Kleene Closure

Definition: Given an alphabet ∑, we define a language in which any string of letters from ∑ is a word, even the null string **Λ**. We call this language the **closure** of the alphabet ∑, and denote this language by ∑*.

Examples: If ∑ = { x } then ∑* = { **Λ**, x, xx, xxx, … }

If ∑ = { 0, 1 } then ∑* = { **Λ**, 0, 1, 00, 01, 10, 11, 000, 001, … }

If ∑ = { a, b, c } then ∑* = { **Λ**, a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, … }

Lexicographic order

Notice that we listed the words in a language in size order (i.e., words of shortest length first), and then listed all the words of the same length alphabetically.

This ordering is called **lexicographic** order, which we will usually follow.

The star in the closure notation is known as the **Kleene star**.

We can think of the Kleene star as an **operation** that makes, out of an alphabet, an *infinite* language (i.e., *infinitely many* words, each of *finite* length).

Kleene Closure

Let us now generalize the use of the Kleene star oprator to sets of words, not just sets of alphabet letters.

Definition: If S is a set of words, then S* is the set of all finite strings formed by concatenating words from S, where any word may be used as often as we like, and where the null string **Λ** is also included.

Example: If S = { aa, b } then

S* = { **Λ** plus any word composed of factors of aa and b }, or

S* = { Λ plus any strings of a's and b's in which the a's occur in **even** clumps }, or

S* = { Λ, b, aa, bb, aab, baa, bbb, aaaa, aabb, baab, bbaa, bbbb, aaaab, aabaa, aabbb, baaaa, baabb, bbaab, bbbaa, bbbbb, … }

Note that the string aabaaab is not in S* because it has a clump of a's of length 3.

Example: Let S = { a, ab }. Then

S* = { **Λ** plus any word composed of factors of a and ab }, or

S* = { **Λ** plus all strings of a's and b's except those that start with b and those that contain a double b }, or

S* = { **Λ**, a, aa, ab, aaa, aab, aba, aaaa, aaab, abaa, abab, aaaaa, aaaab, aaaba, aabaa, aabab, abaaa, abaab, ababa, … }

Note that for each word in S*, every b must have an a immediately to its left, so the double b, that is bb, is not possible; neither any string starting with b.

How to prove a certain word is in the closure language S*

We must show how it can be written as a concatenation of words from the base set S.

In the previous example, to show that abaab is in S*, we can factor it as follows:

abaa
b =
(ab)(
a)(ab
)

These three factors are all in the set S, therefore their concatenation is in S*.

Note that the parentheses, ( ), are used for the sole purpose of demarcating the ends of factors.

Observe that if the alphabet has no letters, then its closure is the language with the null string as its only word; that is

if ∑ = ø (the empty set),
then ∑* = { **Λ** }

Also, observe that if the set S has the null string as its only word, then the closure language S* also has the null string as its only word; that is

if S = { **Λ** }, then S* = { **Λ** }
because **ΛΛ** = **Λ**.

Hence, the Kleene closure always produces an infinite language unless the underlying set is one of the two cases above.

Kleene Closure of different sets

The Kleene closure of two different sets can end up being the same language.

~~**Example**: Consider two sets of words S = { a , b, ab } and T = { a, b, bb }~~

Then, both S* and T* are languages of all strings of a's and b's since any string of a's and b's can be factored into syllables of (a) or (b), both of which are in S and T.

Positive Closure

If we wish to modify the concept of closure to refer only the concatenation of

some (**not zero**) strings from a set S, we use the notation + instead of *.

This "plus operation" is called **positive closure**.

Example: if $\sum = \{ x \}$ then $\sum^+ = \{ x, xx, xxx, \ldots \}$ Observe that:

1. If S is a language that **does not** contain **Λ**, then S^+ is the language S* without the null word **Λ**.
2. If S is a language that **does** contain **Λ**, then $S^+ = S^*$
3. Likewise, if ∑ is an alphabet, then $\sum^+$ is ∑* without the word **Λ**.

S?**

What happens if we apply the closure operator twice?

– We start with a set of words S and form its closure S*

– We then start with the set S* and try to form its closure, which we denote as

(S

*
)
*

o
r

S
*
*

Theorem 1:

For any set S of strings, we have S* = S**

Before we prove the theorem, recall from Set Theory that

– A = B if A is a subset of B **and** B is a subset of A

– A is a subset of B if for all x in A, x is also in B

Proof of Theorem 1:

Let us first prove that S** is a subset of S*:

Every word in S** is made up of factors from S*. Every factor from S* is made up of factors from S. Hence, every word from S** is made up of factors from S. Therefore, every word in S** is also a word in S*. This implies that S** is a subset of S*.

Let us now prove that S* is a subset of S**:

In general, it is true that for any set A, we have A is a subset of A*, because in A* we can choose as a word any factor from A. So if we consider A to be our set S* then S* is a subset of S**

Together, these two inclusions prove that S* = S**.

Example:

Defining language of EVEN

Step 1: 2 is in **EVEN**.

Step 2: If x is in **EVEN** then x+2 and x-2 are also in **EVEN**.

Step 3: No strings except those constructed in above, are allowed to be in **EVEN**.

Example:

Defining the language factorial Step 1: As 0!=1, so 1 is in **factorial**.

Step 2: n!=n*(n-1)! is
in **factorial**.

Step 3: No strings except those constructed in above, are allowed to be in **factorial**.

Defining the language PALINDROME, defined over Σ = {a,b} Step 1: a and b are in **PALINDROME**

Step 2: if x is palindrome, then s(x)Rev(s) and xx will also be palindrome, where s belongs to Σ^*

Step 3: No strings except those constructed in above, are allowed to be in palindrome

Defining the language $\{a^nb^n\}$, n=1,2,3,… , of strings defined over Σ={a,b}

Step 1: ab is in $\{\mathbf{a^nb^n}\}$

Step 2: if x is in $\{\mathbf{a^nb^n}\}$, then axb is in $\{\mathbf{a^nb^n}\}$

Step 3: No strings except those constructed in above, are allowed to be in $\{\mathbf{a^nb^n}\}$

Defining the language L, of strings ending in a , defined over Σ={a,b}

Step 1: a is in L

Step 2: if x is in L then s(x) is also in **L,** where s belongs to Σ^*

Step 3: No strings except those constructed in above, are allowed to be in **L**

Defining the language L, of strings beginning and ending in same letters , defined over Σ={a, b}

Step 1: a and b are in L

Step 2: (a)s(a) and (b)s(b) are also in **L,** where s belongs to Σ^*

Step 3: No strings except those constructed in above, are allowed to be in **L**

Defining the language L, of strings containing aa or bb , defined over Σ={a, b}

Step 1: aa and bb are in L

Step 2: s(aa)s and s(bb)s are also in **L,** where s belongs to Σ^*

Step 3: No strings except those constructed in above, are allowed to be in **L**

Defining the language L, of strings containing exactly aa, defined over Σ={a, b}

Step 1: aa is in L

Step 2: s(aa)s is also in **L,** where s belongs to b^*

Step 3: No strings except those constructed in above, are allowed to be in **L**