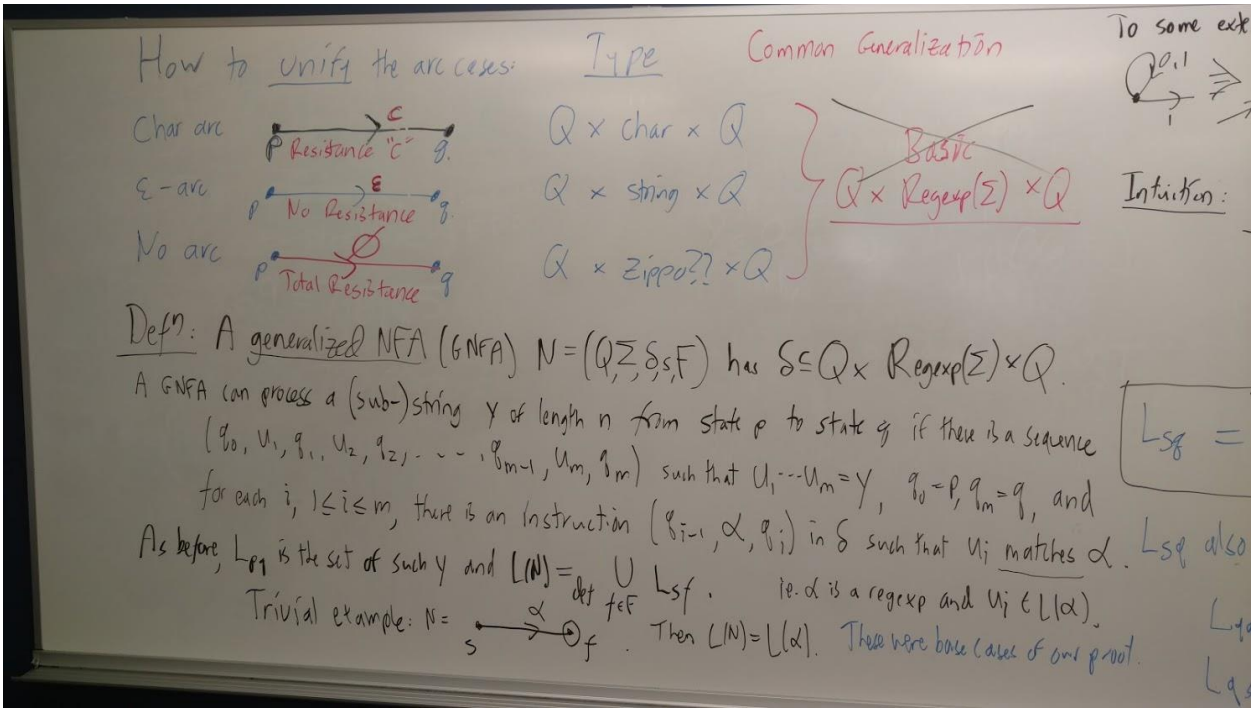
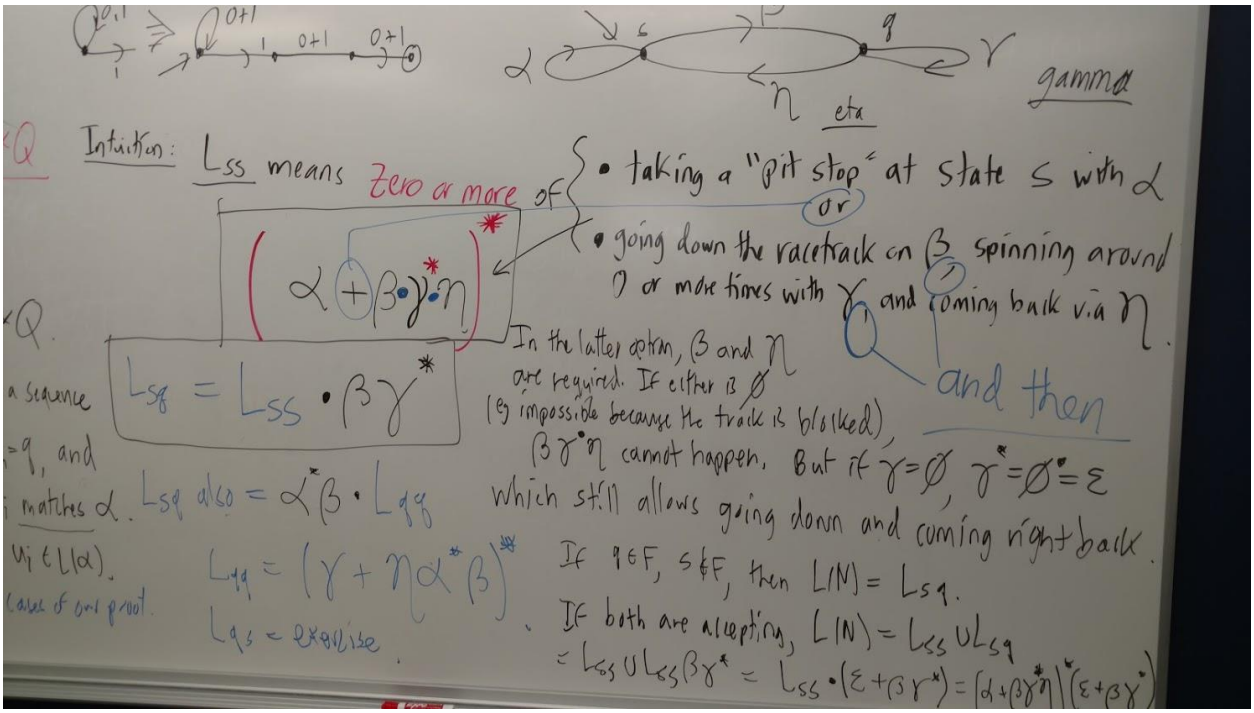
CSE596 Lecture Mon. Sep. 9

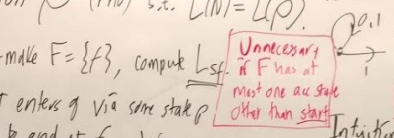




Theorem: Given any GNFA N, we can compute a regular expression ρ (rho) s.t. $L(N) = L(\rho)$.

Proof: Add a new final state f with ε-arcs from all old final states, re-make $F = \{f\}$, compute L_{sf}. Now all states $q \neq s$ in N are nonaccepting states. Hence any processing that enters q via some state p must eventually leave via some state r ($r = p$ allowed) in order eventually to end at f. We can:

Unnecessary if F has at most one acc state other than start



$L(N) = (\ldots$

Intuition: L_{ss} …

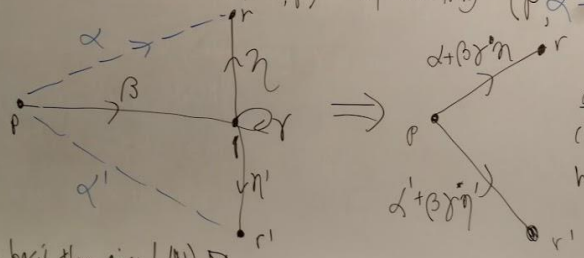
$L_{sq} = L_{ss}$ …

$-_{sq}$ also $= \alpha$ …

$L_{qq} = (\ldots$

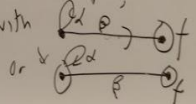
$L_{qs} = \ldots$

bypass the in-edge (p, β, q) and the out-edge (q, η, r) along with (q, γ, q) by creating $(p, \alpha + \beta\gamma^*\eta, r)$



When we have bypassed all (in-from q, out-to-r) combinations, we can delete q.

When we have deleted all $\eta \neq s$, we are left with



The basis then gives $L(N)$. ⊠