Lecture 12: Algebraic Specifications

Kenneth M. Anderson

Foundations of Software Engineering

CSCI 5828 - Spring Semester, 2000

Today's Lecture

- Examine Algebraic Specifications
  - Compare Stack and Queue
  - Introduce Homework 4

 © Kenneth M. Anderson, 2000

Algebraic Specifications

- Algebras are Akin to Abstract Data Types
- Sets of Values
- Operations
- Many Formalisms
  - Larch, CCS, Lotos, …
  - RAISE can be used in an algebraic "style"

 © Kenneth M. Anderson, 2000

Terminology

- Homogeneous Algebra

  Single set and its operations
- Heterogeneous Algebra

  Multiple sets and their operations
- Signature

  Collection of sets in heterogeneous algebra
- Sort

  A set within an algebra

 © Kenneth M. Anderson, 2000

Terminology

- Syntax

  Signature plus operations with domains and ranges

- Semantics

  Equations involving operations; axioms

- Generators

  Operations that create instance of an algebra; inductive rules of inference

© Kenneth M. Anderson, 2000

Algebraic Specification of Stack

algebra StackOfItem
  imports Boolean;
  introduces
    sorts Stack, Item;
    operations
      Create: $\rightarrow$ Stack;
      IsEmpty: Stack $\rightarrow$ Boolean;
      Push: Stack $\times$ Item $\rightarrow$ Stack;
      Pop: Stack $\rightarrow$ Stack;
      Top: Stack $\rightarrow$ Item;
  constrains Create, IsEmpty, Push, Pop, Top so that
    Stack generated by [Create, Push]

© Kenneth M. Anderson, 2000

Algebraic Specification of Queue

algebra QueueOfItem
  imports Boolean;
  introduces
    sorts Queue, Item;
    operations
      Create: $\rightarrow$ Queue;
      IsEmpty: Queue $\rightarrow$ Boolean;
      Enqueue: Queue $\times$ Item $\rightarrow$ Queue;
      Dequeue: Queue $\rightarrow$ Queue;
      Front: Queue $\rightarrow$ Item;
  constrains Create, IsEmpty, Enqueue, Dequeue, Front so that
    Queue generated by [Create, Enqueue]

© Kenneth M. Anderson, 2000

Algebraic Specification of Pizza

algebra Nonsense
  imports Boolean;
  introduces
    sorts Pizza, Car;
    operations
      Cat: $\rightarrow$ Pizza;
      Horse: Pizza $\rightarrow$ Boolean;
      Dog: Pizza $\times$ Car $\rightarrow$ Pizza;
      Bird: Pizza $\rightarrow$ Pizza;
      Mouse: Pizza $\rightarrow$ Car;
  constrains Cat, Horse, Dog, Bird, Mouse so that
    Pizza generated by [Cat, Horse]

© Kenneth M. Anderson, 2000

Semantic Specification of Stack

```
for all [s: Stack; i: Item]
    IsEmpty(Create) = true;
    IsEmpty(Push(s,i)) = false;
    Pop(Create) = error;
    Top(Create) = error;
    Pop(Push(s,i)) = s;
    Top(Push(s,i)) = i;
end StackOfItem;
```


Semantic Specification of Queue

```
for all [q: Queue; i: Item]
    IsEmpty(Create) = true;
    IsEmpty(Enqueue(q,i)) = false;
    Dequeue(Create) = error;
    Front(Create) = error;
    Dequeue(Enqueue(q,i)) = if (IsEmpty(q))
                            then Create
                            else Enqueue(Dequeue(q),i);
    Front(Enqueue(q,i)) = if (IsEmpty(q))
                          then i
                          else Front(q);
end QueueOfItem;
```


Homework 4

- Give the semantics for an algebraic specification of a set of items
  - I give you the syntax
- Sets contain only one instance of a particular value
  - e.g. Adding {2} to {1, 2} gives {1, 2}
  - Adding {3} to {1, 2} gives {1, 2, 3}