[A]**1.44** Let B and C be languages over $\Sigma = \{0,1\}$. Define

$$B \overset{1}{\leftarrow} C = \{w \in B \mid \text{ for some } y \in C, \text{ strings } w \text{ and } y \text{ contain equal numbers of 1s}\}.$$

Show that the class of regular languages is closed under the $\overset{1}{\leftarrow}$ operation.

$\{B, C\} =$ $C =$ DFA $\to$ ε-NFA

- RIMPIAZZATO CON ε OGNI TRANSIZIONI

00 1100 110000

DFA B DFA C

L FINALE $\to$ L INIZIALE

① SE L DI C = REGOLARE $\to \exists$ DFA C $(Q, \Sigma, q_0, \delta, F)$

② CONVERTO C IN ER $\to$ NFA $\exists$ NFA N $(Q, \Sigma, q_0, \delta, F)$

- $F' = F$
- $q_0' = q_0$

$\to \delta(q_{i_i}, 1) = \delta(r_i, 1) \cup q_i$, $i \in \{1 \dots n\}$

OPPURE $\delta(q_i, 1) = \delta(q_i', 1) = \delta(q_i + 1)$

SE $w \in \overset{1}{\leftarrow}$, $q_0 = q_0'$

$S_0 \xrightarrow{w_1} S_1 \xrightarrow{w_2} 1 \xrightarrow{1} \dots$

$S_i \to 1 \xrightarrow{y} 1 \to \dots S_M$

$S_0 \dots S_{M-1}$ TUTTI STATI NON FINALI

COMPUTAZIONE ACCETTATA

1.38 An ***all*-NFA** M is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ that accepts $x \in \Sigma^*$ if *every* possible state that M could be in after reading input x is a state from F. Note, in contrast, that an ordinary NFA accepts a string if *some* state among these possible states is an accept state. Prove that all-NFAs recognize the class of regular languages.

$\exists$! NFA $\subseteq$ ALL-NFA (⇒) OVVIO

POTREBBE (?) AVERE LO STATO ACCETTANTE

ALL-NFA RICONOSCE L. REGOLARI

Direzione 1: Ogni linguaggio regolare è riconosciuto da un all-NFA

(⇒)

Direzione 2: Ogni linguaggio riconosciuto da un all-NFA è regolare

$\exists$ q_{ACC} $\cup$ ECLOSE (q_i, a) → ALGORITMO

↓

$\exists$ FORSE UNO STATO ACCETTANTE

```
for each q ∈ S do
    S' ← S' ∪ δ_n(q,a)
end for
S' ← Eclose(S')
Q_D ← Q_D ∪ {S'}

if S' ⊆ F_n then
    F_D ← F_D ∪ {S'}
end if
```

(⇒)

$\exists$ NFA, ALLORA $\exists$ ALL-NFA |

q_m $\cup$ ECLOSE (NFA)

$\exists$ STATO ACCETTANTE

(⇐) $\exists$ ALL-NFA, $\exists$ NFA CON STATO ACCETTANTE (TRIVIALE / OVVIO)

8. (Bonus Question.) If A is a language, let $A_{\frac{1}{2}}$ be the set of all first halves of strings in A, so that

$$A_{\frac{1}{2}} = \{x \mid \text{ for some } y, |x| = |y| \text{ and } xy \in A\}.$$

Show that if A is regular, then so is $A_{\frac{1}{2}}$.

- **Transizione:** δ'((p,q), a) = {(δ(p,a), r) | δ(r,b) = q per qualche b ∈ Σ}

NFA TALE CHE X Y NONDETERMINISTICAMENTE

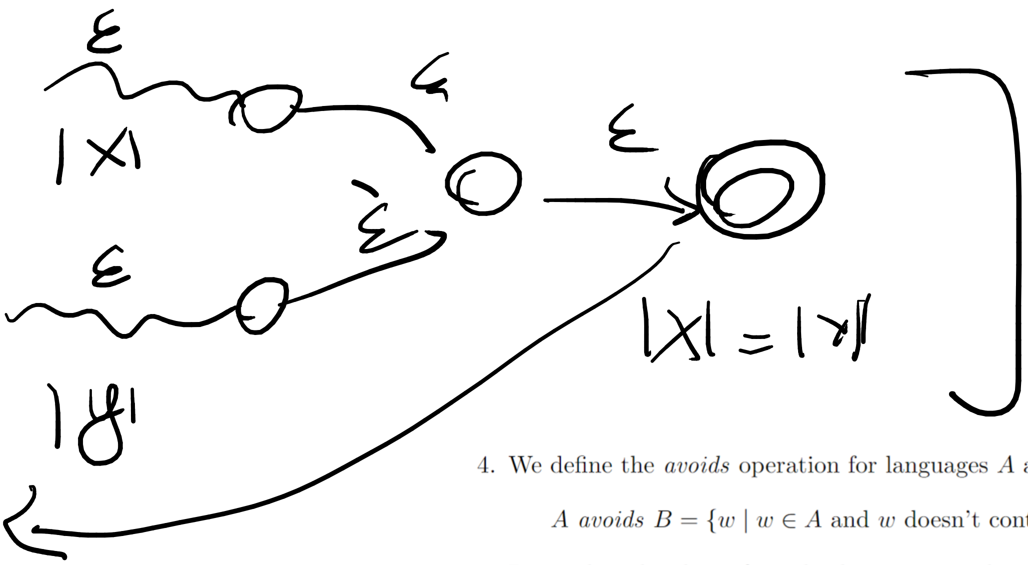
$|X| = |Y|$

1. Partiamo da (q_0, qf) - iniziamo dall'inizio per x, dalla fine per y
2. Leggendo simbolo a da x:
   - Componente "avanti": p → δ(p,a) (normale avanzamento)
   - Componente "indietro": q → r (dove r è tale che δ(r,b) = q per qualche b che "indivinamo" essere in y)
3. Accettiamo quando raggiungiamo (qf, q_0) - abbiamo completato x arrivando a uno stato finale, e completato y "tornando" all'inizio

Interpretazione:

- La prima componente traccia: $q_0 \to^{a_1} q_1 \to^{a_2} \ldots \to^{a_n} qf \in F$
- La seconda componente traccia (al contrario): $qf \leftarrow^{b_n} q_{n-1} \leftarrow^{b_{n-1}} \ldots \leftarrow^{b_1} q_0$

Ma nell'NFA, "←^b" diventa "→^b" scegliendo nondeterministicamente r tale che δ(r,b) = q.



VERIFICA SEPARATA CON METÀ E PREFISSO

4. We define the *avoids* operation for languages A and B to be

$$A \textit{ avoids } B = \{w \mid w \in A \text{ and } w \text{ doesn't contain any string in } B \text{ as a substring}\}.$$

Prove that the class of regular languages is closed under the *avoids* operation.

Solution. The definition of A *avoids* B may be restated equivalently as the set difference between A and $\{w \mid w \text{ contains a string in } B \text{ as a substring}\}$. The set difference between two languages C and D, denoted $C \setminus D$, equals $C \cap \overline{D}$. If both C and D are regular, then $C \setminus D$ is also regular, as the class of regular languages is closed under intersection and complementation.

For a regular language B, the set $\{w \mid w \text{ contains a string in } B \text{ as a substring}\}$ can be expressed as $\Sigma^* R_B \Sigma^*$, where Σ is the alphabet and R_B is a regular expression for B, and hence is also regular. So, the class of regular languages is closed under the *avoids* operation. □

1.53 Let $\Sigma = \{0, 1, +, =\}$ and

$$ADD = \{x{=}y{+}z \mid x, y, z \text{ are binary integers, and } x \text{ is the sum of } y \text{ and } z\}.$$

Show that ADD is not regular.

(FUNZIONA PER L. REGOLARI)

DUE AUTOMI → SOMMA

$$\left.\begin{array}{r} 111\ + \\ 110\ = \\ \hline 101 \end{array}\right.\quad \begin{array}{l} z \\ y \\ x \end{array}$$

$$\underbrace{(10)^{k-p-q}}_{z} = 1^p + 0^q$$

NON VUOTI

$U\,d = \zeta\,U$

$= d + \zeta$

$P > 0$

4

PUÒ LOOPARE!

$$[w = xy^i z,\ i \geq 0,\ y \neq \varepsilon]$$

$$\left\{ 0^{k+1} 1^{k+1} \qquad \frac{k+1}{k+1} = 1 \right\} \quad 0^{n/m} \to \text{è intero}$$

```
Una fabbrica produce sequenze composite inserendo
tra ogni coppia di caratteri di una stringa x ∈ A
una sottostringa di riempimento z,
dove la lunghezza totale del riempimento
deve uguagliare la lunghezza originale di x.

Definisci INTERLEAVE(A) =
{x₁z₁x₂z₂ … xₙzₙxₙ₊₁ | x₁x₂ … xₙ₊₁ ∈ A, z₁z₂ … zₙ ∈ {a,b}*, |z₁z₂ … zₙ| = n}.

Dimostra che se A è regolare, allora INTERLEAVE(A) è context-free
```

∃ NFA EQUIVALENTE

$$\{x_1 z_1 x_2 z_2 \ldots x_n z_n x_{n+1} \mid x_1 \ldots x_{n+1} \in A,\ z_1 \ldots z_n \in \{a,b\}^*,\ |z_1 \ldots z_n| = n\}$$

$$\left\{\begin{array}{l} S \to X_1 S_i \\ S_i \to X_i Z_i S_{i+1} \quad i \in \{1 \ldots n\} \\ \ldots\ F \to X_{n+1} \end{array}\right\}$$

$$\left\{ \begin{array}{l} A \to BC \\ A \to a \end{array} \right\}$$ LOOP infiniti

$\Sigma = \Sigma'$, $V = V'$, $S = S'$

$G = (\Sigma, V, R, S) = G'(\Sigma', V', R', S')$

R → YES – INSERIMENTO

R → NO – INSERIMENTO

CFG – NFA

$(X_1 Z_1 \ldots X_n Z_n) \to$ SOMMARE LE 1, NON AMBIGUA!

$x \in$ INTERVALLI, $\exists$ UNA CFG IN CNF TALE $X_1 Z_1 \ldots X_n Z_n X_{n+1}$

[REGOLE] → OUTPUT

$f(x) \in$ OUTPUT | $\exists$ UNA CFG ...