CA320 What I Need

Regular Languages

Finite Automata

Deterministic - DFA

Each possible input determines the resulting state q’ uniquely. Every input causes a state change and new state is determined by the input. Moreover, the automaton can change state only after reading input. Only one state at finish based upon input.

Non-Deterministic - NFA

Some inputs may allow a choice of resulting states. Some may cause automaton to choke. No new state corresponding to that input. Several possible end states

Pumping Lemma

Let L be element of theta and M has 5 tuples such that L = L(M)

If M has n states then for every x is an element of L satisfying |x| => n there are three strings u v w such that x = uvw and

|uv| =< n

|v| > 0

For all i >= 0, UViW is element of L

Context-Free Languages

Pumping Lemma for Context-Free Languages

If L is a context free language there is an integer n such that for all u is an element of L with |u| >= n, u = vwxyz such that

|wy| > 0

|wxy| <= n

For all => 0, VWmXYmZ is an element of L

Computational Models

Church Turing Thesis states that Turing Machine is a general model of computing that is any algorithmic procedure can be carried out by any computer can be carried out by Turing Machine. It was formulated but not proven because of what defines an algorithmic procedure

Church-Turing thesis

1. The nature of the model makes it seem likely that al the crucial step to a computation can be carried out by Turing Machine
2. All various enhancements to Turing Machine model, though they make the model more efficient, have not increased the computational power of turing machine
3. Theoretical Models have been proposed but are equivalent to Turing Machine
4. Since conception of Turing Machine model no one has suggested ant type of computation that is considered an algorithmic procedure unsolvable by Turing Machine

Undecidability And the Halting Problem

The language Atm = {M,w|M is a turing machine and M accepts w} is undecidable

Reducability

To show undecidability we can reduce the language to a language L2 whose decidability is knnown. The language is mapping reducible to language L2 written L1 <= L2

If L1<= L2 and L2 is decidable then L1 is decidable

If L1<= L2 andL2 is undecidable then L1 is undecidable

Complexity

Complexity Measures

Static Measures - Structure

Dynamic Measures - algorithm inputs

TIME using Turing Machine as the model of computation, let TIME = {TIMEi} where TIMEi denotes the number of steps executed by computation Ti(x).

If Ti(x) does not halt then TIMEi(x) is considered undefined

SPACE must be defined carefully so that SPACEi(x) is undefined whenever Ti(x) does not halt.

1. The number of steps in the shortest accepting computation path of NTi(x), if one exists and all paths halt
2. The number of steps in the shortest computation path of NTi(x) if all paths halt and reject
3. Undefined Otherwise

NSPACE is defined similarly

Reducibility Lemma

Let c1 and c2 be two classes of languages such that C1 is element of C2 amd C1 os c;psed with respect tp a reducibility =< R. Then any language L that is C2 complete with respect to =< R does not belong to C1

Polynomial Time reducibility

Let L1 and L2 be two languages such that L1<L2 and L2 is elem of P