Alloy Modeling Language

Curt Clifton
Rose-Hulman Institute of Technology

Goals for Today

- Wrap up the logic of constraints from last time
- Learn the basics of modeling "classes" in Alloy using signatures

Alloy Logic Wrap-up

Quantified Expressions

- **some** Name
- **some** address
- **no** (address.Addr - Name)
- **all** n: Name | **lone** n.address

```
sig Address {}
sig Name {
    // multi-level address book
    address: set (Name + Address)
}
```

Let expressions

- **let** x = e | A
- Just a shorthand to avoid writing out e multiple times
- **all** a: Alias |
  let w = a.workAddress |
  a.address = (**some** w => w **else** a.homeAddress)

Variable and Formal Declarations

- name: expression
  - name is a **subset** of the relation given by expression
- **Examples:**
  - address: Name->Addr
  - addr: Book->Name->Addr
  - address: Name->(Name + Addr)
  - workAddress, homeAddress: Alias->Addr
    prefAddress: workAddress + homeAddress

Set Multiplicities

- Used to constrain the possible subsets that a variable can be
  - x: **set** e – x can be any subset of e
  - x: **one** e – x is a singleton subset of e (i.e., an Alloy scalar)
  - x: **lone** e – x is an option, either empty set or a scalar
  - x: **some** e – x is a non-empty subset of e

Careful: if e is a unary relation (i.e., a set), then x: e is equivalent to x: **one** e

Relation Multiplicities

- Too bizarre for words
- Almost
- r: A m -> n B means:
  - **Each** member of A maps to **n** members of B
  - and for **each** member of B, **m** members of A map to it

Examples:

r: A -> **one** B
r: A **one** -> B
r: A -> **lone** B
r: A **one** -> **one** B
r: A **some** -> **some** B

```
sig Thing, OtherThing {}

pred relMult[r: Thing some -> some OtherThing]  {}
run relMult for 3
```

old Q8

Cardinality Constraints

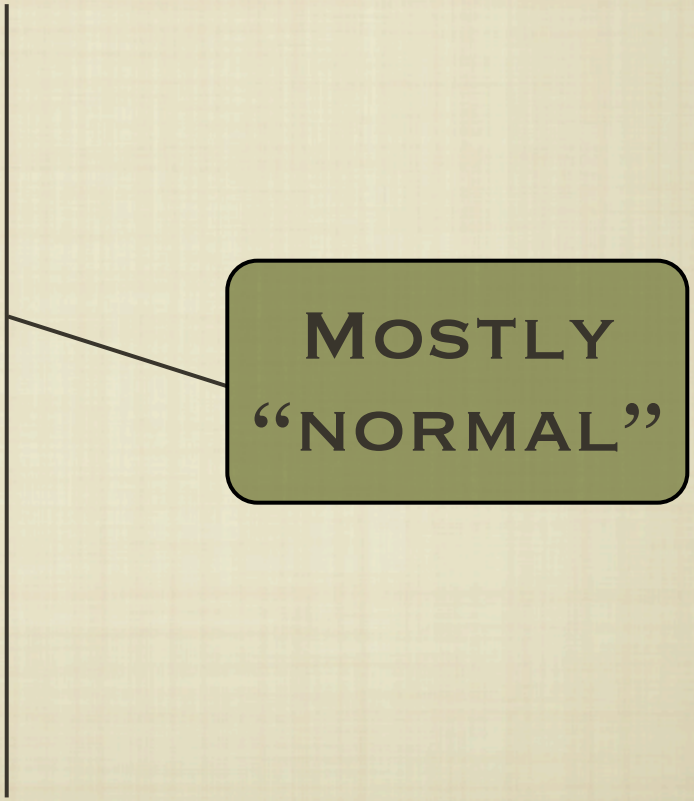
- #e gives the size (number of tuples) in the relation given by e
- Can use all regular integer operations on the result
- Can use 1, 2, 3, ... as constants
- **sum** x: e | ie means $\sum_{x \in e} ie$

HTTP://WWW.CATHOLICVOTEACTION.ORG

Alloy Modeling Language

Alloy Language

- Based on the logic
- Adds structuring tools:
  - Modules
  - Composition
  - Re-use
- Adds syntactic sugars



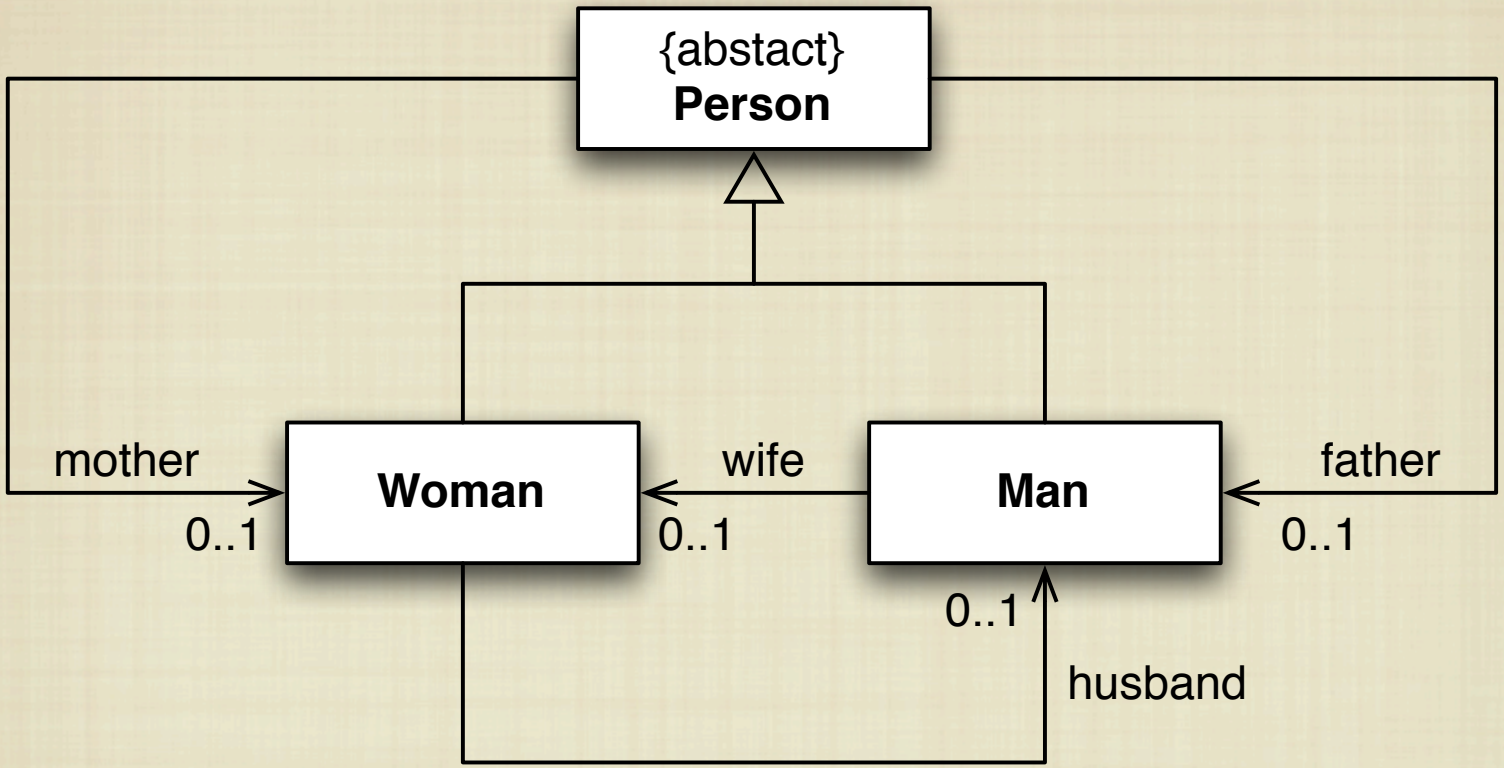
Extended Example

- Self-grandpas
- Background:
  - http://www.youtube.com/watch?v=W7x1ETPkZsk

www.cmt.com

Q1

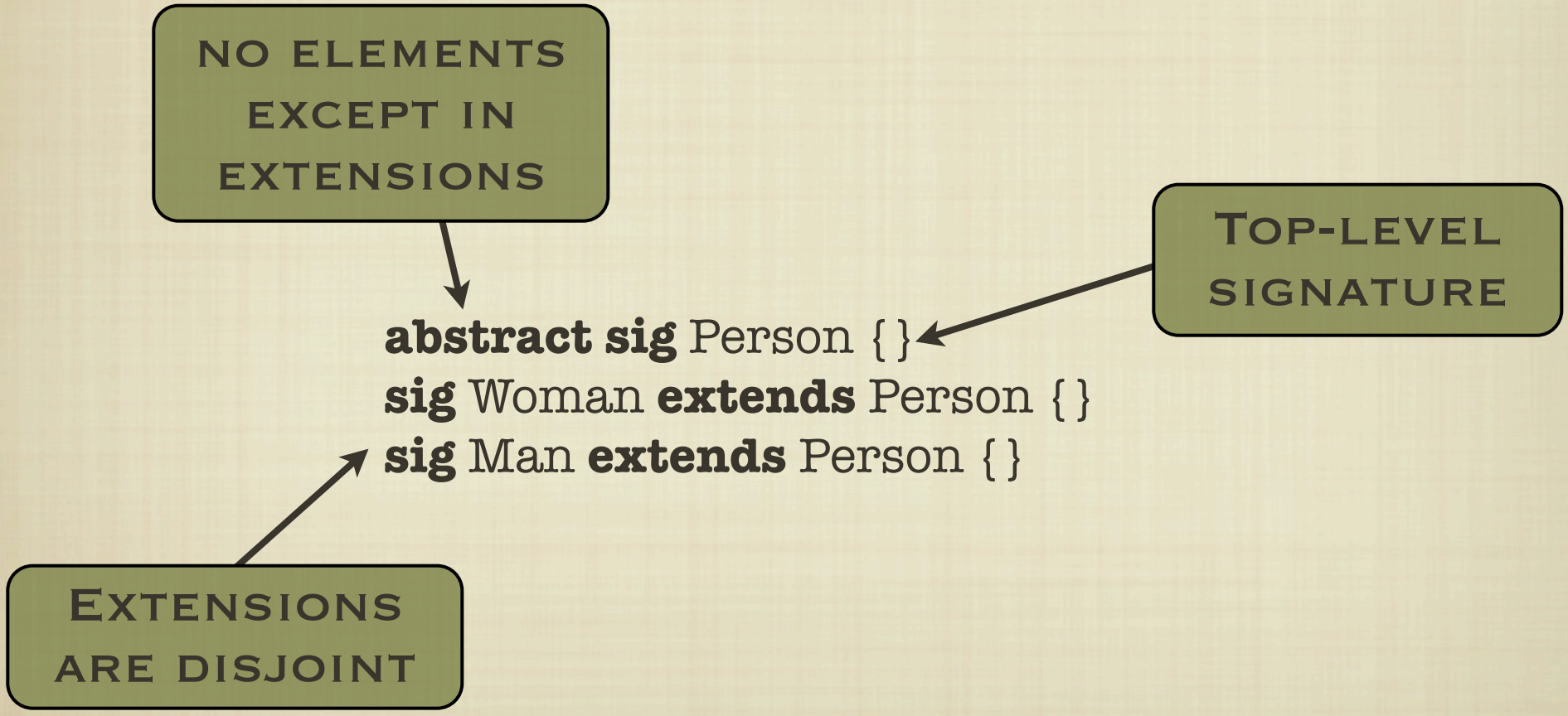
UML for Example



Signatures

Signatures

- Introduce sets of atoms



Subset Signatures: **in**

- Allow multiple classification hierarchies
- Examples:
  - **sig** Faculty **in** Person {}
    sig Staff **in** Person {}
    sig Employee **in** Faculty + Staff {}

Not necessarily disjoint

Employees aren't necessarily in both

Example

```
sig Tea {}
sig IndiaTea, ChinaTea extends Tea {}
sig Assam, Darjeeling extends IndiaTea {}
sig Keemun extends ChinaTea {}
sig EarlGrey in ChinaTea + IndiaTea {}

// Multiple inheritance
sig FlavoredTea in Tea {}
sig JasmineTea extends ChinaTea {}
fact { JasmineTea in FlavoredTea }
```

Constraining Sigs

- Can add multiplicity before signature:
  - m **sig** A {}
- Example:
  - **abstract sig** Color {}
    one sig Red, Yellow, Green
    extends Color {}

Creates an enumeration

Next Time

- Fields
- Types
- "Paragraphs"
  - Facts, Predicates, Functions, Assertions