Post Evaluation Procedure

- Evaluation packages returned
- Review period: Till 5 p.m., Fri., Nov. 7
  - Opportunity to upgrade
  - Make individual appointments
  - Do not wait (Module C Eval Workshop on Nov. 14)
  - *No review period for Module C or D*
- Otherwise, resubmit today

www.tcnj.edu/~komagata/grades.html

General Comments

- Comprehensive
  1. "Amida": Structure problem
     Diagram instance → Structure
  2. Crime Scene: Logic-structure problem
     Implications of "the smallest structure" With fewer objects?
  3. Ping-Pong: Logic-structure problem
     Use of %, etc. Definition of these?
- Mini Project
  - Describe your logic-structure connection
  - Cf. B7 Group Exercises, Group Exercises Today

CABG (Coronary Artery Bypass Graft)

Procedure

1. Opening the chest
2. Harvesting vessels (in the leg)
3. Transitioning to bypass
4. Attaching new vessels
5. Closing the chest

- Structures to represent information of this type
- Logic to specify such structures

Strings (Programming Languages)

Examples

- "pop", "eye", "popeye", "eyepop"
- "dad", "abba", "kayak"
- "abracadabra", "Ockham's razor"
- "a", "b", "c", ... , "A", "B", "C", ...
- "'" (apostrophe), " " (space), "" (nothing)

- Structures to represent information of this type
- Logic to specify such structures

Operational vs. Relational

- Operational
  - Single set
  - Operation/function: e.g., associative, existence of an identity element
- Relational
  - Single set
  - Relation: e.g., reflexive, antisymmetric, transitive
- Hybrid

CABG and Strings?

C1: Operational/Relational Structures

Today

Use the ideas for Comprehensive/ Mini Project review

- Understand the logic-structure connection and how to classify structures
  - Operational structures
  - Relational structures
- Take-home exercises
  - Reviews, Sets, FOL review

Section 1

String

- Alphabet (single char): $\Sigma = \{a, b, c, ..., z\}$
- String: *sequence* of characters (consisting of elements of Σ)
  - Notation for set of all the strings: S^*

Think of this as a single symbol
More details about this notation in a few weeks

- Null/empty string: a string of length zero
  - Notation: e (distinguish from '$\in$')

String as a Structure

- Concatenation operation '+'
  - Form a new string from two strings in a particular order, e.g., *pop* + *eye* = *popeye*
- **Strings** = $(\Sigma^*, +, \varepsilon)$
  - Σ^*: the set of all the strings
  - '+': $\Sigma^* \times \Sigma^* \to \Sigma^*$
  - '+' = $\{((a, b), ab),... ((\varepsilon, a), a),... ((a, \varepsilon), a),...\}$
    More precisely, '+' = $\{((x, y), z) \mid z = xy\}$
  - $\varepsilon: \to \Sigma^*$

schematic

symbol to represent an invisible element

Properties of Concatenation

Strings = $(\Sigma^*, +, \varepsilon)$

- '+' is closed. I.e., the result of concatenating two strings will always be a string.
- '+' is associative.
- There is an identity element, the empty string, represented by the special 0-ary function ε.
- '+' is *not* commutative.

Group Exercise 1

- Formally represent the 3 properties
  - '+' is closed. I.e., the result of "adding" two objects from a set A will also be in A.
  - '+' is associative. I.e., bracketing does not matter.
  - There is an identity element, represented as constant ε.
- Formally define the *specified* structure(s) $\mathbf{A} = (A, +, \varepsilon)$ including all the components

Omit $\in A$

Specification of String-like Structures

- Logical specification
  - Closed: $\forall x \forall y \exists z\ (x + y = z)$
  - Associative: $\forall x \forall y \forall z\ ((x + y) + z = x + (y + z))$
  - Identity: $\exists x \forall y\ ((x + y = y + x = y) \wedge (x = \varepsilon))$
- Structure: $\mathbf{A} = (A, +, \varepsilon)$
  - A is a set.
  - '+': $A \times A \to A$
  - $\varepsilon: \to A$

No further information available

Specify singe/multiple structure(s)?
Same as **String**?

Group Exercise 2

- The following structure **Nat** = (**N**, +, *0*) involving natural numbers **N** also satisfies the same 3 properties (except the use of *0* instead of ε).
- Analyze the logic-structure connection as in Group Exercise 1.
- What can you conclude about **Strings**, **A**, and **Nat**?

Specification of String-like Structures

- Logical specification
  - Closed: $\forall x\, \forall y\, \exists z\ (x + y = z)$
  - Associative: $\forall x\, \forall y\, \forall z\ ((x + y) + z = x + (y + z))$
  - Identity: $\exists x\, \forall y\ ((x + y = y + x = y) \wedge (x = 0))$
- Structure: **Nat** = (**N**, +, *0*)
  - **N** = the set of natural numbers
  - '+': $\mathbf{N} \times \mathbf{N} \rightarrow \mathbf{N}$
  - 0: $\rightarrow \mathbf{N}$

Interim Summary

- Structure **A** is the most general case.
- Structure **Strings** is a special case (an instance) of **A**.
  - Specialized to characters and concatenation
- Structure **Nat** is another special case of **A**.
  - Specialized to the set of natural numbers and addition
- All of these satisfy the same 3 logical statements.

What does '+' mean?

Group Exercise 3

- Analyze the logic-structure connection behind the "integer" data type of modern programming languages (e.g., Java, C++).
  - Find an additional property and formally represent it
  - Formally define structures **Int** = (*I*, +, *0*)
- Hints
  - $I = \{-2{,}147{,}483{,}648, ..., 0, ..., 2{,}147{,}483{,}647\}$, due to 32-bit storage, cf. **String** or **Nat**
  - E.g., $2{,}000{,}000{,}000 + 2{,}000{,}000{,}000 < 0$

Integer Data Type

- Additional property
  - Existence of inverses (because of the loop)
  - $\forall x\, \exists i\ (x + i = 0 \wedge i + x = 0)$
- **Int** = (**I**, +, *0*)
  - $\mathbf{I} = \{-2147483648, ..., 0, ..., 2147483647\}$
  - '+': $\mathbf{I} \times \mathbf{I} \rightarrow \mathbf{I}$
  - 0: $\rightarrow \mathbf{I}$
  - $0 = 0$

Is **Int** a special case of **A**?
Is **A** a special case of **Int**?

Clock

Clock = (*H*, +, *0*)

- $H = \{0, 1, 2, ..., 23\}$
- $+ : H \times H \rightarrow H$
- $0 : \rightarrow H$
- '+' is closed, associative, and commutative.
- Existence of identity element, 0
- Existence of inverses

Max

Max = (**R**, *max*)

- **R** = the set of real numbers
- *max*: **R** ´ **R** ® **R** [with the usual meaning]
- *max* is closed, associative, and commutative.
- There is *no* identity element. I.e., there is no smallest number.

Section Summary

- Operational structures
  - Single set
  - Operation(s): input(s) → output

Also called algebraic structures, algebras

- Classification of operational structures

Examples	Closed	Associative	Identity	Inverse	Names
Int	Y	Y	Y	Y	Groups
Strings	Y	Y	Y	N	Monoids
Max	Y	Y	N	N	Semigroups