

THE LOGIC OF ALLOY

CURT CLIFTON

ROSE-HULMAN INSTITUTE OF TECHNOLOGY

LABELS IN THE VISUALIZER

- LOOK AT EXAMPLE MODEL `addressBook1g.als`
- WHAT IS THE COUNTEREXAMPLE FOR `delUndoesAdd?`

SEE TASK 25 IN THE
ALLOY LAB

FROM HW1: HOPE TO LEARN

- PRACTICAL TECHNIQUES
- WHY DISCO MATTERS



FORMAL METHODS—THE
BFG OF DEVELOPMENT
TECHNIQUES

FROM HW1: CONCERNS

- DIFFICULTY OF COURSE
- REMEMBERING DISCO
- DOING PROOFS
- COURSE LOAD (SEVERAL TAKING PLC NOW)
- NOTATION
- RELEVANCE

GOALS FOR TODAY

- REMEMBER SOME SET THEORY IDEAS AND SEE HOW THEY'RE USED IN ALLOY
- SEE HOW WE CAN USE RELATIONS TO REPRESENT DESIGN MODELS

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
日
漫

ABUSE OF NOTATION-1

- **SETS IN ALLOY?**

- **JUST UNARY RELATIONS**

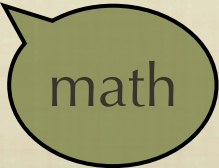
- **EXAMPLES:**

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

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



Alloy



math

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:**

- $\text{myName} = \{(\text{Name}\$0)\}$

- $\text{myAddr} = \{(\text{Addr}\$0)\}$

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



CALL THIS A
"TUPLE"

WHY ALL THE ABUSE?

- **ALLOWS UNIFORM USE OF ALMOST ALL OPERATORS**
- **VERY FEW SPECIAL CASES**
- **VERY FEW TYPE ERRORS**
- **NO TYPE CASTS/CONVERSIONS**
- **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 DESCRIBING STRUCTURE

JUST A SKETCH SO YOU CAN SEE
WHERE WE'RE HEADED.

HOTEL KEY CARDS



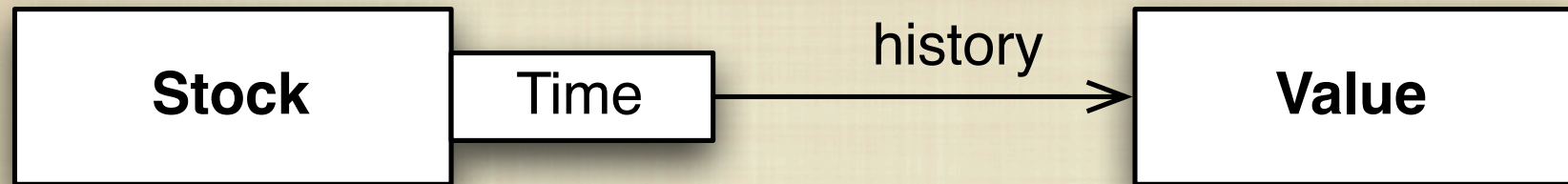
```
sig Card {
  first: CryptoKey,
  second: CryptoKey
}
```

```
sig CryptoKey {}
```

```
fact {
  all c: Card | c.first != c.second
}
```

```
...
```

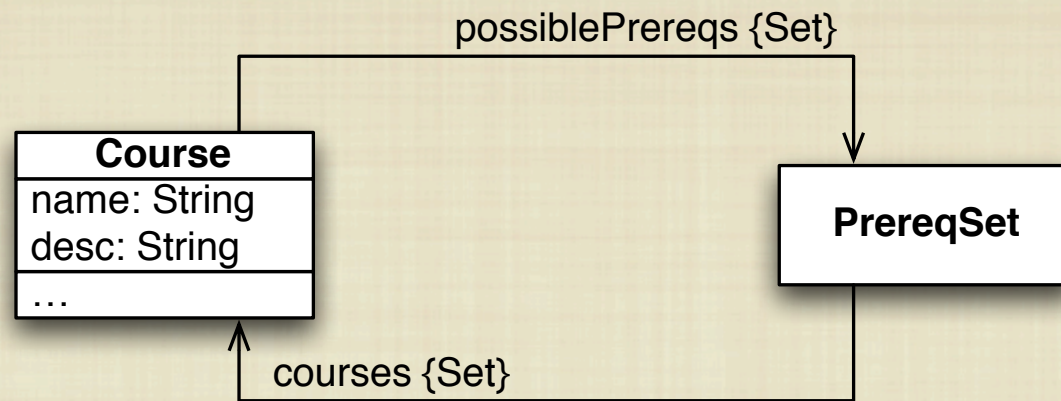
STOCK VALUES OVER TIME



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

```
fact {  
    all s: Stock, t: Time | #s.history[t] <= 1  
}  
...
```

COURSE PREREQUISITES



```
sig Course { prereqs: set PrereqSet }
sig PrereqSet { 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 prereq sets that are non-empty and used by some course
  all p: PrereqSet | some p.courses and p in Course.prereqs
}
...
```

ALLOY MODELS OF
EXAMPLES FOR YOUR
REFERENCE

HOTEL KEY CARDS

```
sig Card {  
  first: CryptoKey,  
  second: CryptoKey  
}
```

```
sig CryptoKey {}
```

```
fact {  
  all c: Card | c.first != c.second  
}
```

```
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 PrereqSet }  
sig PrereqSet { 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 prereq sets that are non-empty and used by some course  
  all p: PrereqSet | 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: PrereqSet | #p.courses > 2  
}
```

```
run show for 6
```