Ordered Structures

Class 4
1.3 Ordered Structures

• a set is an unordered collection of distinct “things” called elements
  1. no element is repeated
  2. there is no ordering in the collection

• a bag relaxes the requirement of distinctness: bags can have duplicated elements, but is still unordered

• this section is about several related structures that relax the other condition, ordering

• this section is about ordered structures:
  • tuples
  • lists
  • strings
Random Access

- throughout this section the author uses the term random access
- this is a terrible phrase — it should be arbitrary access or better yet direct access
- it conveys the concept that elements can be directly accessed in any order, including a random order
- but in real life, data are almost never accessed randomly, they are accessed arbitrarily depending on the application
- “random” is way overused today, and as an informatics professional, you should know better
Tuple

• a tuple is a collection that has order, allows repetition, and has direct access via position
• tuples are denoted by parentheses
• \((a, b, c)\) is a 3-tuple
• () is the empty tuple
• \((x_0, \ldots, x_{n-1}) = (y_0, \ldots, y_{n-1})\) if \(x_i = y_i\) for all \(0 \leq i < n\)
  \LaTeX:\n  \((x_0, \ldots, x_{n-1}) = (y_0, \ldots, y_{n-1})\) if \(x_i = y_i\) for all \(0 \leq i < n\)$
• \((1, 3, 5) = (1, 3, 5)\)
  \((1, 3, 5) \neq (1, 5, 3)\)
List

- a list is a collection that has order, allows repetition, but does not allow direct access
- lists are denoted by angle brackets
- \( \langle a, b, a, c \rangle \) is a 4-element list
- \( \langle \rangle \) is the empty list
- equality of two lists similar to tuples
- \( \langle x_0, \ldots x_{n-1} \rangle = \langle y_0, \ldots y_{n-1} \rangle \) if \( x_i = y_i \) for \( 0 \leq i < n \)

\text{\LaTeX:}
\[
\langle x_0, \ldots x_{n-1} \rangle = \langle y_0, \ldots y_{n-1} \rangle \text{ if } x_i = y_i \text{ for } 0 \leq i < n
\]

- \( \langle 1, 3, 5 \rangle = \langle 1, 3, 5 \rangle \)
- \( \langle 1, 3, 5 \rangle \neq \langle 1, 5, 3 \rangle \)
String

• similar to tuple (ordered, duplicates allowed, direct access)
• but represented as juxtaposed elements drawn from some set of symbols called an alphabet
• if $A = \{a, b, c\}$ is an alphabet, then some strings over $A$ are:
  • $a$
  • $ba$
  • $ababa$
  • $ccaa$
  • $\Lambda$ (\LaTeX: \texttt{\textbackslash Lambda}) the empty string
Sets of Tuples

• given sets $A = \{1, 3, 5\}$ and $B = \{\text{Ann, Bob}\}$

• the Cartesian product $A \times B$ is a set of tuples

\[
A \times B = \{(x, y) \mid x \in A, y \in B\}
\]

(\text{\LaTeX}: \ A \times B = \{(x,y) \mid x \in A, y \in B\})

\[
A \times B = \{(1, \text{Ann}), (1, \text{Bob}),
(3, \text{Ann}), (3, \text{Bob}),
(5, \text{Ann}), (5, \text{Bob})\}
\]
Sets of Tuples

\[ A \times A = A^2 = \{(1, 1), (1, 3), (1, 5), (3, 1), (3, 3), (3, 5), (5, 1), (5, 3), (5, 5)\} \]

- \[ A^1 = \{(x) \mid x \in A\} \]
- \[ A^0 = \{()\} \]
List Access and Operations

- lists are special in that direct access does not exist
- instead, access is only at the head of the list
- head() is the operation that gives access to the first element
- given $L = \langle a, b, a, c \rangle$, then head($L$) = $a$
- the return type of head() is an element

- a second list operator is tail()
- given $L = \langle a, b, a, c \rangle$, then tail($L$) = $\langle b, a, c \rangle$

- a third list operator is cons()
- cons($a, \langle a, b, a \rangle$) = $\langle a, a, b, a \rangle$
List Operations

- the elements of a tuple can themselves be tuples
- the elements of a list can themselves be lists
- the elements of a string cannot themselves be strings, as strings are over specific alphabets
- given $L = \langle \langle a \rangle, b, \langle c, d \rangle \rangle$, find head($L$) and tail($L$)
List Operations

• the elements of a tuple can themselves be tuples
• the elements of a list can themselves be lists
• the elements of a string cannot themselves be strings, as strings are over specific alphabets

• given $L = \langle \langle a \rangle, b, \langle c, d \rangle \rangle$, find $\text{head}(L)$ and $\text{tail}(L)$

\[
\text{head}(L) = \langle a \rangle
\]

\[
\text{tail}(L) = \langle b, \langle c, d \rangle \rangle
\]
List Operations

• the elements of a tuple can themselves be tuples
• the elements of a list can themselves be lists
• the elements of a string cannot themselves be strings, as strings are over specific alphabets

• given $L = \langle\langle a\rangle, b, \langle c, d\rangle\rangle$, find head($L$) and tail($L$)

  $\text{head}(L) = \langle a\rangle$

  $\text{tail}(L) = \langle b, \langle c, d\rangle\rangle$

• $\text{tail}(\text{tail}(L)) =$
List Operations

• the elements of a tuple can themselves be tuples
• the elements of a list can themselves be lists
• the elements of a string cannot themselves be strings, as strings are over specific alphabets

• given $L = \langle \langle a \rangle, b, \langle c, d \rangle \rangle$, find $\text{head}(L)$ and $\text{tail}(L)$

  \[
  \text{head}(L) = \langle a \rangle
  \]

  \[
  \text{tail}(L) = \langle b, \langle c, d \rangle \rangle
  \]

• $\text{tail}(\text{tail}(L)) = \langle \langle c, d \rangle \rangle$
String Concatenation

- the concatenation of two strings is their juxtaposition, a new string
- the concatenation of $ab$ and $bab$ is $abbab$
- for any string $s$, $s\Lambda = \Lambda s = s$
- for any string $s$, $ss$ denotes $s$ concatenated with itself
- $s^n$ is $s$ concatenated with itself $n$ times
- $(ab)^3 = ababab$
- for any string $s$, $s^0 = \Lambda$
• a language is a set of strings over some alphabet
• if $A$ is an alphabet, then $A^*$ is the set of all possible strings over $A$
• thus $A^*$ is a language

• consider the set of strings

\[ \{ab^n a \mid n \in \mathbb{N}\} = \{aa, aba, abba, abbbba, \ldots \} \]

this is a language over $\{a, b\}$
Language Product

• we define the product operation for languages
• given languages (sets of strings) $L$ and $M$, then $LM$, the product of $L$ and $M$, is a new language

$$LM = \{st \mid s \in L, t \in M\}$$

• if $L = \{a, bb\}$ and $M = \{ab, bc\}$, then

$$LM = \{aab, abc, bbab, bbbc\}$$

• if

$$\{\Lambda, a, b\} L = \{\Lambda, a, b, aa, ba, aba, bba\}$$

solve for $L$
Language Product

- we define the product operation for languages
- given languages (sets of strings) $L$ and $M$, then $LM$, the product of $L$ and $M$, is a new language

$$LM = \{st \mid s \in L, t \in M\}$$

- if $L = \{a, bb\}$ and $M = \{ab, bc\}$, then

$$LM = \{aab, abc, bbab, bbbc\}$$

- if

$$\{\Lambda, a, b\}L = \{\Lambda, a, b, aa, ba, aba, bba\}$$

solve for $L$

$$L = \{\Lambda, a, ba\}$$
Language Product and Closure

- given $L = \{a, ab\}$, then $LL = L^2 = \{aa, aab, aba, abab\}$
- and $L^3 = LL^2 = \{aaa, aaab, aaba, aabab, abaa, abaab, ababa, ababab\}$
- $L^1 = L$ and $L^0 = \{\Lambda\}$
- given language $L$, $L^*$ is the closure of $L$
- it is the set of all possible concatenations of strings in $L$

$$L^* = L^0 \cup L^1 \cup L^2 \cup \ldots \cup L^n \cup \ldots$$
Positive Closure

• the positive closure of $L$, denoted $L^+$ is

$$L^+ = L^1 \cup L^2 \cup \ldots \cup L^n \cup \ldots$$

and so

$$L^* = L^+ \cup \{\Lambda\}$$

• use language product form to describe all floating point decimal numbers

let $D = \{.\}$ and $N = \{d \mid d$ is a digit\}$

then $L = N^+ DN^+$
Counting Cartesian Products

- for sets of tuples, lists, and strings built up from a Cartesian product, the basis of counting is the product rule:

\[ |A \times B| = |A||B| \]

(provided there are no duplicates in the set)

- so, if \( A = \{a, b\} \) and \( B = \{x, y, z\} \), then

\[ A \times B = \{(ax), (ay), (az), (bx), (by), (bz)\} \]

whose cardinality is 6
Counting Example

how many strings of length 6 over the alphabet $A = \{a, b, c, d\}$
begin with $a$ or $c$ and contain at least one $b$?

the number of length-6 strings that start with $a$ or $c$ is

$$|\{a, c\} \times A^5| = 2(4^5) = 2048$$

but not all of them have a $b$; how many do not have a $b$?

$$|\{a, c\} \times \{a, c, d\}^5| = 2(3^5) = 486$$

so the answer to the original question is

$$2048 - 486 = 1562$$

how many length-6 strings have a $b$?
Counting Example

how many strings of length 6 over the alphabet $A = \{a, b, c, d\}$ begin with $a$ or $c$ and contain at least one $b$?

the number of length-6 strings that start with $a$ or $c$ is

$$|\{a, c\} \times A^5| = 2(4^5) = 2048$$

but not all of them have a $b$; how many do not have a $b$?

$$|\{a, c\} \times \{a, c, d\}^5| = 2(3^5) = 486$$

so the answer to the original question is

$$2048 - 486 = 1562$$

how many length-6 strings have a $b$?

$$4096 - 729 = 3367$$
Relations

- we’ll have much to say about relations in this course
- here we’ll just introduce them
- fundamentally, a relation in a set of tuples
- specifically, a subset of some Cartesian product
Less-Than

• let $A = \{0, 1, 2, 3\}$; then

\[ A \times A = \{(0, 0), (0, 1), (0, 2), (0, 3), \\
(1, 0), (1, 1), (1, 2), (1, 3), \\
(2, 0), (2, 1), (2, 2), (2, 3), \\
(3, 0), (3, 1), (3, 2), (3, 3)\} \]

• now we can define the relation less-than, $<$, as a subset of $A \times A$

\[ < = \{(0, 1), (0, 2), (0, 3), \\
(1, 2), (1, 3), \\
(2, 3)\} \]
Alternate Notations

- less-than is a **binary** relation
- we can write
  - set notation: \((2, 3) \in <\)
  - function notation: \(< (2, 3)\)
  - infix notation: \(2 < 3\)
page 54 and example 10 on page 55 is a discussion about relational databases
we will not do anything with that material