

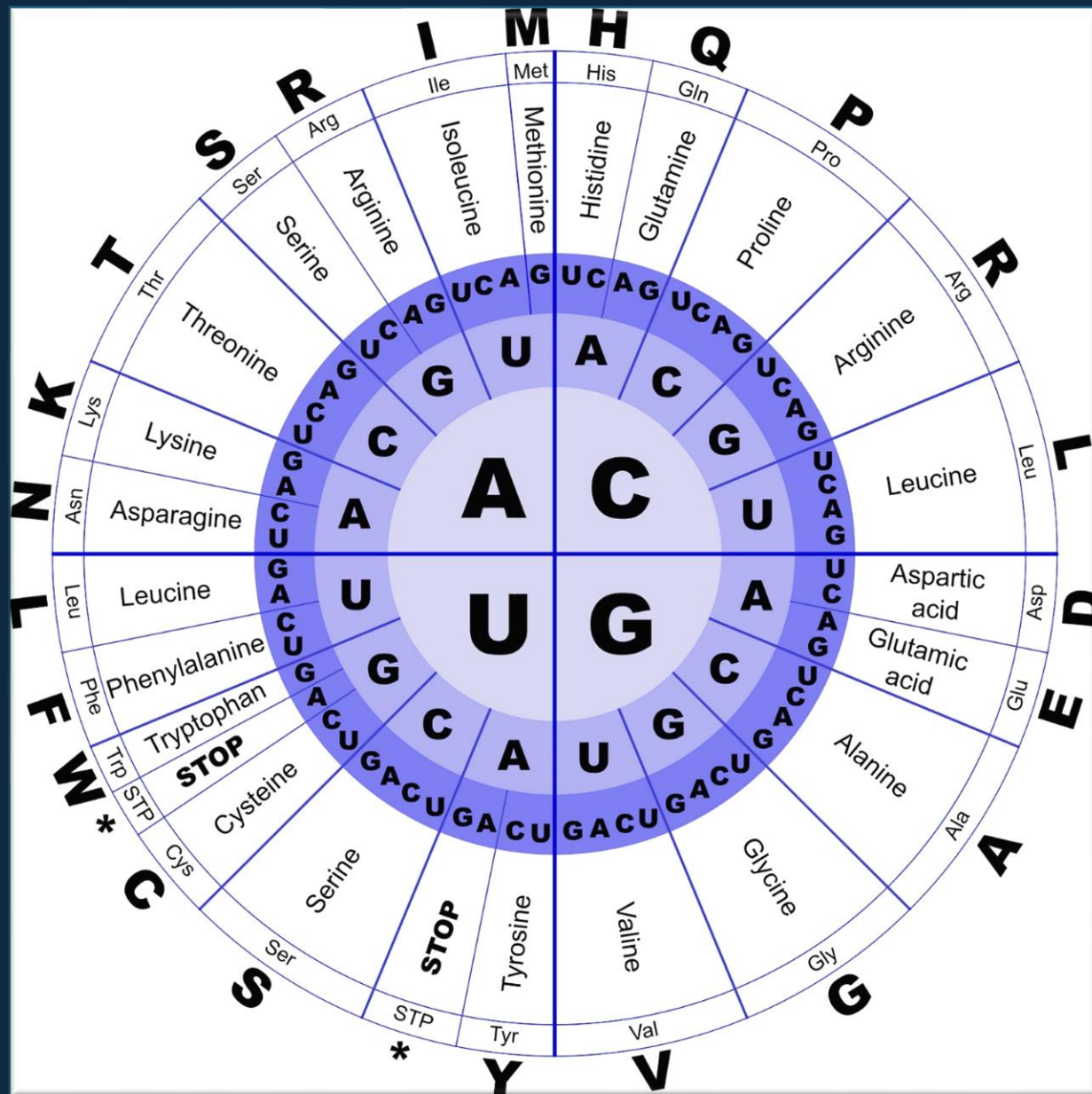
Automa a stati finiti per il riconoscimento delle proteine

Emanuele Galiano
Dipartimento di matematica e informatica UNICT

Breve introduzione

L'RNA viene tradotto in
proteine quando abbiamo:

- Una sequenza congrua a 0 in modulo 3
- Terne di amminoacidi fino ad una particolare terna chiamata codone di STOP



Automa a stati finiti

Teoria

Un automa a stati finiti (ASFD) viene matematicamente rappresentato come $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ dove:

1. Q è l'insieme degli stati
2. Σ è l'alfabeto finito
3. $\delta: Q \times \Sigma \rightarrow Q$ funzione di transizione
4. q_0 è lo stato iniziale
5. F è l'insieme degli stati finali

Tabella di transizione

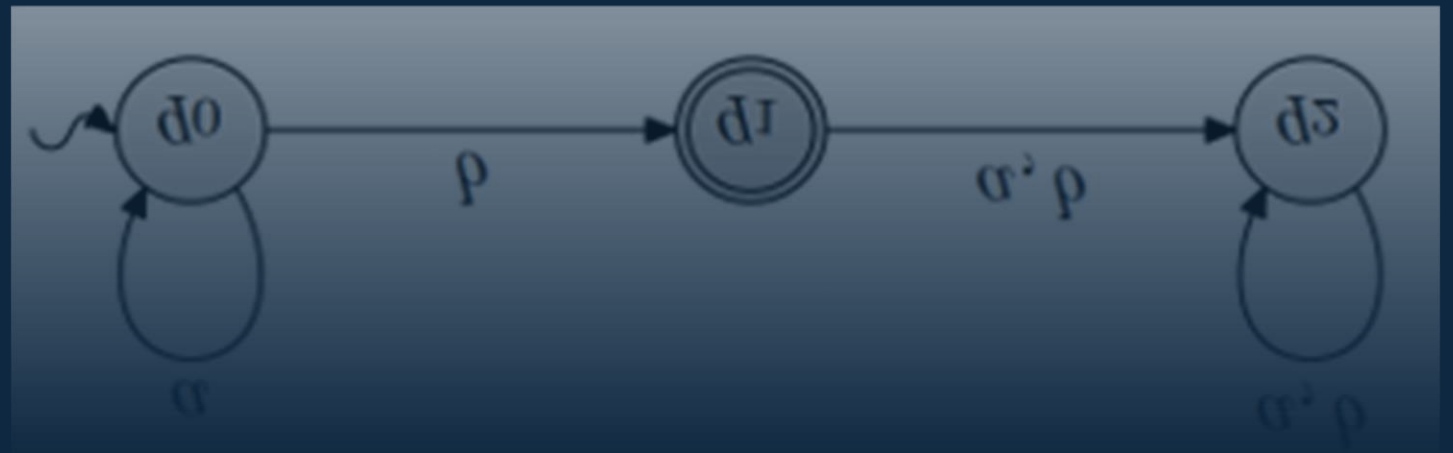
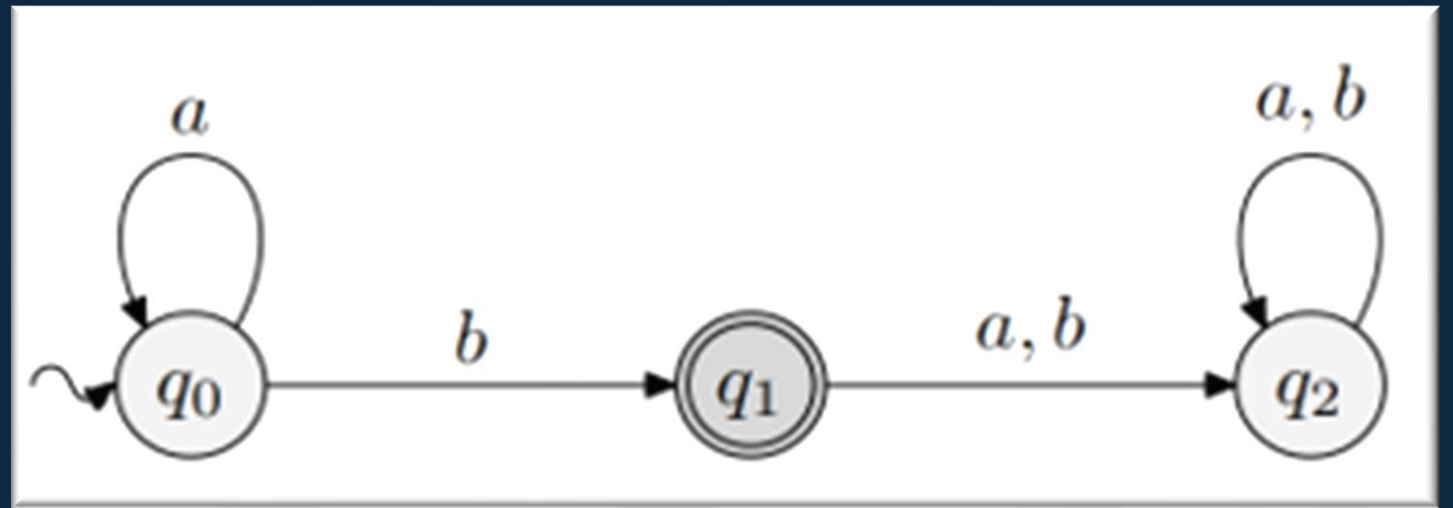
δ	a	b
q_0	q_0	q_1
q_1	q_2	q_2
q_2	q_2	q_2

δ	δ	δ
----------	----------	----------

Grafo dell'automa

Un automa può essere rappresentato anche come un grafo con:

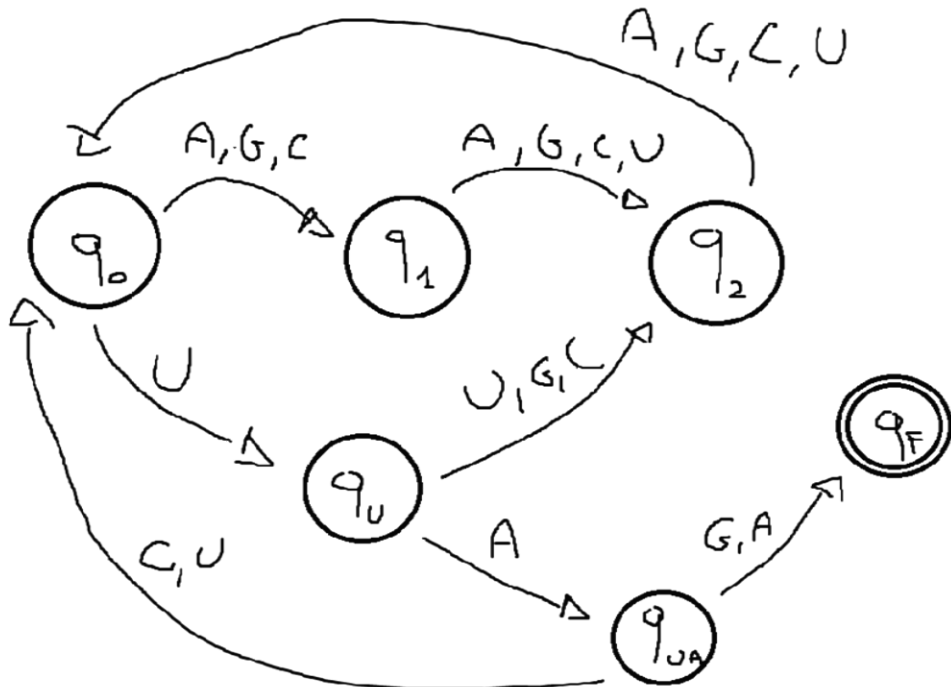
- **Vertici** che rappresentano gli stati
- **Vertici doppiamente segnati** che rappresentano gli stati finali
- **Arch**i che rappresentano i caratteri letti dall'automa



Riconoscimento delle proteine

Per effettuare il riconoscimento delle proteine attraverso un automa si ha:

$$R = \{X * Y : X \in \Sigma^{3n} \setminus Y, n \in \mathbb{N}, 0 \leq n \leq 2^{30}\}$$
$$Y = \{UAG, UAA, UGA\}$$



In particolare, l'automata ha:

- L'insieme $Q = \{q_0, q_1, q_2, q_U, q_{UA}, q_F\}$
- $\Sigma = \{A, G, C, U\}$
- $F = \{q_F\}$

Piccolo Script in Python

```
recognition.py > translate
1 def translate(sequence: str) -> str:
2     protein: str = ""
3     ammToProt: dict = {"UUU" : "F", "CUU" : "L", "AUU" : "I", "GUU" : "V",
4         "UUC" : "F", "CUC" : "L", "AUC" : "I", "GUC" : "V",
5         "UUA" : "L", "CUA" : "L", "AUA" : "I", "GUA" : "V",
6         "UUG" : "L", "CUG" : "L", "AUG" : "M", "GUG" : "V",
7         "UCU" : "S", "CCU" : "P", "ACU" : "T", "GCU" : "A",
8         "UCC" : "S", "CCC" : "P", "ACC" : "T", "GCC" : "A",
9         "UCA" : "S", "CCA" : "P", "ACA" : "T", "GCA" : "A",
10        "UCG" : "S", "CCG" : "P", "ACG" : "T", "GCG" : "A",
11        "UAU" : "Y", "CAU" : "H", "AAU" : "N", "GAU" : "D",
12        "UAC" : "Y", "CAC" : "H", "AAC" : "N", "GAC" : "D",
13        "UAA" : "STOP", "CAA" : "Q", "AAA" : "K", "GAA" : "E",
14        "UAG" : "STOP", "CAG" : "Q", "AAG" : "K", "GAG" : "E",
15        "UGU" : "C", "CGU" : "R", "AGU" : "S", "GGU" : "G",
16        "UGC" : "C", "CGC" : "R", "AGC" : "S", "GGC" : "G",
17        "UGA" : "STOP", "CGA" : "R", "AGA" : "R", "GGA" : "G",
18        "UGG" : "W", "CGG" : "R", "AGG" : "R", "GGG" : "G"
19    }
20
21    foundCodonStop: bool = False
22    i: int = 0
23
24    while(not foundCodonStop):
25        if ammToProt[sequence[i:i+3]] == "STOP":
26            foundCodonStop = True
27        else:
28            protein += ammToProt[sequence[i:i+3]]
29            i += 3
30
31    return protein
32
```

```
33
34    return protein
35
```