Basics

"It is always best to start at the beginning"

-- Glynda, the good witch of the North

Outline

- Sets
- Relations
- Languages

Sets

- A <u>set</u> is a collection of elements.
  - Finite set – has finite number of elements
    - A = { 1, 3, 5, 6, 9 }
  - Infinite Set – has infinite number of elements
    - B = { x | x is an odd integer }
  - Null (Empty) set – has no elements.
    - $\emptyset$

Sets

- Membership
  - If an element x is a member of a set A, we write:
    - $x \in A$.
  - If an element x is not a member of a set A, we write:
    - $x \notin A$
- Subsets:
  - A is a subset of B if all elements of A are also in B.
  - $A \subseteq B$ if $x \in A$ then $x \in B$
  - A = B is the same as saying $A \subseteq B$ and $B \subseteq A$.
- Power Set
  - 2^{A} = Set of all subsets of A

Set Operations

- Union:
  - $A \cup B$ = set consisting of all elements in either A or B or both.
- Intersection
  - $A \cap B$ = set of elements that are in both A and B
  - If $A \cap B = \emptyset$, A and B are disjoint.
- Difference
  - A – B = set of all elements of A that are not elements of B
- Complement (wrt a universal set U)
  - A' = All elements in U that are not in A
  - U - A

Set Operations

- A = { 1, 3, 5, 6, 9 }
- B = { x | x is an odd integer }

- $A \cup B$ = { x | x = 6 or x is an odd integer}
- $A \cap B$ = { 1, 3, 5, 9 }
- A – B = { 6 }
- B' = { x | x is an even integer }
  - With respect to the Universal set of Positive integers.

Cartesian Product of Sets

- A x B = set of <u>ordered pairs</u> (a,b) such that
  - $a \in A$ and $b \in B$.
- A x B x C = set of ordered triplets (a, b, c) such that
  - $a \in A$ and $b \in B$ and $c \in C$
- In general: $A_1 \times A_2 \times \ldots \times A_n$ = the set of all n-tuples $(a_1, a_2, \ldots, a_n)$ such that
  - $a_i \in A_i$ for all i.

Relations on sets

- Means to relate or associate a member of one set with a member of another:
  - $R : A \rightarrow B$
  - Relation is simply a subset of A x B
    - If $a \in A$ is related to $b \in B$ then $(a, b) \in R$
    - Can also be written aRb
    - Or R(a) contains b

Functions

- Functions are restricted relations
  - $f: A \rightarrow B$

  - Every element of A is associated with <u>exactly 1</u> element of B

Functions

- A function, $f: A \rightarrow B$ is:
  - <u>Onto (Surjective)</u> if every element of B is related to at least one element of A
  - <u>One-to-one (Injective)</u> if every element of B is related to at most one element of A.
  - A <u>bijection</u> if it is both one-to-one and onto
    - Every element in B is related to one and only one element of A.

Functions

- $f : A \rightarrow B$
  - A and B can themselves be Cartesian Products
  - $f: A \times B \rightarrow C$
    - Elements of f are ((a,b), c) where $a \in A$, $b \in B$, $c \in C$
    - f (a,b) = c

Equivalence Relations

- If R is a relation on A (R: A → A), R is an equivalence relation if:
  - R is reflexive – (a, a) ∈ R for all a in A
  - R is symmetric – if (a,b) ∈ R then (b,a) ∈ R
  - R is transitive – if (a, b) ∈ R and (b,c) ∈ R then (a,c) ∈ R

Equivalence Relations

- An equivalence relation on A (R: A → A) partitions the elements of A into disjoint equivalence classes.
  - For each class, elements in the class are related only to all elements in the same partition.
  - Example:
    - $\equiv_4$: N → N : x $\equiv_4$ y if (x mod 4) == (y mod 4)
    - [0] = { 0, 4, 8, 12, …}
    - [1] = { 1, 5, 9, 13, …}
    - [2] = { 2, 7, 10, 15, …}
    - [3] = { 3, 8, 11, 16, …}

Functions and Relations

- Questions?
- Next: languages

What is a Language?

- A language is a set of strings made of of symbols from a given alphabet.
- An alphabet is a finite set of symbols (usually denoted by Σ)
  - Examples of alphabets:
    - {0, 1}
    - {α, β, χ, δ, ε, φ, γ, η}
    - {a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t , u, v, w, x, y, z}
    - {a}

What is a string?

- A *string over Σ* is a finite sequence (possibly empty) of elements of Σ.
- Λ denotes the *null string*, the string with no symbols.
  - Example strings over {a, b}
    - Λ, a, aa, bb, aba, abbba
  - NOT strings over {a, b}
    - aaaa…., abca

The length of a string

- The length of a string *x*, denoted |*x*|, is the number of symbols in the string
  - Example:
    - |abbab| = 5
    - |a| = 1
    - |bbbbbbb| = 7
    - | Λ| = 0

Strings and languages

- For any alphabet Σ, the set of all strings over Σ is denoted as Σ^*.
- A language over Σ is a subset of Σ^*
  - Example
    - $\{a,b\}^* = \{\Lambda, a, b, aa, bb, ab, ba, aaa, bbb, baa, \ldots\}$
  - Example Languages over {a,b}
    - $\{\Lambda, a, b, aa, bb\}$ $\varnothing$
    - $\{x \in \{a,b\}^* \mid |x| = 8\}$ $\{x \in \{a,b\}^* \mid |x| \text{ is odd}\}$
    - $\{x \in \{a,b\}^* \mid n_a(x) = n_b(x)\}$ $\{\Lambda\}$
    - $\{x \in \{a,b\}^* \mid n_a(x) = 2 \text{ and x starts with b}\}$

Concatenation of String

- For $x, y \in \Sigma^*$
  - xy is the concatenation of x and y.
    - x = aba, y = bbb, xy=ababbb
    - For all x, $\Lambda x = x\Lambda = x$
  - x^i for an integer i, indicates concatenation of x, i times
    - x = aba, x^3 = abaabaaba
    - For all x, $x^0 = \Lambda$

Some string related definitions

- x is a substring of y if there exists $w,z \in \Sigma^*$ (possibly Λ) such that $y = wxz$.
  - *car* is a substring of *carnage*, *descartes*, *vicar*, *car*, but not a substring of charity.
- x is a suffix of y if there exists $w \in \Sigma^*$ such that $y = wx$.
- x is a prefix of y if there exists $z \in \Sigma^*$ such that $y = xz$.

Operations on Languages

- Since languages are simply sets of strings, regular set operations can be applied:
  - For languages L_1 and L_2 over Σ^*
    - $L_1 \cup L_2$ = all strings in L_1 or L_2
    - $L_1 \cap L_2$ = all strings in both L_1 and L_2
    - $L_1 - L_2$ = strings in L_1 that are not in L_2
    - $L' = \Sigma^* - L$

Concatenation of Languages

- If L_1 and L_2 are languages over Σ^*
  - $L_1L_2 = \{xy \mid x \in L_1 \text{ and } y \in L_2\}$
  - Example:
    - L_1 = {hope, fear}
    - L_2 = {less, fully}
    - L_1L_2 = {hopeless, hopefully, fearless, fearfully}

Concatenation of Languages

- If L is a language over Σ^*
  - L^k is the set of strings formed by concatenating elements of L, k times.
  - Example:
    - L = {aa, bb}
    - L^3 = {aaaaaa, aaaabb, aabbaa, aabbbb, bbbbbb, bbbbaa, bbaabb, bbaaaa}
    - $L^0 = \{\Lambda\}$

Kleene Star Operation

- The set of strings that can be obtained by concatenating any number of elements of a language L is called the Kleene Star, L^*

$$L^* = \bigcup_{i=0}^{\infty} L^i = L^0 \cup L^1 \cup L^2 \cup L^3 \cup L^4 \ldots$$

✓ Note that since, L^* contains L^0, Λ is an element of L^*

Kleene Star Operation

- The set of strings that can be obtained by concatenating one or more elements of a language L is denoted L^+

$$L^+ = \bigcup_{i=1}^{\infty} L^i = L^1 \cup L^2 \cup L^3 \cup L^4 \ldots$$

Specifying Languages

- How do we specify languages?
  - If language is finite, you can list all of its strings.
    - L = {a, aa, aba, aca}
  - Using basic Language operations
    - $L = \{aa, ab\}^* \cup \{b\}\{bb\}^*$
  - Descriptive:
    - $L = \{x \mid n_a(x) = n_b(x)\}$

Specifying Languages

- Next time we will define how to specify languages recursively
- In future classes, we will describe how to specify languages by defining a mechanism for generating the language
- Any questions?