**Lecture Week 10**

**Software Engineering**

**Formal Specification Example**

Using Ƶ – Logic (predict columns) and sets.

Ƶ – Invented in Oxford Progr. Lab

– Successfully used in many software projects.

t sentient – man() 🡪 likes(, football)

– man(x) 🡪 mortal()

Prolog – Progr. In Logic

Assert man (John)

* Mortal(john)

**Telephone directory for a department**

Given set [Person, Phone]

*Schema*

|  |  |
| --- | --- |
| **Tel Directory** | **Explanation** |
| Tel: Person 🡨🡪 phone | * Declarations * “🡨🡪” = Relation |
| Dept: F Person | * Declarations |
| Dom tel c dept | * Invariant(predicate always true) |

A = {1,2,3}

A = { { }, {1}, {2}, {3}, {12}, {13}, {23}, {1,2,3}}

Powerset of A

ϝA – Finite subsets of A

**e.g. suppose:**

tel = {Jim 🡪 1234, Sue 🡪 1234, Tim 🡪 1111, Tim 🡪 1112, Joe 🡪 2222}

dom tel = {Jim, Sue, Tim, Joe}

ran tel = {1234, 1111, 1112, 2222}

*ran = range operation*

*dom = domain operation*

**Operation**

Update – Changes state of telDir

Query – Lookup, no change of state

For any operation, have before and after states

State before telDir’

State after telDir

|  |
| --- |
| **TelDir ’** |
| Tel ‘ : Person 🡨🡪 Phone  Dept ‘ : IF Person |
| Dom tel ‘ c dept’ |

For bevity, use “syntactic sugar”

|  |
| --- |
| ∆ telDir |
| telDir  telDir |

|  |
| --- |
| addNum |
| ∆ telDir  N? : Phone  P? : Person |
| P? dept  Tel ‘ = tel u { p? 🡪 n? } |