

Section 1.3 Ordered Structures

Tuples: Have order and can have repetitions. For example, $(6, 7, 6)$ is a 3-tuple and $()$ is the empty tuple. We write $(x_1, \dots, x_n) = (y_1, \dots, y_n)$ to mean $x_i = y_i$ for $1 \leq i \leq n$.

Cartesian Product: $A \times B = \{(x, y) \mid x \in A \text{ and } y \in B\}$. The definition extends to more than two sets. For example, $A \times B \times C = \{(x, y, z) \mid x \in A, y \in B, z \in C\}$.

- Notation: $A^0 = \{()\}$, $A^1 = \{(x) \mid x \in A\}$, and in general, $A^n = \{(x_1, \dots, x_n) \mid x_i \in A\}$.

Quiz (1 minute). Does $(A \times B) \times C = A \times (B \times C)$?

Lists: Are like tuples but there is no random access. For example, $[a, b, a, c]$ is a list with 4 elements and $[]$ is the empty list.

- The operations on lists are *head*, *tail*, and *cons*. For example,

$$\text{head}([a, b, a, c]) = a, \quad \text{tail}([a, b, a, c]) = [b, a, c] \quad \text{cons}(a, [b, a, c]) = [a, b, a, c]$$

- The set of lists whose elements are in A is denoted by $\text{lists}(A)$.

Quiz (1 minute). For $L = [a, b, c, d]$ find $\text{head}(L)$ and $\text{tail}(L)$.

Strings: Are like lists, but are represented as juxtaposed elements from a given alphabet.

For example, if $A = \{a, b\}$, then some strings over A are: $a, b, aa, ab, ba, bb, aaa, bbb$.

- The *empty string* is denoted by ϵ .
- The *concatenation* of two strings is their juxtaposition. For example, the concatenation of ab and bab is $abbab$.
- For any string s we have $s\epsilon = \epsilon s = s$.
- If s is a string, s^n denotes the concatenation of s with itself n times. Also $s^0 = \epsilon$. For example, $(ab)^3 = ababab$.

Languages

A *language* is a set of strings, usually taken over some alphabet.

Notation: If A is an alphabet, then the set of all strings over A is denoted by A^* .

Example. Some languages over A are: \emptyset , $\{\epsilon\}$, A , and A^* .

Example. $\{ab^n a \mid n \in \mathbf{N}\} = \{aa, aba, abba, abbba, \dots\}$ is a language over $\{a, b\}$.

Language Operations

Let L and M be languages. The *product* of L and M , denoted LM , is the language

$$LM = \{st \mid s \in L \text{ and } t \in M\}.$$

- Notation: $L^0 = \{\epsilon\}$; and $L^n = \{s_1 \dots s_n \mid s_i \in L\}$.

Quiz (1 minute). What are the products $L\emptyset$ and $L\{\epsilon\}$?

Quiz (1 minute). Solve for L in the equation $\{\epsilon, a, b\}L = \{\epsilon, a, b, aa, ba, aba, bba\}$.

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

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

Quiz (1 minute). What are $\{\epsilon\}^*$ and \emptyset^* ?

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

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

1. Since $x \in L^*(ML)^*$, it follows that $x = uv$ where $u \in L^*$ and $v \in (ML)^*$.
2. Since $u \in L^*$, either $u = \epsilon$ or $u = s_1 \dots s_n$ for some n where $s_i \in L$.
3. Since $v \in (ML)^*$, either $v = \epsilon$ or $v = r_1 t_1 \dots r_k t_k$ for some k where $r_i \in M$ and $t_i \in L$.

So x has one of four forms: ϵ , $s_1 \dots s_n$, $r_1 t_1 \dots r_k t_k$, or $s_1 \dots s_n r_1 t_1 \dots r_k t_k$. 2

Relations.

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

Example. Let $R = \{(0, 0), (1, 1), (4, 2), (9, 3), \dots, (n^2, n), \dots\} = \{(n^2, n) \mid n \in \mathbf{N}\}$. We might call R the “is square of” relation on \mathbf{N} . Notice also that $R \subseteq \mathbf{N} \times \mathbf{N}$.

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 also write

$$R(9, 3) \quad \text{or} \quad 9 R 3 \quad \text{or} \quad 9 \text{ is square of } 3.$$

Relational Databases

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

Example. Let $\text{Students} = \{(x, y, z) \mid x \text{ is a Name, } y \text{ is a Major, and } z \text{ is Credits}\}$.

Who are the people majoring in CS?

$$\{x \mid (x, \text{CS}, z) \in \text{Students, for some } z\}.$$

Note: We need “for some z ” to indicate that z is a variable.

How many math majors are upper division students?

$$\mid \{x \mid (x, \text{math}, z) \in \text{Students and } z \geq 90\} \mid.$$

What is the major of AbeLincoln?

$$\{y \mid (\text{AbeLincoln}, y, z) \in \text{Students, for some } z\}.$$

What is the history department database of names and their credits?

$$\{(x, z) \mid (x, \text{history}, z) \in \text{Students}\}.$$

Counting Tuples (or strings or lists)

Product Rule: $|A \times B| = |A| |B|$ and $|A^n| = |A|^n$.

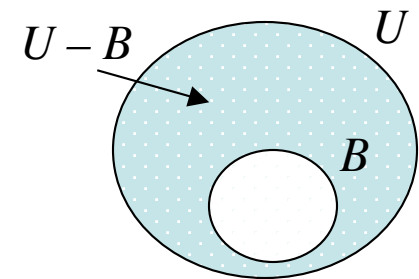
Example. If $A = \{a, b\}$ and $B = \{1, 2, 3\}$, then

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

So $|A \times B| = 6 = (2)(3) = |A| |B|$.

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 .

A Solution: Split the problem up into easier problems and combine the results (*divide and conquer*). Let U be the universe consisting of the 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. Then the set of strings to count is $U - B$, as pictured.



It is easy to calculate the cardinality of $U - B$:

$$|U - B| = |U| - |U \cap B| = |U| - |B| \text{ (since } B \text{ is a subset of } U)$$

It is also easy to count U because it has the same cardinality as the set $\{a, c\} \times A^7$, which is

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

It is also easy to count B because it has the same cardinality as the set $\{a, c\}^8$, which is

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

So we have the answer:

$$|U - B| = |U| - |U \cap B| = |U| - |B| = (2)3^7 - 2^8, \text{ which is } 4118.$$