Section 1.3 Ordered Structures

Tuples

Have order and can have repetitions.

(6,7,6) is a 3-tuple

() is the empty tuple

A 2-tuple is called a "pair" and a 3-tuple is called a "triple".

We write $(x_1, \ldots, x_n) = (y_1, \ldots, y_n)$ to mean $x_i = y_i$ for $1 \leq i \leq n$.

Cartesian Product:

$A \times B = \{ (x,y) \mid x \in A \text{ and } y \in B \}$

This definition extends naturally:

$A \times B \times C = \{ (x,y,z) \mid x \in A \text{ and } y \in B \text{ and } z \in C \}$

Notation:

$A^0 = \{ () \}$

$A^1 = \{ (x) \mid x \in A \}$

$A^2 = \{ (x_1,x_2) \mid x_2 \in A \text{ and } x_2 \in A \}$

$A^n = \{ (x_1, \ldots, x_n) \mid x_i \in A \}$

In-Class Quiz:

Does $(A \times B) \times C = A \times (B \times C)$?

Lists

Like tuples but there is no random access.

Example:

<a,b,c,b> is a list with 4 elements

<> is the empty list.

List operations: **head**, **tail**, **cons**

head (<a,b,c,b>) = a

tail (<a,b,c,b>) = <b,c,b>

cons (e, <a,b,c,b>) = <e,a,b,c,b>

The set of lists whose elements are in A is denoted by **lists**(A).

Lists can contain lists:

< 3 , <a,b,c> , 4 , <7,8> , e , <> , g >

In-class Quiz:

For L = <<a>,b,<c,d>>

Find head(L)

Find tail(L)

Strings

Like lists.

All elements come from an **alphabet**.

The elements are juxtaposed.

Example: alphabet is A={a,b}.

Some strings: **a**, **b**, **aa**, **ab**, **ba**, **bb**, **aaa**, **bbb**, …

The **empty string** is denoted by Λ (lambda).

Concatenation of two strings is their juxtaposition.

The concatenation of **ab** and **bab** is **abbab**.

This is true of any string s:

$s \Lambda = \Lambda s = s$

If s is a string, s^n denotes the concatenation of s with itself n times.

$s^0 = \Lambda$.

Example:

$(\mathbf{ab})^3 = \mathbf{ababab}$

Languages

Given an alphabet A, a **language** is a set of strings over A.
Notation:
If A is an alphabet, then the set of *all* strings over A is denoted A^*.
Some languages over A are:
Ø, $\{\Lambda\}$, A, A^*
Example:
Let alphabet be {a,b}
$\{ab^na \mid n \in \mathbb{N}\} = \{aa, aba, abba, abbba, \ldots\}$

Language Operations:
Let L and M be two languages.
The **product** of L and M, denoted LM, is
$LM = \{ st \mid s \in L \text{ and } t \in M \}$
Example:
Let L = { **a**, **b** } and M = { **cc**, **ee**}. Then…
LM = { **acc**, **aee**, **bcc**, **bee** }
ML = { **cca**, **ccb**, **eea**, **eeb** }

In-class Quiz:

What are the products $L\emptyset$ and $L\{\Lambda\}$?

In-class Quiz:

Solve for L in the equation

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

Notation:

$L^0 = \{\Lambda\}$

$L^1 = L$

$L^2 = LL$

$L^n = \{ s_1 s_2 \ldots s_n \mid s_i \in L \}$

The closure L^* is the set of all possible concatenations of strings in L.

$L^* = L^0 \cup L^1 \cup \ldots \cup L^n \cup \ldots$

In-class quiz:

What are $\{\Lambda\}^*$ and $\emptyset^*$?

Example:

Examine the structure of an arbitrary string $x \in L^*(ML)^*$.

Approach: Use the definitions to write x in terms of strings in L and M.

Since $x \in L^*(ML)^*$, it follows that $x = uv$, where $u \in L^*$ and $v \in (ML)^*$.
Since $u \in L^*$, either $u = \Lambda$ or $u = s_1 \ldots s_n$ for some n where $s_i \in L$.
Since $v \in (ML)^*$, either $v = \Lambda$ or $v = r_1 t_1 \ldots r_k t_k$ for some n where
$r_i \in M$ and $t_i \in L$.
So x must have one of four forms:

Λ

$s_1 \ldots s_n$

$r_1 t_1 \ldots r_k t_k$

$s_1 \ldots s_n r_1 t_1 \ldots r_k t_k$

Relations

A relation is a set of tuples.
If R is a relation and $(x_1, \ldots, x_n) \in R$, we write $R(x_1, \ldots, x_n)$.
We can usually represent a relation as a subset of some cartesian product.
Example:

Let $R = \{(0,0), (1,1), (4,2), (9,3), \ldots, (k^2,k), \ldots\}$
$= \{ (k^2,k) \mid k \in \mathbb{N} \}$

We might call R the "is square of" relation on $\mathbb{N}$.

Notice that $R \subseteq \mathbb{N}\times\mathbb{N}$.

Notation:

If R is binary, we can use **infix** to represent pairs in R.

For example, from the previous example, we have

$(9,3) \in R$

So we can write:

R(9,3)
9 R 3
9 is-square-of 3

Relational Databases

A relational database is a relation where the indexes of a tuple have associated names, called **attributes**.

Example:

Let Students = { (x,y,z) | x is a Name, y is a Major, and z is Credits)

Name	Major	Credits
JohnSmith	cs	70
FredBrown	math	85
JackGreen	math	120
SueJones	cs	130

Who are the students majoring in CS?

{ x | (x, "cs", z) ∈ Students }

Note: we need a way to tell values apart from variables: (x,cs,z)?

How many math majors are upper division students?

| { x | (x, "math", z) ∈ Students and z ≥ 90 } |

What is the major of JohnSmith?

{ y | ("JohnSmith", y, z) ∈ Students }

What is the Math departments database of names and credits?

{ (x,y) | (x, "math", z) ∈ Students }

Counting Tuples (or strings or lists)

Product Rules:

$|A \times B| = |A|\ |B|$

$|A^n| = |A|^n$

Example: If A = {a,b} and B={1,2,3} then

$A \times B = \{ (a,1), (a,2), (a,3), (b,1), (b,2), (b,3) \}$

So $|A \times B| = |A|\ |B| = 2 \times 3 = 6$

Example:

Count the number of strings of length 8 over A = {a,b,c} that begin with either **a** or **c** and have at least one **b**.

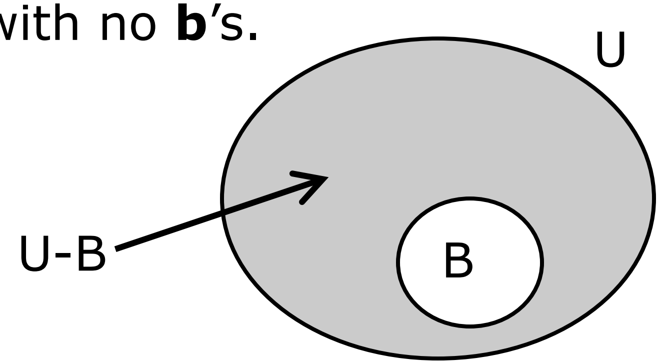
Solution: **Divide and conquer!**

Split the problem up into easier problems and combine the results.

Let U be the universe = the set of strings over A of length 8 that begin with either **a** or **c**.

Let B be the subset of U consisting of strings with no **b**'s.

The set we want to count is then U-B.



Calculate the cardinality of U-B.

$|U\text{-}B| = |U| - |U \cap B|$

$= |U| - |B|$ since B is a subset of U

What is the cardinality of U?

$U = \{a,c\} \times A^7$

$|\{a,c\} \times A^7| = |\{a,c\}| \times |A^7| = |\{a,c\}| \times |A|^7 = 2(3)^7$

What is the cardinality of B, the set of strings not containing **b**?

$|\{a,c\}^8| = |\{a,c\}|^8 = 2^8$

So the answer is:

$|U\text{-}B| = |U| - |U \cap B|$

$= |U| - |B| = 2(3)^7 - 2^8 = 4118$