LIKE, MAKE UP
YOUR OWN CRAZY NOTATIONS
AND SHIT?

YEAH... CUT THAT OUT.

LOVE,
MATHEMATICIANS
 $\langle X, 0 \rangle$

全
吉
博

ABUSE OF NOTATION-1

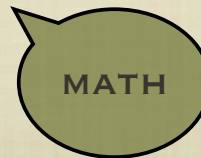
- SETS IN ALLOY?

- JUST UNARY RELATIONS

- EXAMPLES:

- $\text{Name} = \{(\text{Name}\$0), (\text{Name}\$1), (\text{Name}\$2)\}$

- $\text{Addr} = \{(\text{Addr}\$0), (\text{Addr}\$1)\}$



ABUSE OF NOTATION-2

- **SCALARS IN ALLOY?**

- **JUST SINGLETON, UNARY RELATIONS**

- **EXAMPLES:**

- `myName = {(Name$0)}`

- `yourAddr = {(Addr$1)}`

ABUSE OF NOTATION-3

- **OPTIONS IN ALLOY**

- **REPRESENTED AS A RELATION THAT IS A SCALAR OR IS EMPTY**

- **EXAMPLE:**

```
sig Addr {}
```

```
sig Message {  
    replyTo: set Addr
```

```
    ...
```

```
}
```

```
fact {
```

```
    all m: Message | #m.replyTo = 0 or #m.replyTo = 1
```

```
}
```

ABUSE OF NOTATION-4

- **TUPLES IN ALLOY**

- **JUST SINGLETON, NON-UNARY RELATIONS**

- **EXAMPLE:**

- `myName = {(Name$0)}`

- `myAddr = {(Addr$0)}`

- `myAssociation = {(Name$0, Addr$0)}`



CALL THIS A
"TUPLE"

WHY ALL THE ABUSE?

- **ALLOWS UNIFORM USE OF ALMOST ALL OPERATORS**
- **VERY FEW SPECIAL CASES**
- **VERY FEW TYPE ERRORS**
- **OK, BECAUSE NO “HIGHER-ORDER” RELATIONS**

THREE STYLES IN ALLOY

DON'T BE SCARED
BY THE \$5 WORDS!

■ PREDICATE CALCULUS STYLE:

- **all** n: Name, d, d': Addr |
n->d **in** addr **and** n->d' **in** addr **implies** d=d'

■ NAVIGATION EXPRESSION STYLE:

- **all** n: Name | **lone** n.addr

■ RELATIONAL CALCULUS STYLE:

- **no** ~addr.addr - **iden**

RELATIONS FOR STRUCTURE

- HOTEL KEY CARDS WITH TWO CRYPTO KEYS EACH
- STOCK VALUES OVER TIME
- COURSE PREREQUISITES

HOTEL KEY CARDS

```
sig Card {  
  fst: Key,  
  snd: Key  
}
```

```
sig Key {}
```

```
fact {  
  all c: Card | c.fst != c.snd  
}
```

```
pred show {}
```

```
run show for 4 but 2 Card
```

STOCK VALUES OVER TIME

```
sig Stock { history: Time -> Value }  
sig Value {}  
sig Time {}
```

```
fact {  
  // A stock has just one value for a given point in time:  
  all s: Stock, t: Time | #s.history[t] <= 1  
}
```

```
pred show {  
  #Stock = 3  
}
```

```
run show for 4
```

COURSE PRE-REQS

```
sig Course { prereqs: set Prereq }
```

```
sig Prereq { courses: set Course }
```

```
fact {
```

```
  // a course may not be its own prereq, directly or transitively
```

```
  all c: Course | c not in c.^(prereqs.courses)
```

```
  // only allow prereqs sets that non-empty and used by some course
```

```
  all p: Prereq | some p.courses and p in Course.prereqs
```

```
}
```

```
pred show {
```

```
  // ensures that some prereq consists of more than 2 course,
```

```
  // just to make things interesting
```

```
  some p: Prereq | #p.courses > 2
```

```
}
```

```
run show for 6
```