Generalized Transition Graph (GTG)

Since any regular language has an associated NFA and hence a transition graph, all we need to do is to find a regular expression capable of generating the labels of all the walks from q_0 to any final state.

A GTG is a transition graph whose edges are labeled with regular expressions; otherwise it is same as the usual transition graph.

The label of any walk from the initial state to a final state is the concatenation of several regular expressions, hence itself a regular expression.

The strings denoted by such regular expressions are a subset of the language accepted by the generalized transition graph, with the full language being the union of ~~the~~ all such generated subsets.

If a GTG, after conversion from an NFA, has some edges missing, we put them in and label them with ϕ.

A complete GTG with $|V|$ vertices has exactly $|V|^2$ edges.

Procedure NFA-to-rex

1. Start with an NFA with states $q_0, q_1, \ldots, q_n$, and a single final state, distinct from its initial state.
2. Convert the NFA into a complete generalized transition graph. Let r_{ij} stand for the label of the edge from q_i to q_j.
3. If the GTG has only two states, with q_i as its initial state and q_j its final state, its associated regular expression is $r = r_{ii}^* r_{ij} (r_{jj} + r_{ji} r_{ii}^* r_{ij})^*$.

4. If the GTG has three states, with initial state q_i, final state q_j, and third state q_k, introduce new edges labeled

$$r_{pq} + r_{pk} r_{kk}^{*} r_{kq}$$

for $p = i, j$, $q = i, j$. When this is done, remove q_k and its associated edges.

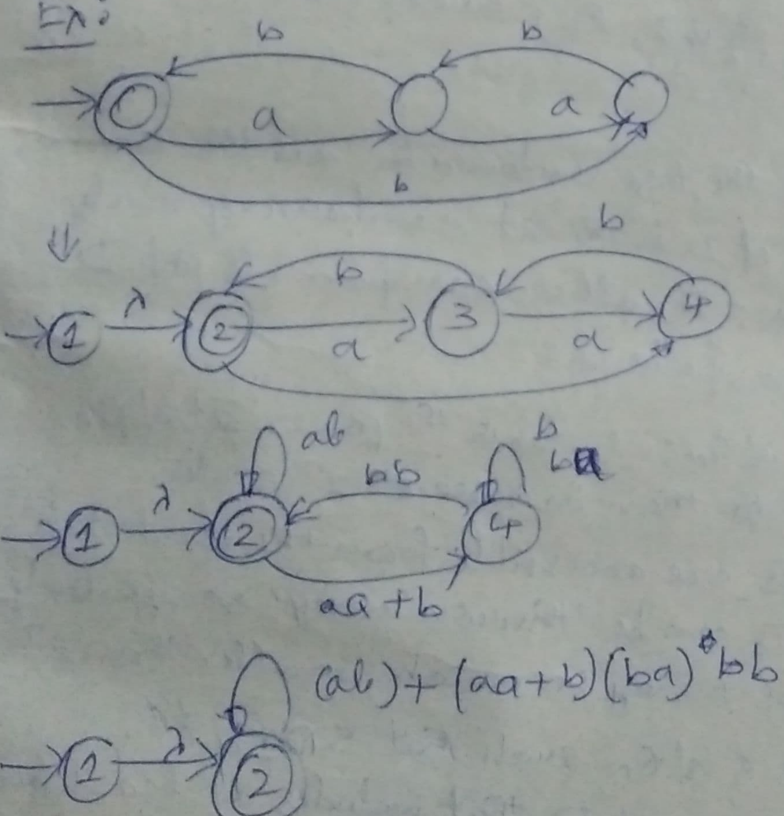
5. If the GTG has four or more states, pick a state q_k to be removed. Apply rule 4 for all pairs of states (q_i, q_j), $i \neq k$, $j \neq k$. At each step apply the simplifying rules,

$$r + \phi = r,$$
$$r\phi = \phi,$$
$$\phi^{*} = \lambda$$

wherever possible. When this is done, remove state q_k.

6. Repeat steps 3 to 5 until the correct regular expression is obtained.

Ex:



$$\therefore\ r = \left((ab) + (aa+b)(ba)^{*} bb\right)^{*}$$

EQUIVALENCE OF DETERMINISTIC AND NONDETERMINISTIC FINITE AUTOMATA

DFA $\Rightarrow 2^n$ states

smallest NFA for the same language can have n states only.

The proof that DFA's can do whatever NFA's can do involves an important "construction" called the subset construction because it involves constructing all subsets of the set of states of the NFA.

In general, many proofs about automata involve constructing one automaton from another.

It is important for us to observe the subset construction as an example of how one formally describe one automaton in terms of states and transitions of another, without knowing the specifics of the latter automaton.

The subset construction starts from an NFA $N = (Q_N, \Sigma, \delta_N, q_0, F_N)$. Its goal is the description of a DFA $D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$ such that

$L(D) = L(N)$.

- Input alphabets of the two automata are the same, and the start state of D is the set containing only the start state of N. The other components of D are constructed as follows:

→ Q_D is the set of subsets of Q_N; i.e., Q_D is the power set of Q_N. Note that if Q_N has n states, then Q_D have 2^n states. Often, not all these states are accessible from the start state of Q_D. Inaccessible states can be 'thrown away' so effectively, the number of states of D may be much smaller than 2^n.

→ F_D is the set of subsets S of Q_N such that $S \cap F_N \neq \phi$. That is F_D is all sets of N's states that includes at least one accepting state of N.

→ For each set $S \subseteq Q_N$ and for each input symbol a, in Σ,

$$\delta_D(S, a) = \bigcup_{p \text{ in } S} \delta_N(p, a)$$

To compute $\delta_D(S, a)$ we look at all states p in S, see what states N goes to from p on input a, and take the union of ~~the~~ all these states.

<u>Theorem</u>

If $D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$ is the DFA constructed from NFA $N = (Q_N, \Sigma, \delta_N, q_0, F_N)$ by the subset construction then $L(D) = L(N)$.

<u>Proof.</u> What we actually prove first, by induction on $|w|$, is that $\hat{\delta}_D(\{q_0\}, w) = \hat{\delta}_N(q_0, w)$

Notice that each of the $\hat{\delta}$ functions returns a set of states from Q_N, but $\hat{\delta}_D$ interprets this set as one of the states of Q_D (which is the power set of Q_N), while $\hat{\delta}_N$ interprets this set as a subset of Q_N.

<u>BASIS:</u> Let $|w| = 0$; that is $w = \epsilon$. By the basis definitions of $\hat{\delta}$ for DFA's and NFA's, both $\hat{\delta}_D(\{q_0\}, \epsilon)$ and $\hat{\delta}_N(q_0, \epsilon)$ are $\{q_0\}$.

<u>INDUCTION:</u> Let w be of length $n+1$, and assume the statement for length n. Break w up as $w = xa$, where a is the final symbol of w. By the inductive hypothesis, $\hat{\delta}_D(\{q_0\}, x) = \hat{\delta}_N(q_0, x)$. Let both these sets of N's states be $\{p_1, p_2, \dots, p_k\}$.

The inductive part of the definition of $\hat{\delta}$ for NFA's tells us

$$\hat{\delta}_N(q_0, w) = \bigcup_{i=1}^{k} \delta_N(p_i, a) \quad \dots (1)$$

The subset construction, on the other hand, tells us that

$$\delta_D(\{p_1, p_2, \dots, p_k\}, a) = \bigcup_{i=1}^{k} \delta_N(p_i, a) \quad \dots (2)$$

Let us use (2) and the fact that $\hat{\delta}_D(\{q_0\}, x) = \{p_1, p_2, \dots, p_k\}$ in the inductive part of the definition of $\hat{\delta}$ for DFA's:

$$\hat{\delta}_D(\{q_0\}, w) = \delta_D(\hat{\delta}_D(\{q_0\}, x), a)$$
$$= \delta_D(\{p_1, p_2, \ldots, p_k\}, a)$$
$$= \bigcup_{i=1}^{k} \delta_N(p_i, a) \quad \cdots \text{(3)}$$

Thus (2) & (3) demonstrate that $\hat{\delta}_D(\{q_0\}, w) = \hat{\delta}_N(q_0, w)$. When we observe that D and N both accept w if and only if $\hat{\delta}_D(\{q_0\}, w)$ or $\hat{\delta}_N(q_0, w)$, respectively, contain a state in F_N, we have a complete proof that $L(D) = L(N)$.

Theorem

A language L is accepted by same DFA if and only if L is accepted by same NFA.

Proof. If part $\Rightarrow$ as previous

(Only-if) This part is easy; we have only to convert a DFA into an identical NFA. Put intuitively, if we have the transition diagram for a DFA, we can also interpret it as the transition diagram of an NFA, which happens to have exactly one choice of transition in any situation. More formally, let $D = (Q, \Sigma, \delta_D, q_0, F)$ be a DFA. Define $N = (Q, \Sigma, \delta_N, q_0, F)$ to be the equivalent NFA, where δ_N is defined by the rule:

- If $\delta_D(q, a) = p$, then $\delta_N(q, a) = \{p\}$.

It is then easy to show by induction on $|w|$, that if $\hat{\delta}_D(q_0, w) = p$ then $\hat{\delta}_N = (q_0, w) = \{p\}$.

As a consequence, w is accepted by D if and only if it is accepted by N; i.e., $L(D) = L(N)$.

Th

A language L is accepted by some ϵ-NFA if and only if L is accepted by some DFA.

PROOF. (If) Suppose $L = L(D)$ for some DFA. Turn D into an ϵ-DFA E by adding transitions $\delta(q, \epsilon) = \emptyset$ for all states q of D. Technically, we must also convert the transitions of D on input symbols, e.g., $\delta_D(q, a) = p$ into an NFA transition to the set containing only p, that is $\delta_E(q, a) = \{p\}$. Thus, the transitions of E and D are the same, but E explicitly states that there are no transitions out of any state on ϵ.

(Only-if) Let $E = (Q_E, \Sigma, \delta_E, q_0, F_E)$ be an ϵ-NFA. Apply the modified subset construction to produce the DFA. $D = (Q_D, \Sigma, \delta_D, q_D, F_D)$.

We need to show that $L(D) = L(E)$, and we do so by showing that the extended transition functions of E and D are the same. Formally, we show $\hat{\delta}_E(q_0, w) = \hat{\delta}_D(q_D, w)$ by induction on the length of w.

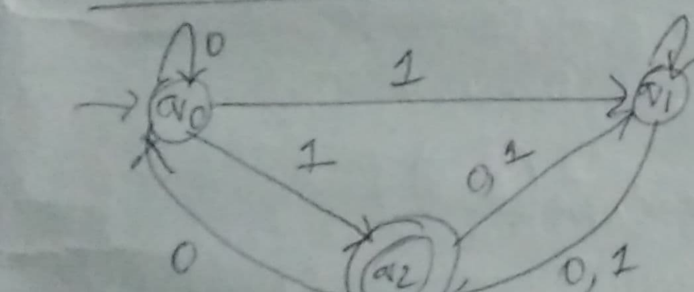
BASIS. If $|w| = 0$, then $w = \epsilon$. We know $\hat{\delta}_E(q_0, \epsilon) = ECLOSE(q_0)$. We also know that $q_D = ECLOSE(q_0)$, because that is how the start state of D is defined. Finally, for a DFA, we know that $\hat{\delta}(p, \epsilon) = p$ for any state p, so in particular, $\hat{\delta}_D(q_D, \epsilon) = ECLOSE(q_0)$. We thus have proved that $\hat{\delta}_E(q_0, \epsilon) = \hat{\delta}_D(q_D, \epsilon)$.

INDUCTION: Suppose $w = xa$, where a is the final symbol of w, and assume that the statement holds for x. That is, $\hat{\delta}_E(q_0, x) = \hat{\delta}_D(q_D, x)$. Let both these sets of states be $\{p_1, p_2, \dots, p_k\}$.
By the definition of $\hat{\delta}$ for ϵ-NFA's, we compute $\hat{\delta}_E(q_0, w)$ by:

1. Let $\{r_1, r_2, \dots, r_m\}$ be $\bigcup_{i=1}^{k} \delta_E(p_i, a)$.
2. Then $\hat{\delta}_E(q_0, w) = \bigcup_{j=1}^{m} ECLOSE(r_j)$.

If we examine the construction of DFA D in the modified subset construction, we see that $\delta_D(\{p_1, p_2, \dots, p_k\}, a)$ is constructed by the same two steps (1) & (2). Thus, $\hat{\delta}_D(q_D, w)$, which is $\delta_D(\{p_1, p_2, \dots, p_k\}, a)$ is the same set as $\hat{\delta}_E(q_0, w)$.
So, $\hat{\delta}_E(q_0, w) = \hat{\delta}_D(q_D, w)$ → Hence, proved.

NFA to DFA using subset construction



$(q_0, 0\} = \{q_0\} = A$

$(q_0, 1) = \{q_1, q_2\} = B$

$(\{q_1, q_2\}, 0) = \{q_0, q_1, q_2\} = C$

$(\{q_1, q_2\}, 1) = \{q_1, q_2\}$

$(\{q_0, q_1, q_2\}, 0) = \{q_0, q_1, q_2\}$

$(\{q_0, q_1, q_2\}, 1) = \{q_1, q_2\}$