Definire le espressioni regolare e la nozione di implicazione fra espressioni regolari
Un'espressione regolare è una stringa costituita da caratteri dell'alfabeto terminale, dall'insieme vuoto, dalla stringa vuota e dagli operatori di unione, star e concatenamento.

Definire per casi in modo formale le espressioni regolari
Il linguaggio regolare si basa sulle espressioni regolare.
- espressione regolare = ϕ (insieme vuoto)
- espressione regolare = Ɛ (stringa vuota)
- espressione regolare = a (con a ∈ Σ, cioè qualsiasi simbolo che appartiene all'alfabeto)
- espressione regolare = e_1 U e_2 (unione di due espressioni regolari)
- espressione regolare = e_1* e_2* (star di due espressioni regolari)
- espressione regolare = $e_1 \cdot e_2$ (concatenamento di due espressioni regolari)

Definire formalmente l'automa a stati finiti <u>deterministico</u> equivalente ad un automa <u>non deterministico</u> (senza epsilon mosse)
L'**AUTOMA A STATI FINITI DETERMINISTICO**, quindi con le Ɛ mosse, è definito da 5 elementi:
<Q, Σ, ∂, q_0, F>.
Q, rappresenta l'insieme degli stati
Σ, rappresenta l'alfabeto
∂, rappresenta la funzione di transizione, cioè cosa succede ad un certo stato con un certo input. **∂ = Q x Σ -> Q**, significa che uno stato con un certo input può andare solo ed esclusivamente in un altro stato.
q_0, rappresenta lo stato iniziale
F, è un sottoinsieme di Q, rappresenta l'insieme degli stati finiti

L'**AUTOMA A STATI FINITI NON DETERMINISTICO**, quindi senza le Ɛ mosse, è definito da 5 elementi:
<Q, Σ, ∂, q_0, F>.
Q, rappresenta l'insieme degli stati
Σ, rappresenta l'alfabeto
∂, rappresenta la funzione di transizione, cioè cosa succede ad un certo stato con un certo input. **∂ = Q x Σ -> 2^Q**, significa che uno stato con un certo input può andare in altri stati.
q_0, rappresenta lo stato iniziale
F, è un sottoinsieme di Q, rappresenta l'insieme degli stati finiti

Definire le regole che permettono di costruire in modo automatico a partire da un automa a stati finiti che riconosce il linguaggio <u>complemento</u>
L'**AUTOMA COMPLEMENTO** di un automa a stati finiti deterministico, è definito da 5 elementi:
<Q', Σ', ∂', q_0', F'>.
Q' = Q U {P}, cioè uguale a Q dell'automa a stati finiti con l'aggiunta dello stato P, detto stato pozzo
Σ', rimane lo stesso di quello dell'automa a stati finiti
∂', viene suddivisa in casi: (per ogni q ∈ Q e per ogni a ∈ Σ)

- se $\partial(q, a) = k$ allora $\partial'(q, a) = k$, cioè ogni arco dell'automa a stati finiti ci sarà anche nell'automa complemento
- se $\partial(q, a) = /$ allora $\partial'(q, a) = P$, cioè se nell'automa a stati finiti non c'era l'arco con un simbolo dell'alfabeto, allora in quello complemento andrà nello stato P
- $\partial'(p, a) = P$, cioè si crea una arco che va da P a P con tutti i simboli dell'alfabeto

q_0' = q_0, cioè rimane sempre lo stesso stato iniziale dell'automa a stati finiti
F' = (Q - F) U {P}, cioè tutti gli stati non finali dell'automa a stati finiti e lo stato P

Definire formalmente l'automa a pila

L'**AUTOMA A PILA** è definito da 7 elementi:
$<Q, \Sigma, T, \partial, q_0, Z_0, F>$.
Q, rappresenta l'insieme degli stati
Σ, rappresenta l'alfabeto dell'input
T, rappresenta l'alfabeto della pila, cioè i simboli che si possono mettere o togliere dalla pila
∂, rappresenta la funzione di transizione, cioè cosa succede ad un certo stato con un certo input.
q_0, rappresenta lo stato iniziale
Z_0, rappresenta la pila vuota ed è un simbolo della pila
F, è un sottoinsieme di Q, rappresenta l'insieme degli stati finiti

Definire la nozione di "configurazione" per un automa a pila

La **CONFIGURAZIONE** è descritta così: $(q, y, \gamma) \in Q \times \Sigma^* \times T^*$, dove q è lo stato, y è la stringa in input che dovrà essere consumata e la γ è la stringa che dovrà essere consumata dalla pila.
La configurazione iniziale deve specificare tutti e 3 gli elementi: (q_0, A, Z_0).

Definire la nozione di "mossa" per un automa a pila

Nell'automa a pila, data una configurazione viene definita **MOSSA** il passaggio da una configurazione ad un'altra tramite la funzione di transizione.
Ci sono 2 casi: $(\gamma, x \in T^*)$

- $(q, aY, \gamma x) \rightarrow (p, Y, \gamma\pi)$ se $(p, \pi) \in \partial(q, a, x)$, cioè si consuma un elemento dall'input e si fa una push sulla pila
- $(q, Y, \gamma x) \rightarrow (p, \gamma\pi)$ se $(p, \pi) \in \partial(q, \varepsilon, x)$

Definire la nozione di "stato" per un automa a pila

cccc

Definire l'accettazione per un input negli automi a pila

L'**ACCETTAZIONE** di un input in un automa a pila avviene per:

- **stato finale**, cioè quando si consuma tutta l'input e si arriva in uno stato finale
  $(q_0, A, Z_0) \rightarrow (q, \varepsilon, \gamma)$ se $q \in F$
- **pila vuota**, cioè quando si consuma tutto l'input e alla fine la pila è vuota, cioè non contiene nessun simbolo, nemmeno Z_0
  $(q_0, A, Z_0) \rightarrow (q, \varepsilon, \varepsilon)$ se $q \in Q$

Definire formalmente le regole che permettono di costruire in modo automatico un automa a pila non deterministico data una grammatica context free

- si utilizza un'accettazione a pila vuota, cioè F è un insieme vuoto, $<Q, \Sigma, T, \partial, q_0, Z_0, \emptyset>$
- $Q = \{q_0\}$, esiste solo lo stato iniziale
- $\Sigma = \Sigma$, l'alfabeto dell'input è sempre lo stesso
- $Z_0 = Z_0$, rimane invariato il simbolo della pila vuota
- $T = \{ Z_0\} \cup \Sigma \cup V$, cioè sulla pila ci sono tanti simboli quanti sono i simboli dell'alfabeto e dei non terminali
- vengono gestite le regole di produzione, cioè per ogni $A \to \alpha$:
    - se α = terminale allora $\alpha = a\beta \to \partial(q_0, a, A) = (q_0, \beta^R)$
    - se α = non terminale allora $\alpha = X\beta \to \partial(q_0, \varepsilon, A) = (q_0, \beta^R X)$
- vengono gestito ogni simbolo dell'alfabeto, cioè per ogni $A \in \Sigma \to \partial(q_0, a, a) = (q_0, \varepsilon)$
- vengono gestite le regole della funziona di transizione
    - inizializzazione $\to \partial(q_0, \varepsilon, Z_0) = (q_0, \swarrow S)$
    - terminazione $\to \partial(q_0, \swarrow, \swarrow) = (q_0, \varepsilon)$