The Grubstake Methodology
David A Gustafson

Creating a New Theory
Why should a theory devised for physical phenomenon be useful for software - an artificial phenomenon?

Empirical Relations
- One cludge that appears very often in the computer science literature is to define the empirical relation as that which makes the representation condition hold for the measure
- The truth is that we do not have centuries of consensus on the empirical relationships

Artificial
- Software has been described as an artificial science because we control the objects. In fact, the objects are artifacts - made by humans
- However, if there are constraints on these artifacts, it is usually in the operations that can change the artifacts

Transformations
- Consider the transformations that you can do to a syntactically correct program that maintains the correctness:
  - add a statement
  - add a condition
  - remove a statement

MOM
- Model - Order - Mapping
- the model is the abstraction
- the order is an ordering on the abstraction
  - can be partial, can be constructive, can be based on operations
- the mapping is from the model to the answer set
  - must preserve the order (I.e. rep condition)
Partial Order

- A partial order is a relationship such that
  - 1) reflexive
    - $a \leq_{po} a$
  - 2) antisymmetric
    - if $a \geq_{po} b$ and $b \leq_{po} a$ then $a = b$
  - 3) transitive
    - if $a \leq_{po} b$ and $b \leq_{po} c$ then $a \leq_{po} c$

- “Less than or equal” is a partial order

---

Classic Example

- McCabe’s Cyclomatic Number
- Abstraction - Control Flow Graph
- Partial Order - Containment (construct $a$ from $b$ by adding construct)
- mapping - E-N+2
- Answer Set - integers

---

Triangle Problem

```cpp
cin>>a>>b>>c;
type="scalene";
if (a==b||b==c|a==c) type="isosceles";
if (a==b&&b==c) type="equilateral";
if (a+b+c|b=a+c|c=a+b) type="not a triangle";
if (a<=0|b<=0|c<=0) type="bad inputs";
cout<<type;
```

---

Partial Order

- Originally, for all cfg $x,y$, $x \leq y$ if and only if all nodes and edges in $x$ are also in $y$.
- Thus adding an edge to $x$ made a new cfg $y$ such that $x \leq y$ but $y$ may not be a cfg for a valid program.
- Constructive definition - $x \leq y$ if there is a set of transforms (additive) such that the transforms are applied to $x$ produce $y$. 

**Representation Condition Proof**

- Can prove that for all cfg’s x and y, if x <= y then m(x) <= m(y)

**Halstead’s Vocabulary (eta)**

- Abstraction
  - Set of operators – S1
  - Set of operands – S2
- Operations
  - Add operator – O1
  - Add operand – O2

**Partial Order**

- Constructive
- Mapping to answer set
  - $\text{Card}(S1) + \text{Card}(S2)$

**Eta – proof of rep condition**

**What abstraction?**

- Bipartite graph
  - Attribute nodes
  - Method nodes
  - Arc if method uses/sets attribute

**LCOM**

Lack of Cohesion in Methods
What transformations?
- Add an attribute
- Add a statement with attribute ref
- Add a statement without attribute ref
- Add a function

Partial Order on Abs
- Containment?
- Related to transformations?

Mapping Abs to Ans
- What is the mapping from the abstraction to the answer set?

Rep Condition
- Does the mapping between the partial order on the abstraction and the answer set satisfy the representation condition?