AUTOMI: dall'analisi di un esempio alla loro formalizzazione

Fondamenti di Programmazione: AUTOMI E LINGUAGGI FORMALI
Corso di Laurea in MATEMATICA
a.a. 2023/2024

PROBLEMA 1

Un robot lancia in aria una moneta. I primi due tiri su tre danno lo stesso risultato (**T**esta, **T**esta o **C**roce**, C**roce).

Trovare l'automa A che riconosce se una singola tripletta w è corretta rispetto alle regola:

L = {w in {T,C}* è OK sse w = ε oppure w = xxy, con x,y in {T,C}}

Stringhe w composte da elementi in {T,C}

PROBLEMA 1

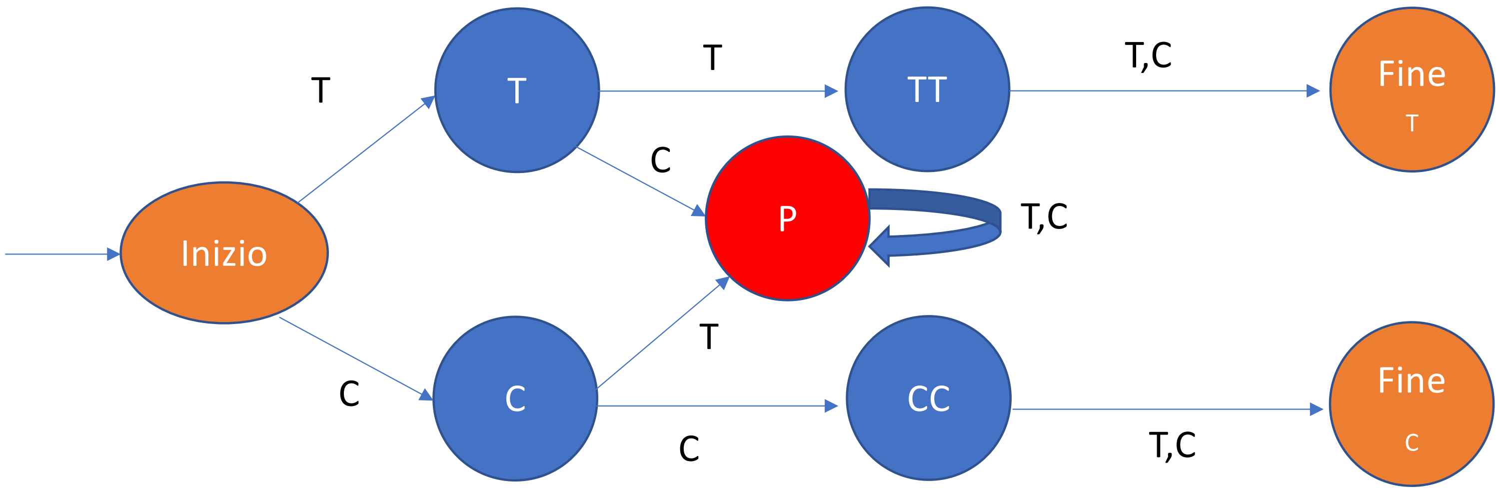
Trovare l'automa A che riconosce se una singola tripletta w è corretta rispetto alle regola:

L = {w in {T,C}* è OK sse w = ε oppure w = xxy, con x,y in {T,C}}

PROBLEMA 1

Trovare l'automa A che riconosce se una singola tripletta w è corretta rispetto alle regola:

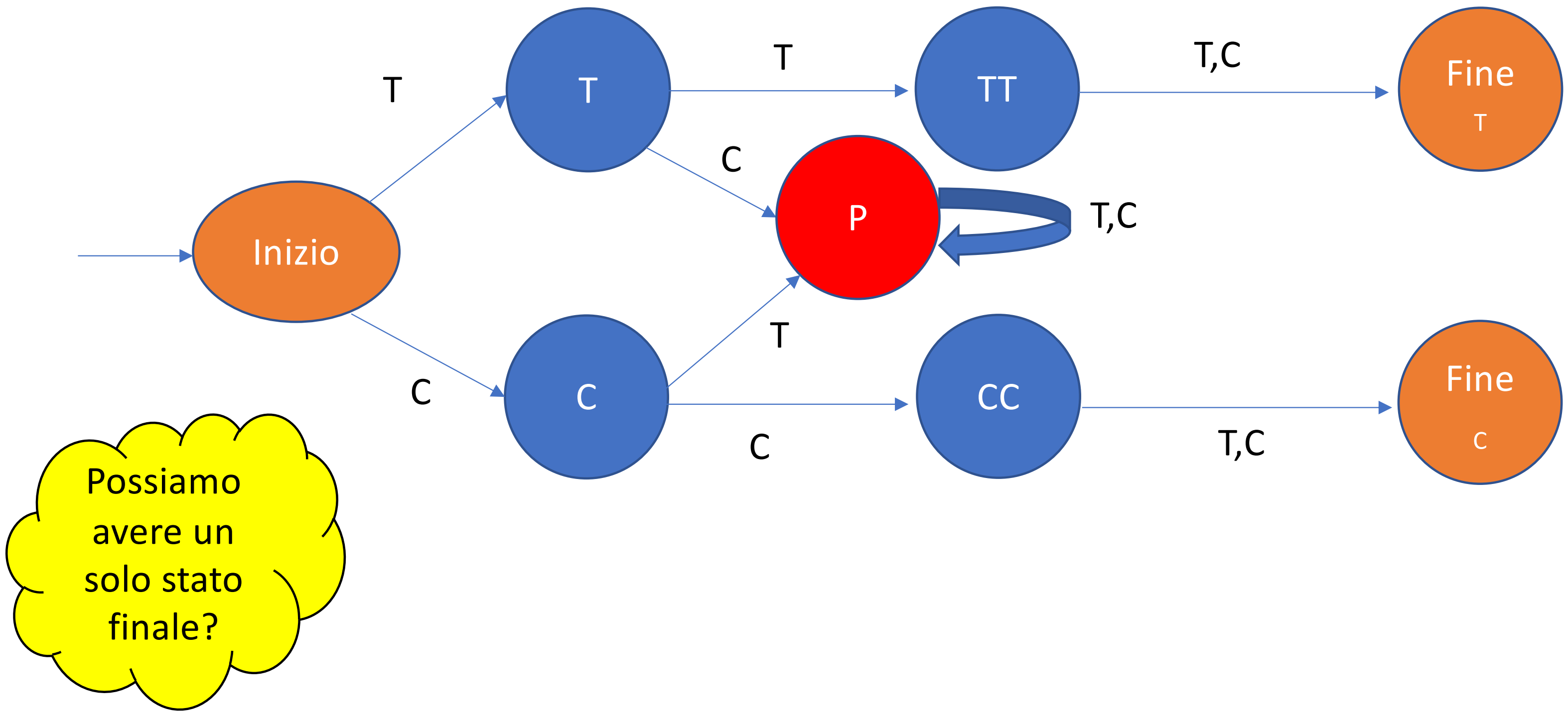
L = {w in {T,C}* è OK sse w = ε oppure w = xxy, con x,y in {T,C}}



PROBLEMA 1

Trovare l’automa A che riconosce se una singola tripletta w è corretta rispetto alle regola:

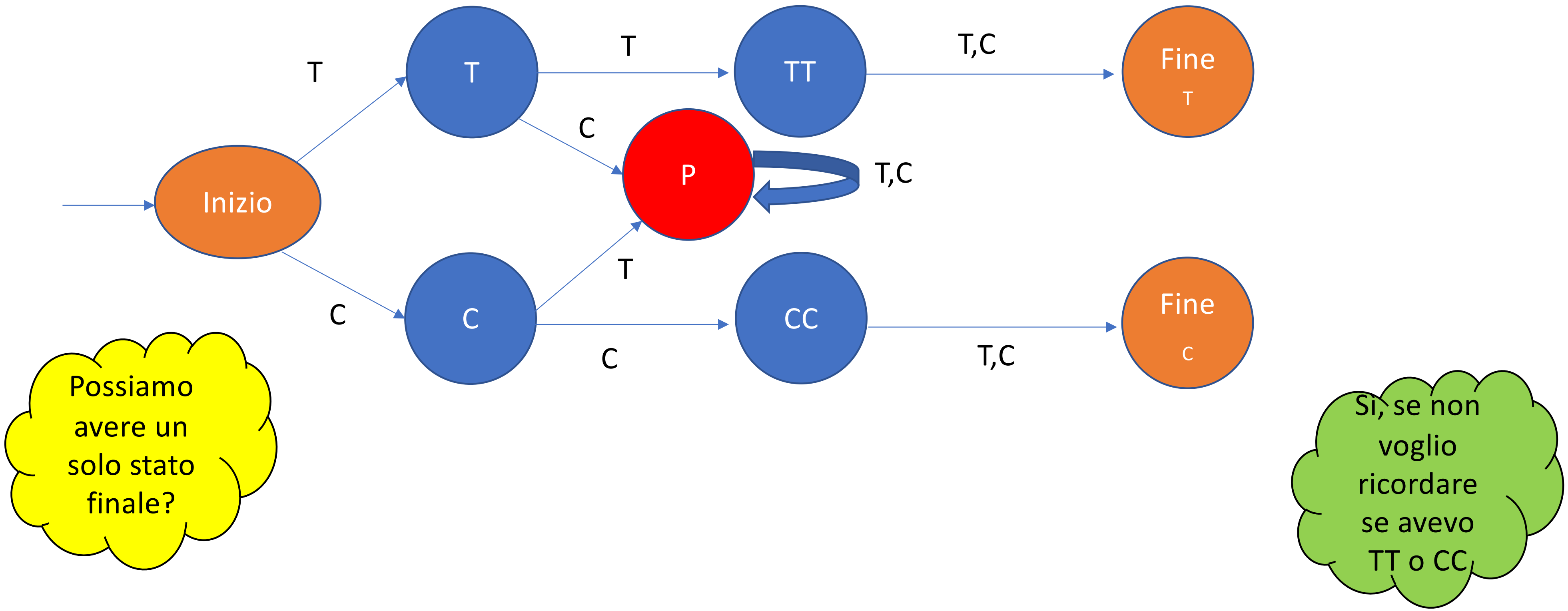
L = {w in {T,C}* è OK sse w = ε oppure w = xxy, con x,y in {T,C}}



PROBLEMA 1

Trovare l'automa A che riconosce se una singola tripletta w è corretta rispetto alle regola:

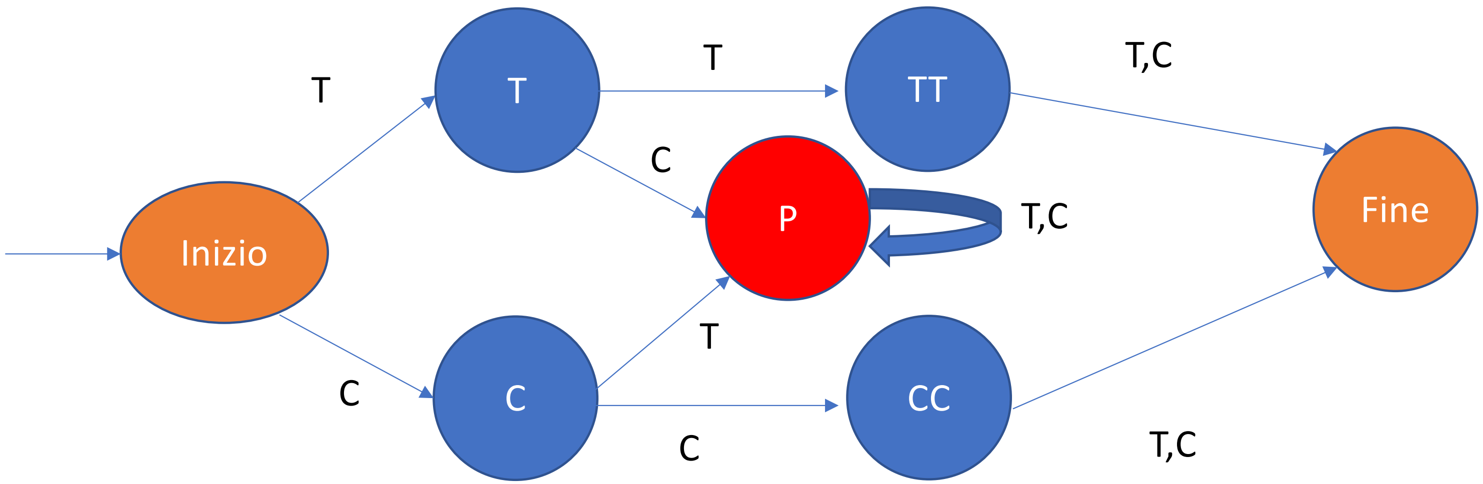
L = {w in {T,C}* è OK sse w = ε oppure w = xxy, con x,y in {T,C}}



PROBLEMA 1

Trovare l'automa A che riconosce se una singola tripletta w è corretta rispetto alle regola:

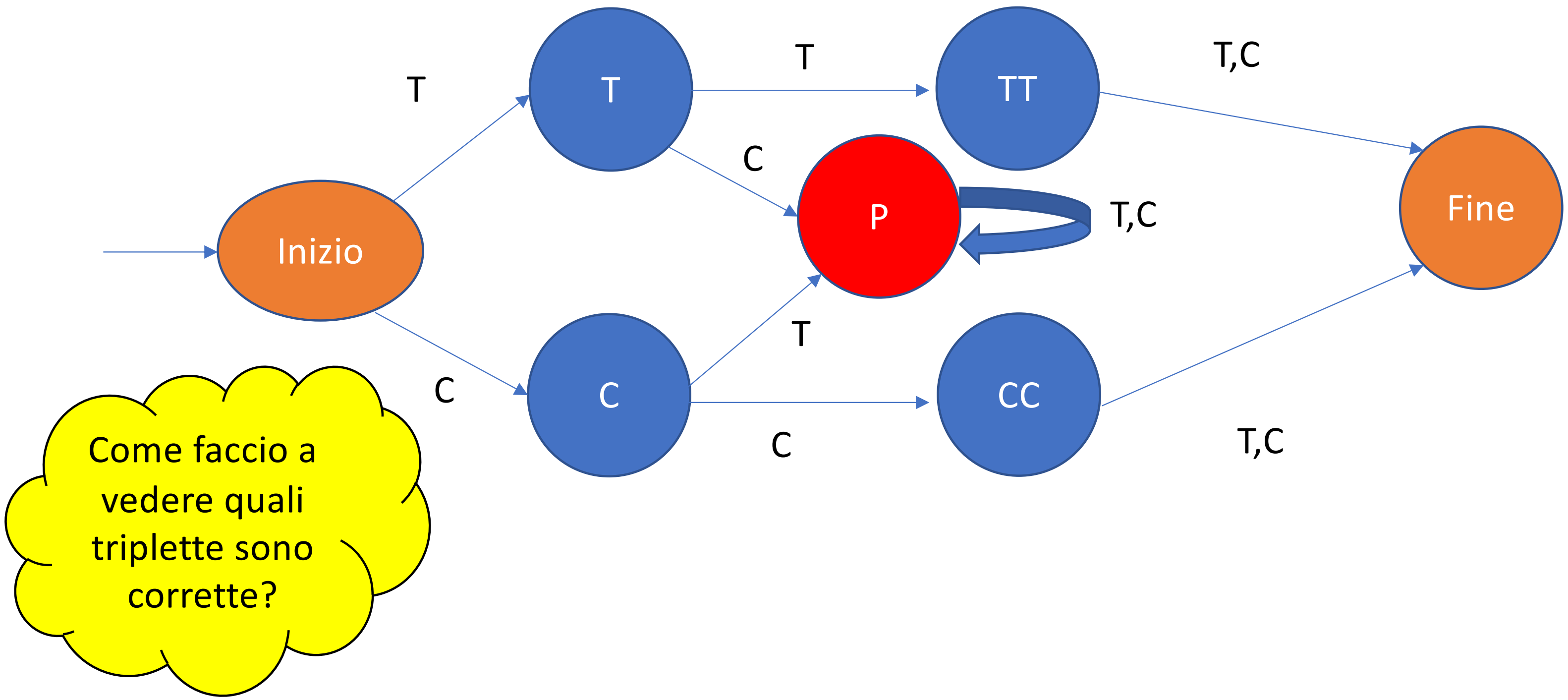
L = {w in {T,C}* è OK sse w = ε oppure w = xxy, con x,y in {T,C}}



PROBLEMA 1

Trovare l’automa A che riconosce se una singola tripletta w è corretta rispetto alle regola:

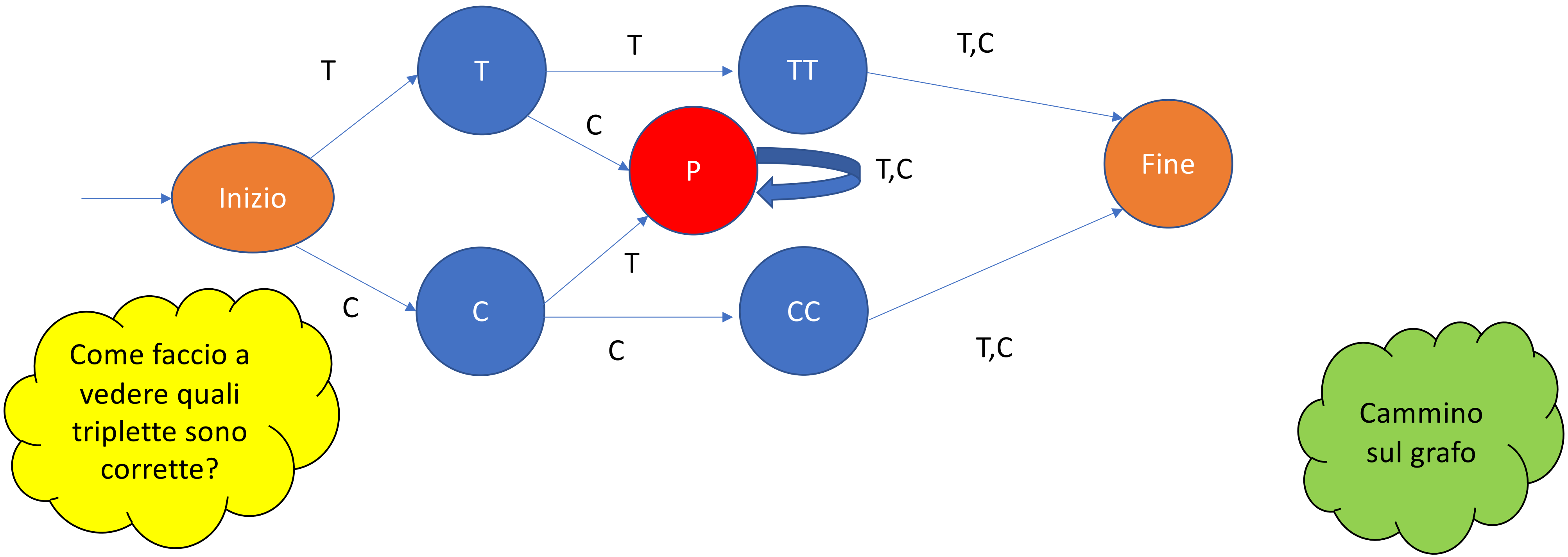
L = {w in {T,C}* è OK sse w = ε oppure w = xxy, con x,y in {T,C}}



PROBLEMA 1

Trovare l'automa A che riconosce se una singola tripletta w è corretta rispetto alle regola:

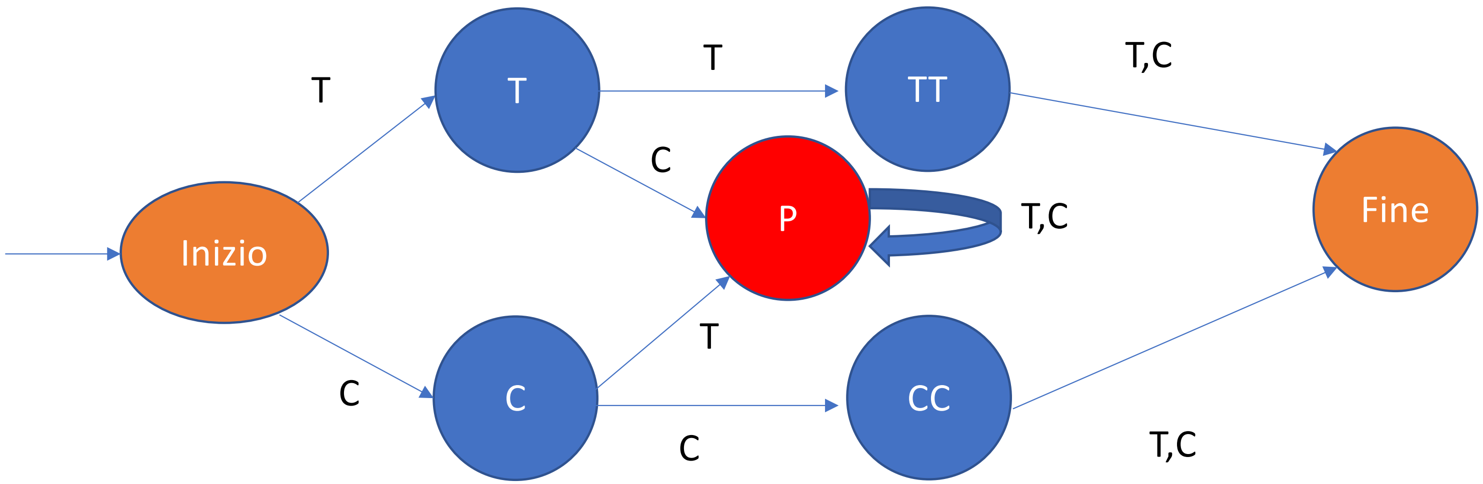
L = {w in {T,C}* è OK sse w = ε oppure w = xxy, con x,y in {T,C}}

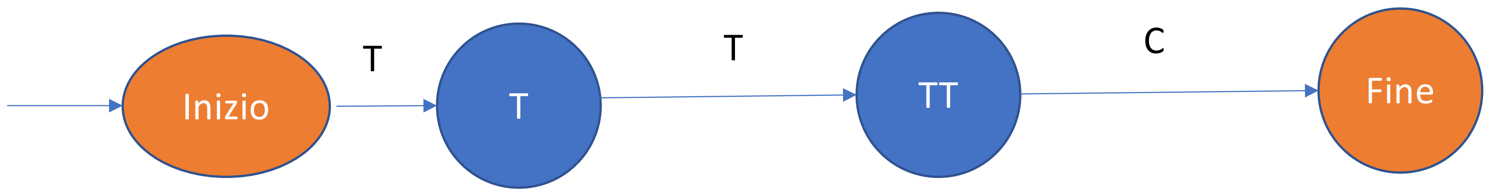


PROBLEMA 1

Trovare l'automa A che riconosce se una singola tripletta w è corretta rispetto alle regola:

L = {w in {T,C}* è OK sse w = ε oppure w = xxy, con x,y in {T,C}}



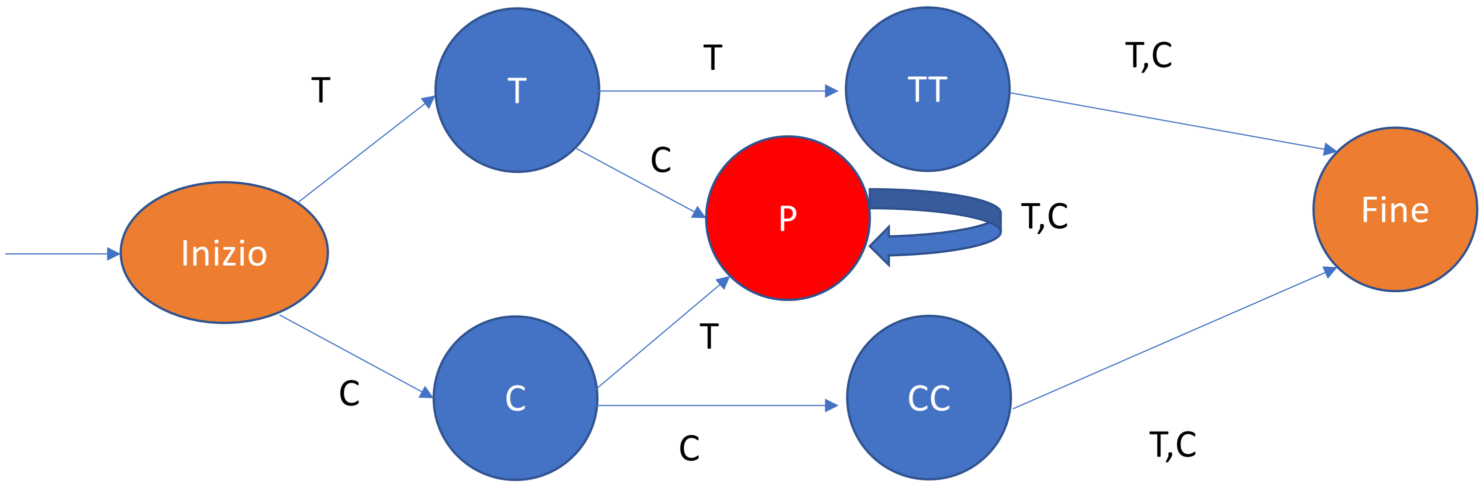


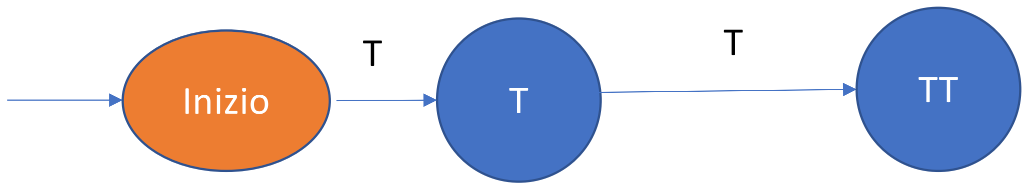
Sequenza accettata TTC

PROBLEMA 1

Trovare l'automa A che riconosce se una singola tripletta w è corretta rispetto alle regola:

L = {w in {T,C}* è OK sse w = ε oppure w = xxy, con x,y in {T,C}}



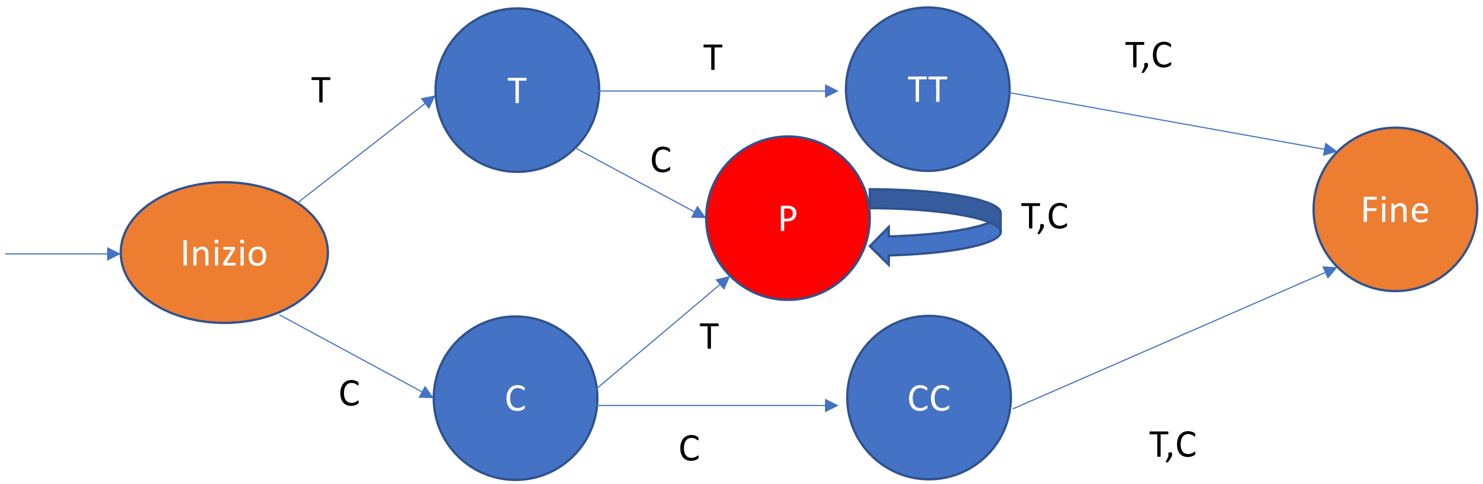


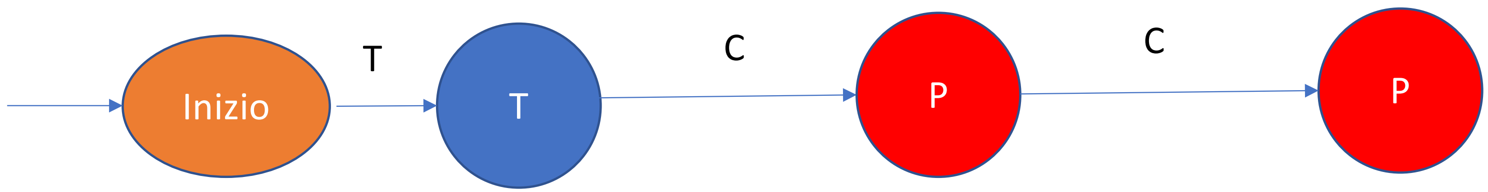
Sequenza non accettata
TT

PROBLEMA 1

Trovare l’automa A che riconosce se una singola tripletta w è corretta rispetto alle regola:

L = {w in {T,C}* è OK sse w = ε oppure w = xxy, con x,y in {T,C}}



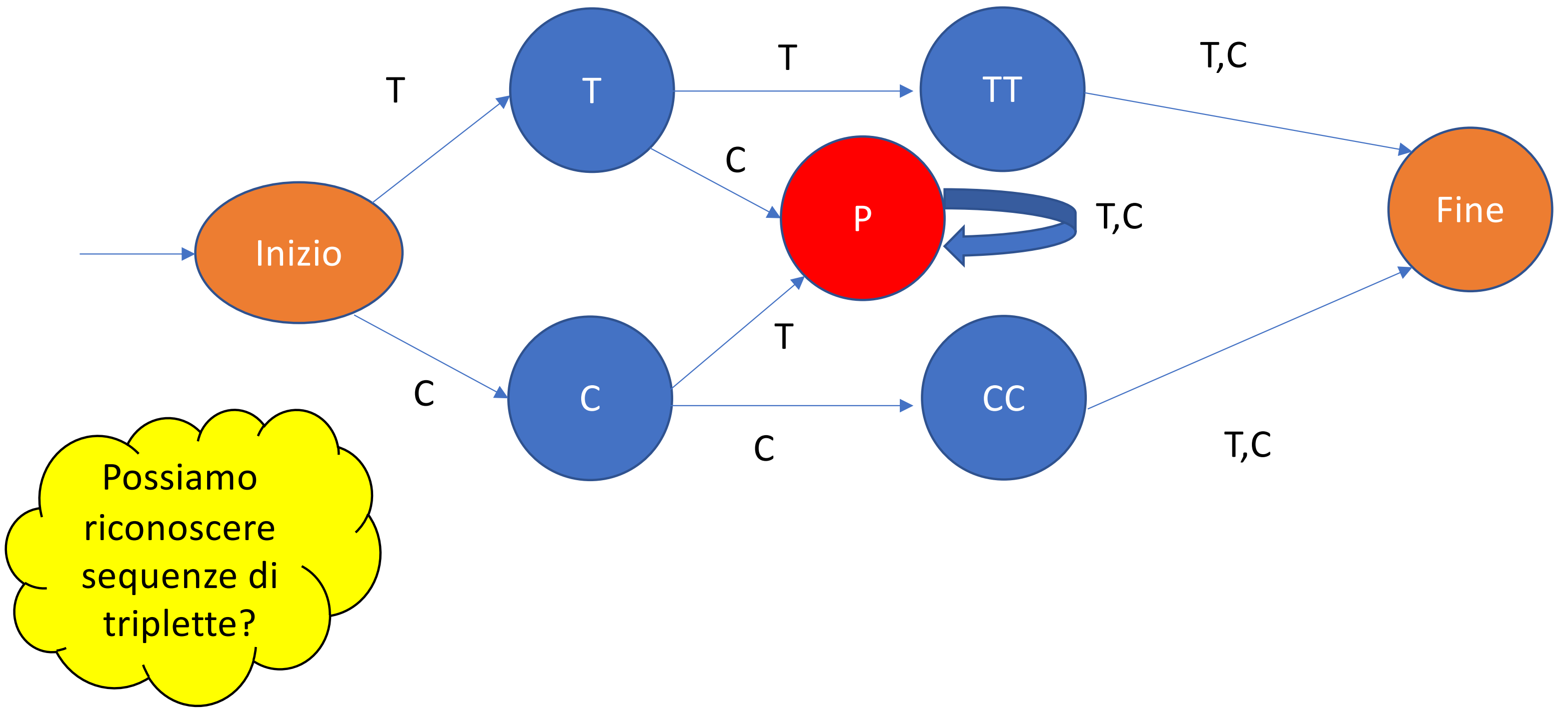


Sequenza rifiutata
TCC

PROBLEMA 1

Trovare l'automa A che riconosce se una singola tripletta w è corretta rispetto alle regola:

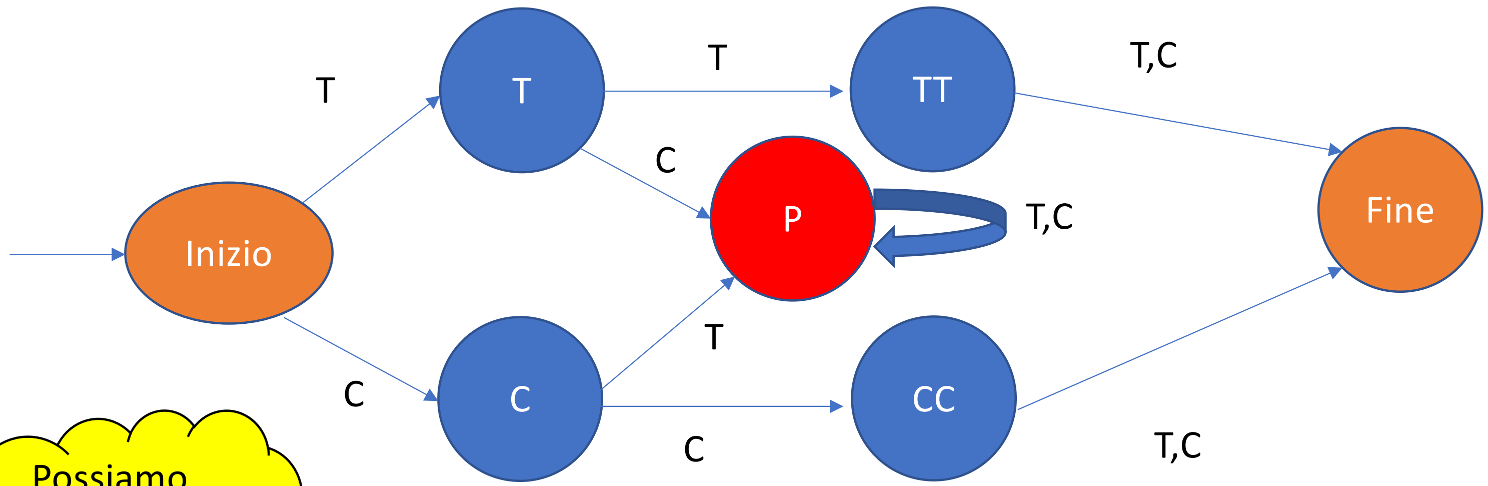
L = {w in {T,C}* è OK sse w = ε oppure w = xxy, con x,y in {T,C}}



PROBLEMA 1

Trovare l'automa A che riconosce se una singola tripletta w è corretta rispetto alle regola:

L = {w in {T,C}* è OK sse w = ε oppure w = xxy, con x,y in {T,C}}



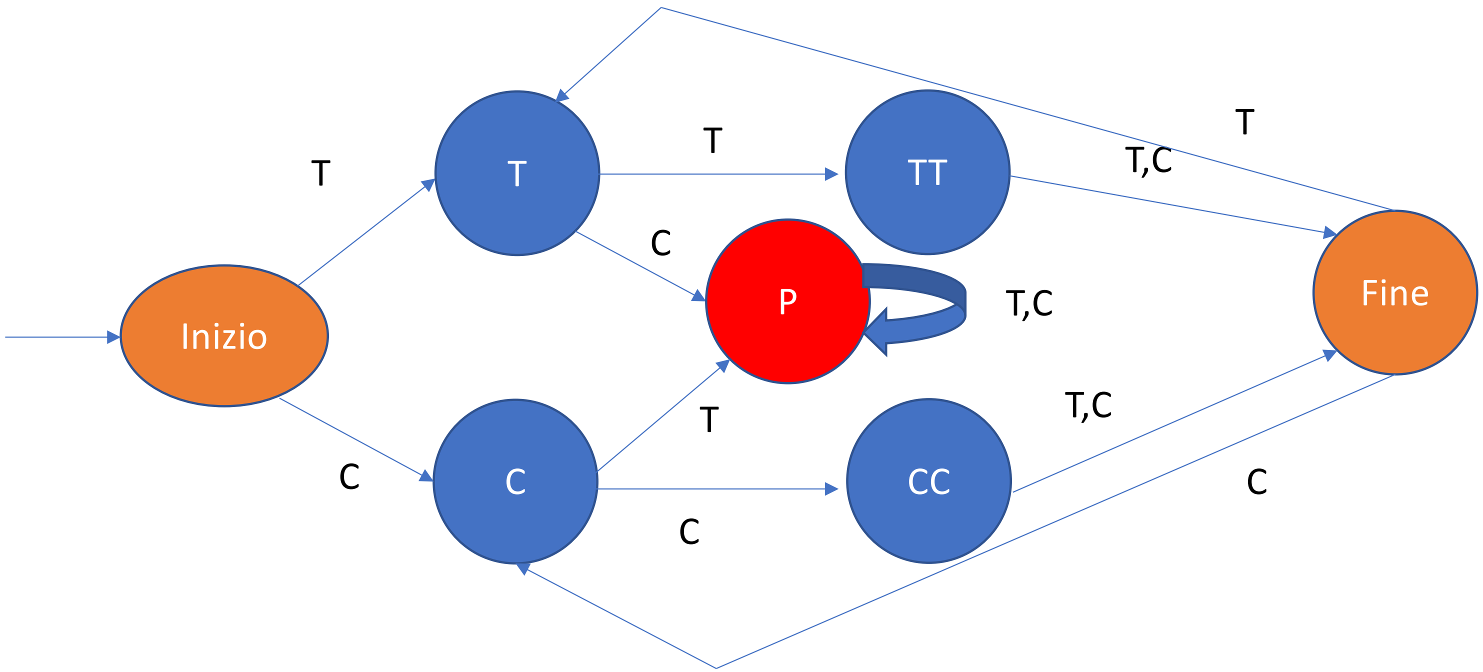
Possiamo riconoscere sequenze di triplette?

Sì, se ogni volta posso iniziare ancora

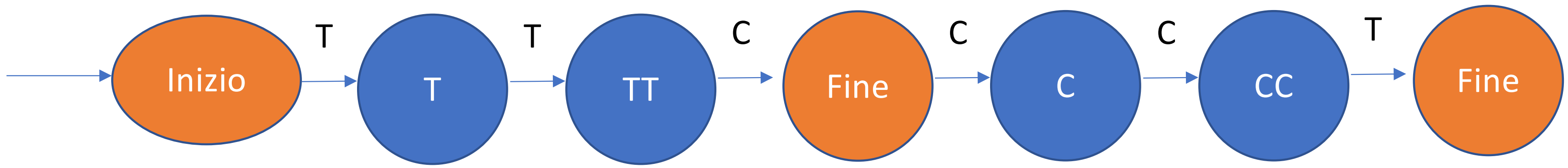
PROBLEMA 2

Trovare l'automa A che riconosce se una sequenza di triplette w abbia tutte le triplette in regola:

L' = {w in {T,C}* è OK sse w = ε oppure w =w_1…w_k && w_i =xxy, con x,y in {T,C}}



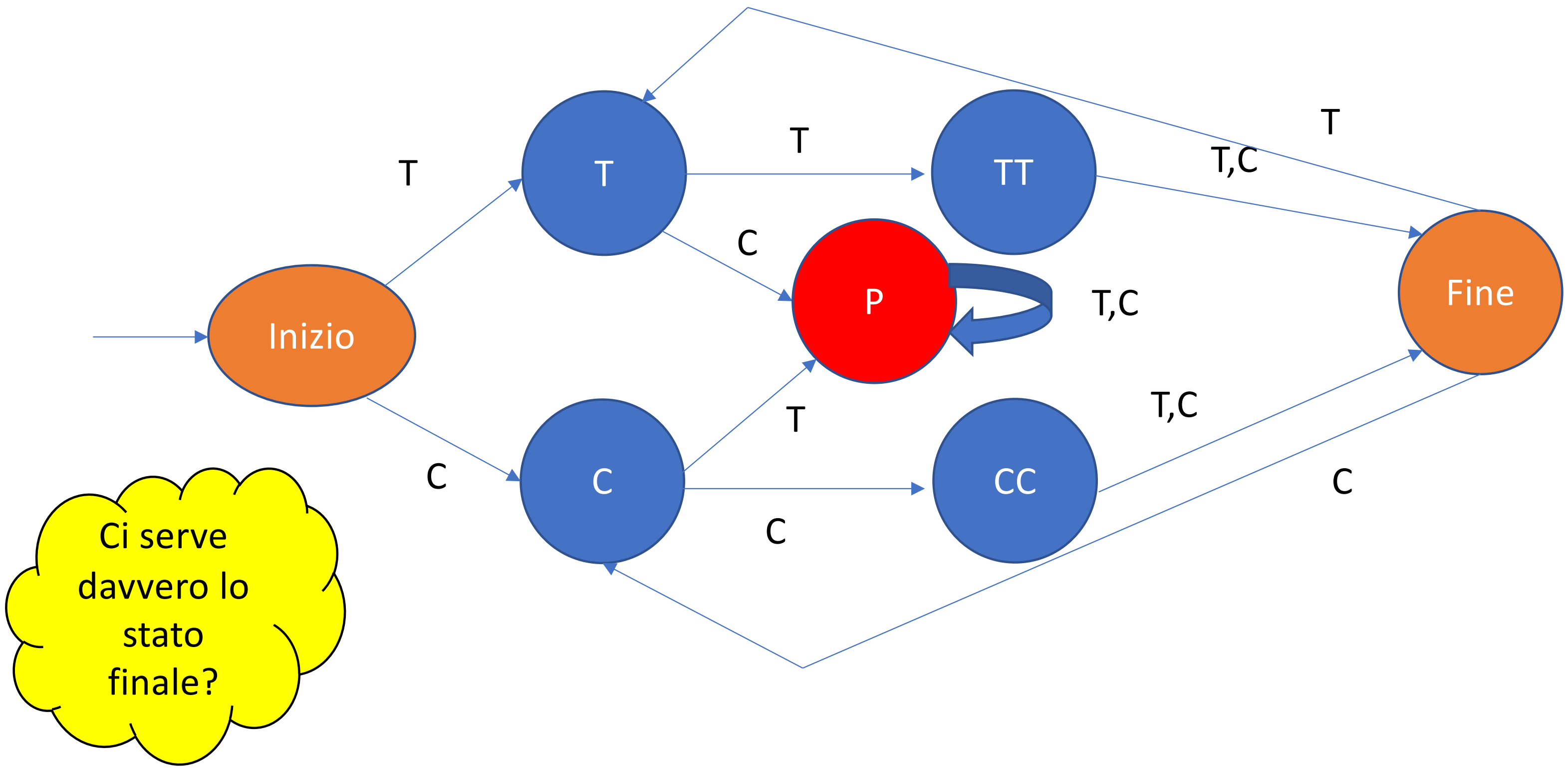
Sequenza accettata
TTC CCT



PROBLEMA 2

Trovare l’automa A che riconosce se una sequenza di triplette w abbia tutte le triplette in regola:

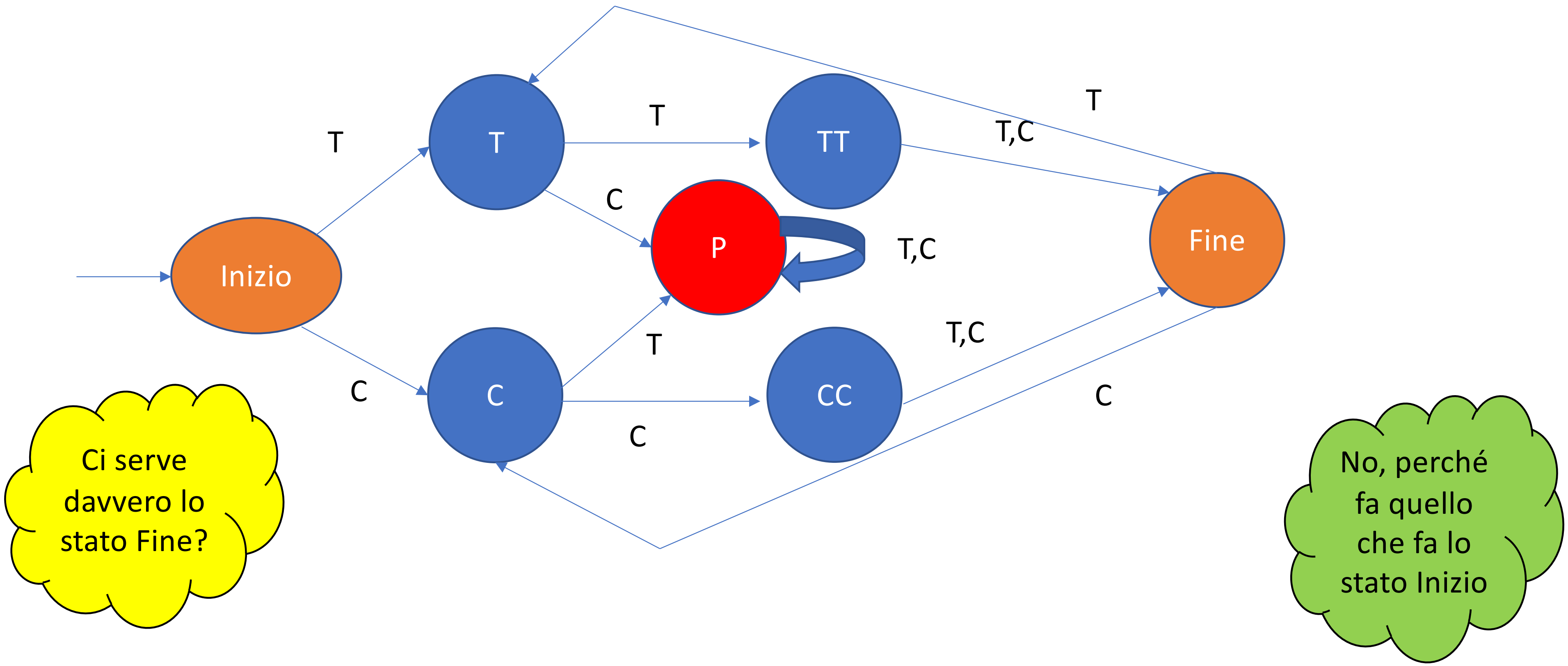
L’ = {w in {T,C}* è OK sse w = ε oppure w =w_1…w_k && w_i =xxy, con x,y in {T,C}}



PROBLEMA 2

Trovare l'automa A che riconosce se una sequenza di triplette w abbia tutte le triplette in regola:

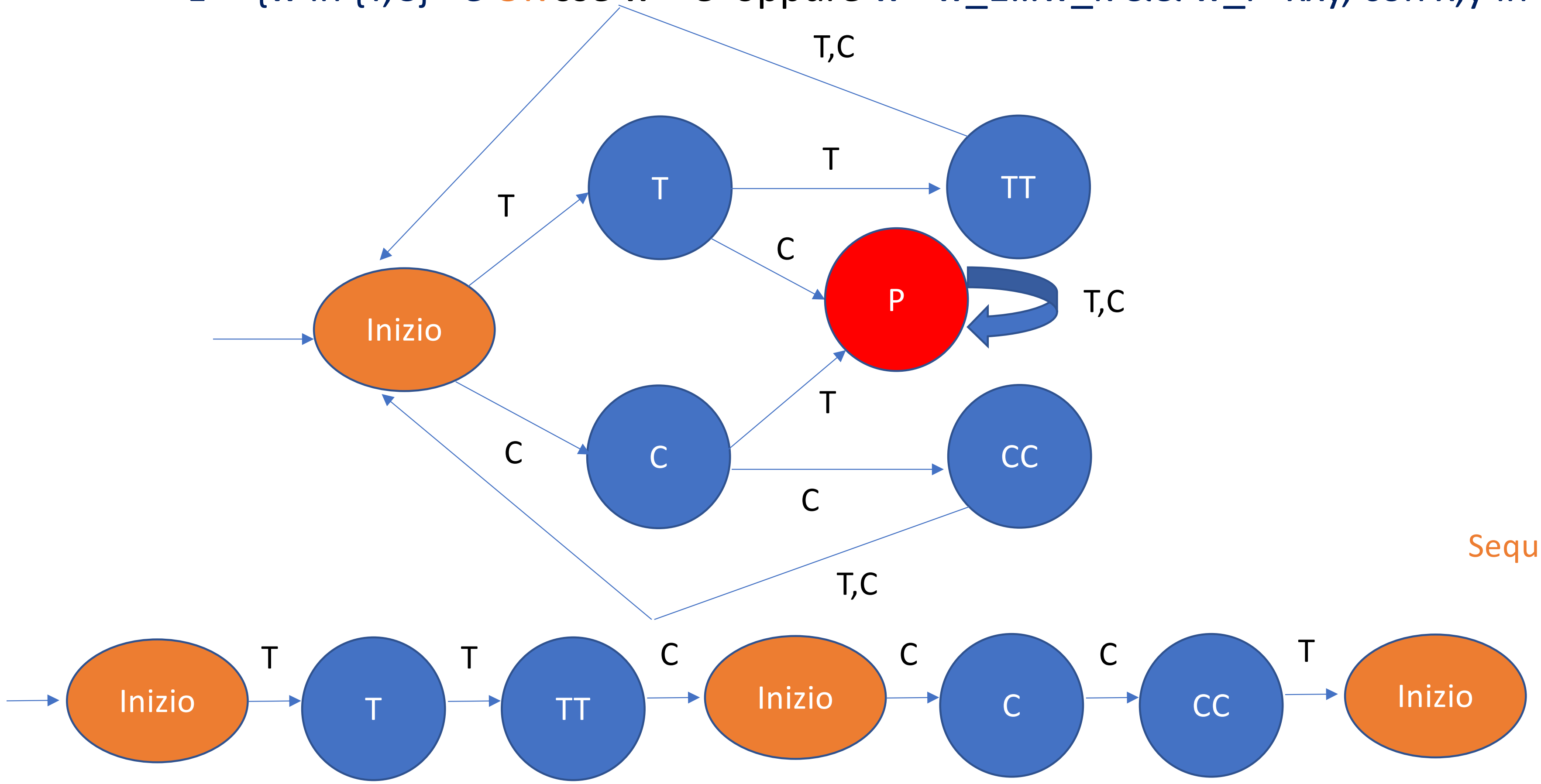
L' = {w in {T,C}* è OK sse w = ε oppure w =w_1…w_k && w_i =xxy, con x,y in {T,C}}



PROBLEMA 2

Trovare l'automa A che riconosce se una sequenza di triplette w abbia tutte le triplette in regola:

L' = {w in {T,C}* è OK sse w = ε oppure w =w_1…w_k && w_i =xxy, con x,y in {T,C}}

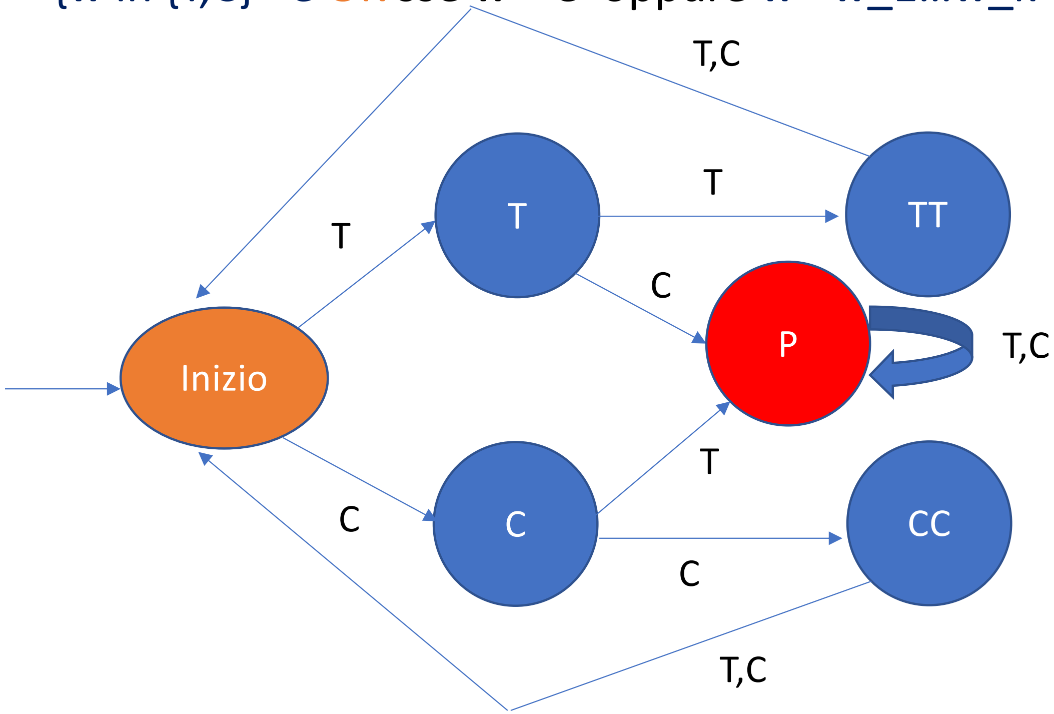


Sequenza accettata
TTC CCT

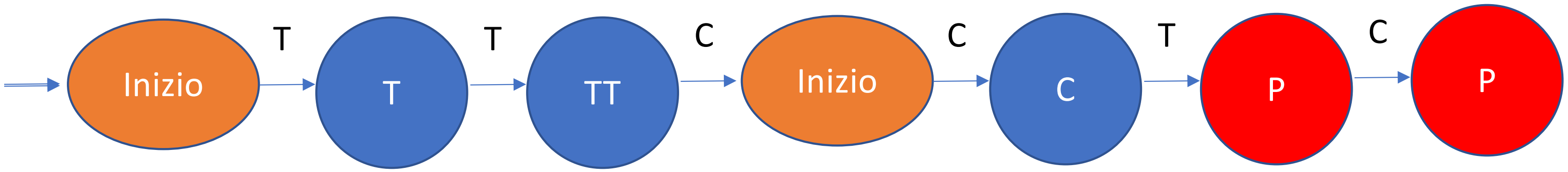
PROBLEMA 2

Trovare l'automa A che riconosce se una sequenza di triplette w abbia tutte le triplette in regola:

L' = {w in {T,C}* è OK sse w = ε oppure w =w_1…w_k && w_i =xyy, con x,y in {T,C}}



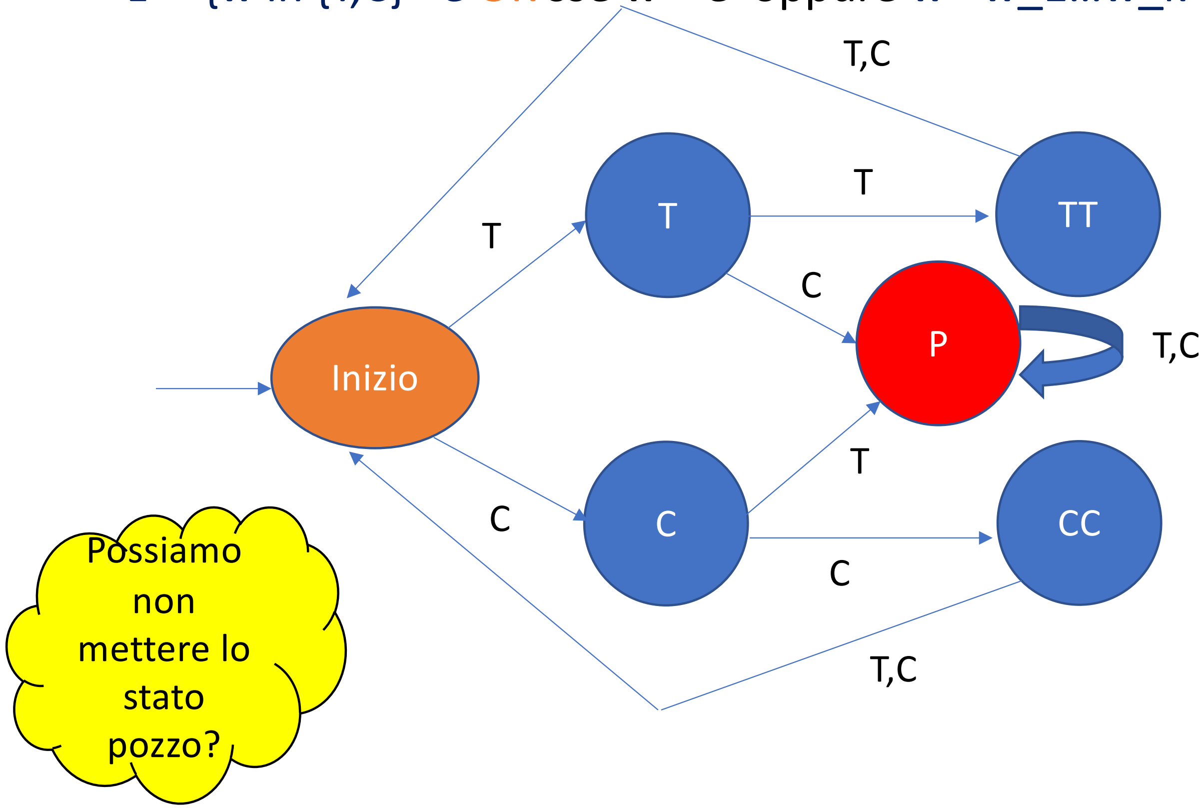
Sequenza rifiutata
TTC CTC



PROBLEMA 2

Trovare l’automa A che riconosce se una sequenza di triplette w abbia tutte le triplette in regola:

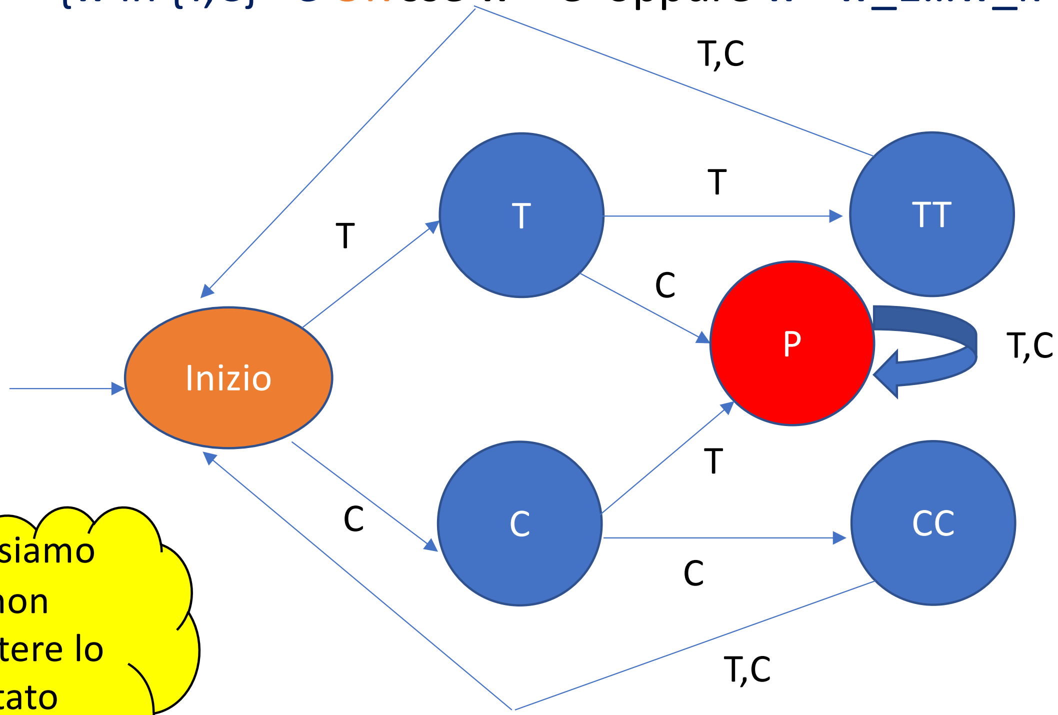
L’ = {w in {T,C}* è OK sse w = ε oppure w =w_1…w_k && w_i =xxy, con x,y in {T,C}}



PROBLEMA 2

Trovare l'automa A che riconosce se una sequenza di triplette w abbia tutte le triplette in regola:

L' = {w in {T,C}* è OK sse w = ε oppure w =w_1…w_k && w_i =xxy, con x,y in {T,C}}



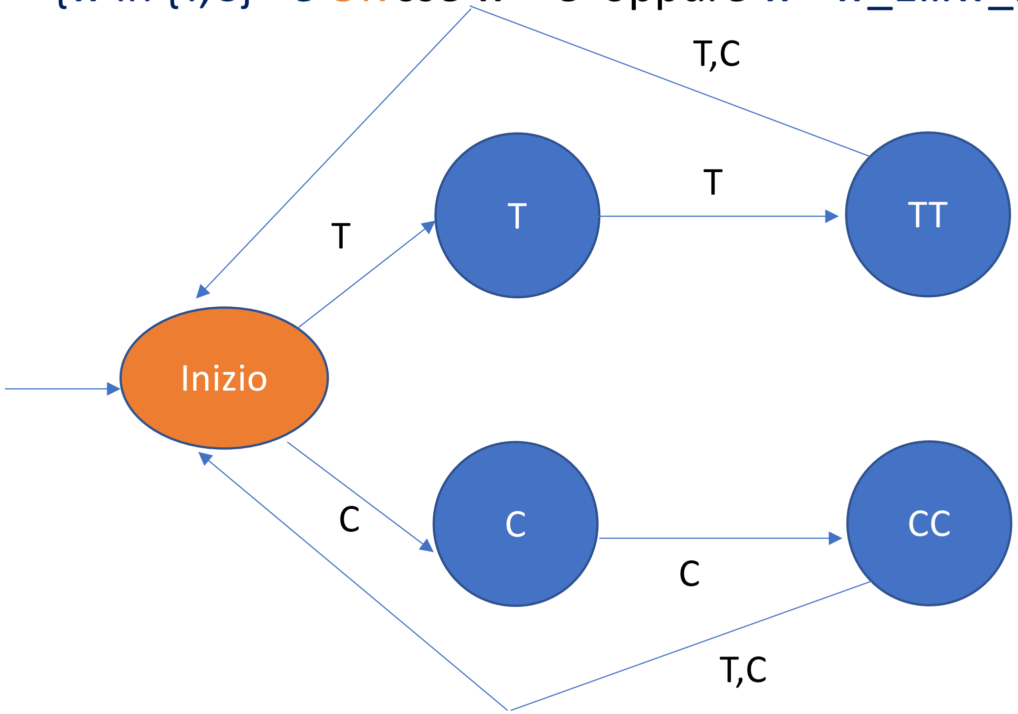
Possiamo non mettere lo stato pozzo?

Sì, possiamo considerarlo implicitamente

PROBLEMA 2

Trovare l’automa A che riconosce se una sequenza di triplette w abbia tutte le triplette in regola:

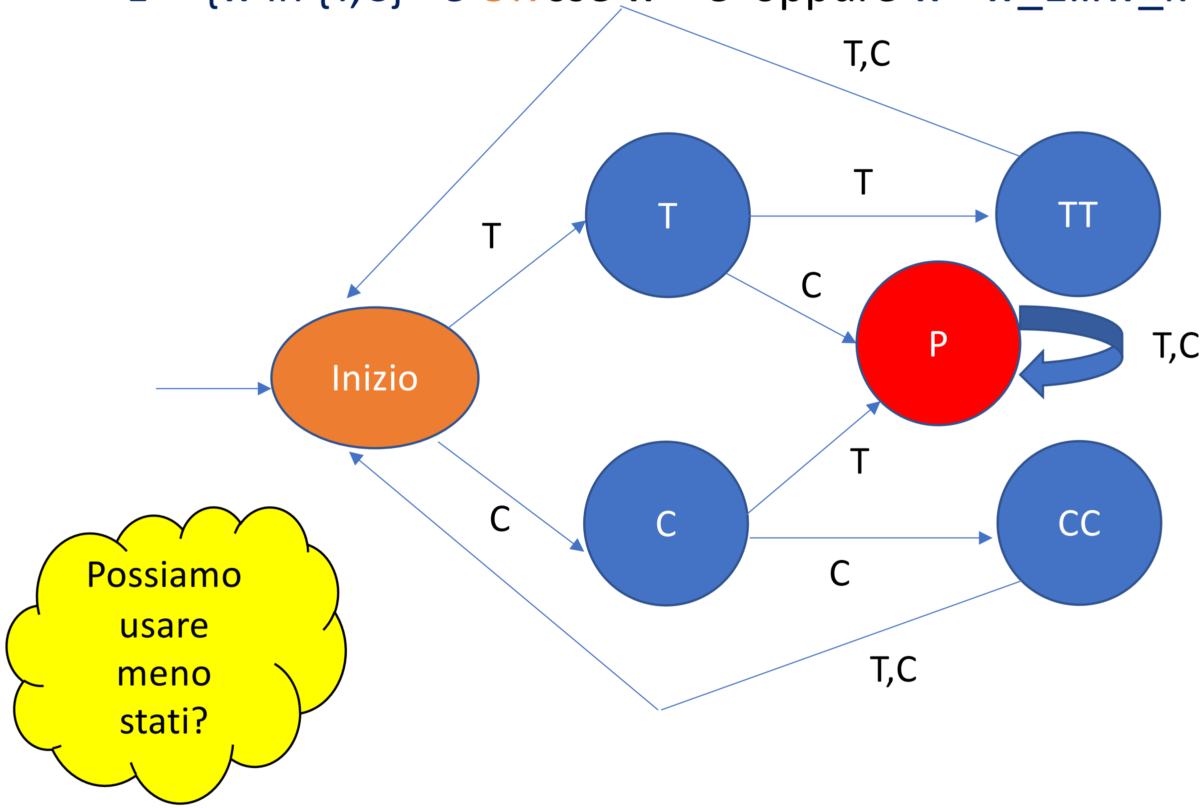
L’ = {w in {T,C}* è OK sse w = ε oppure w =w_1…w_k && w_i =xxy, con x,y in {T,C}}



PROBLEMA 2

Trovare l'automa A che riconosce se una sequenza di triplette w abbia tutte le triplette in regola:

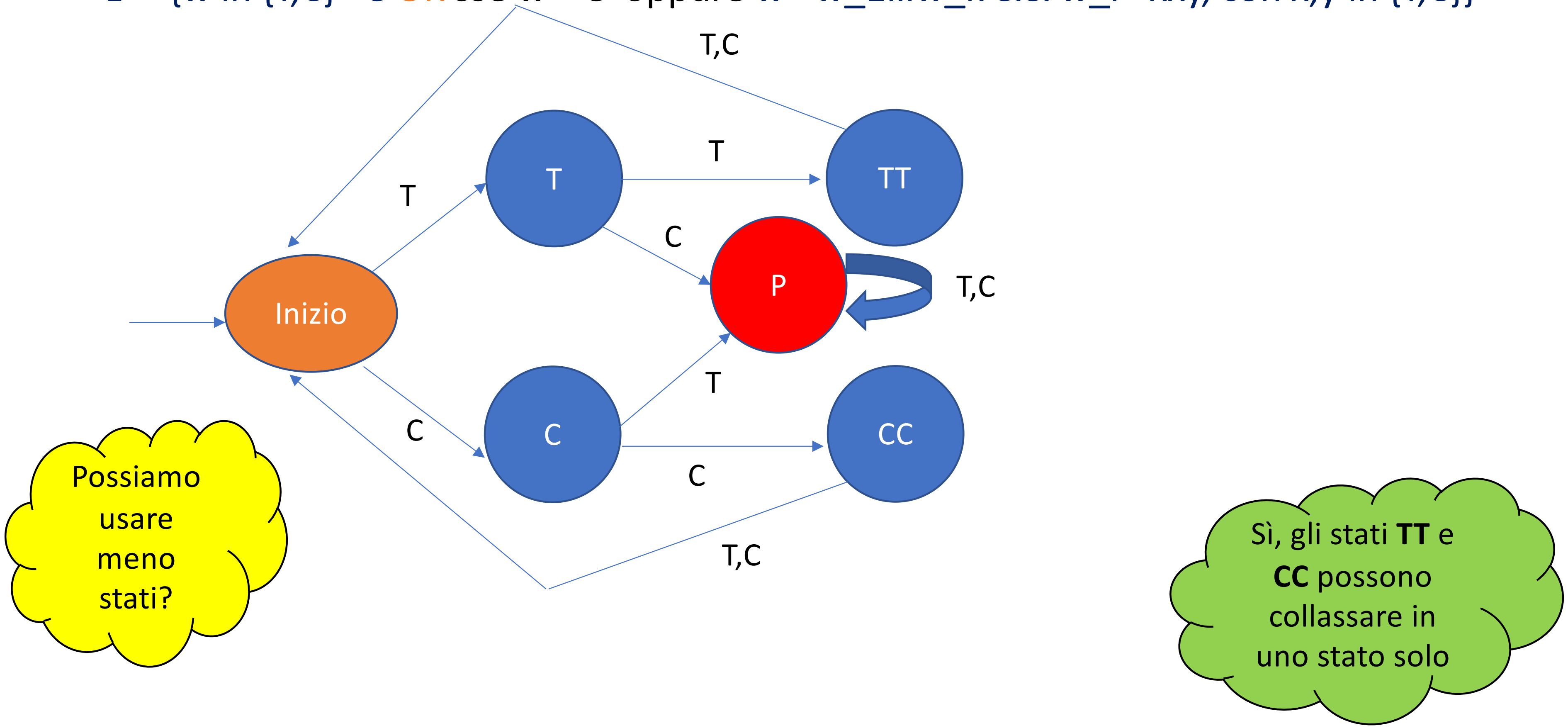
L' = {w in {T,C}* è OK sse w = ε oppure w =w_1…w_k && w_i =xxy, con x,y in {T,C}}



PROBLEMA 2

Trovare l'automa A che riconosce se una sequenza di triplette w abbia tutte le triplette in regola:

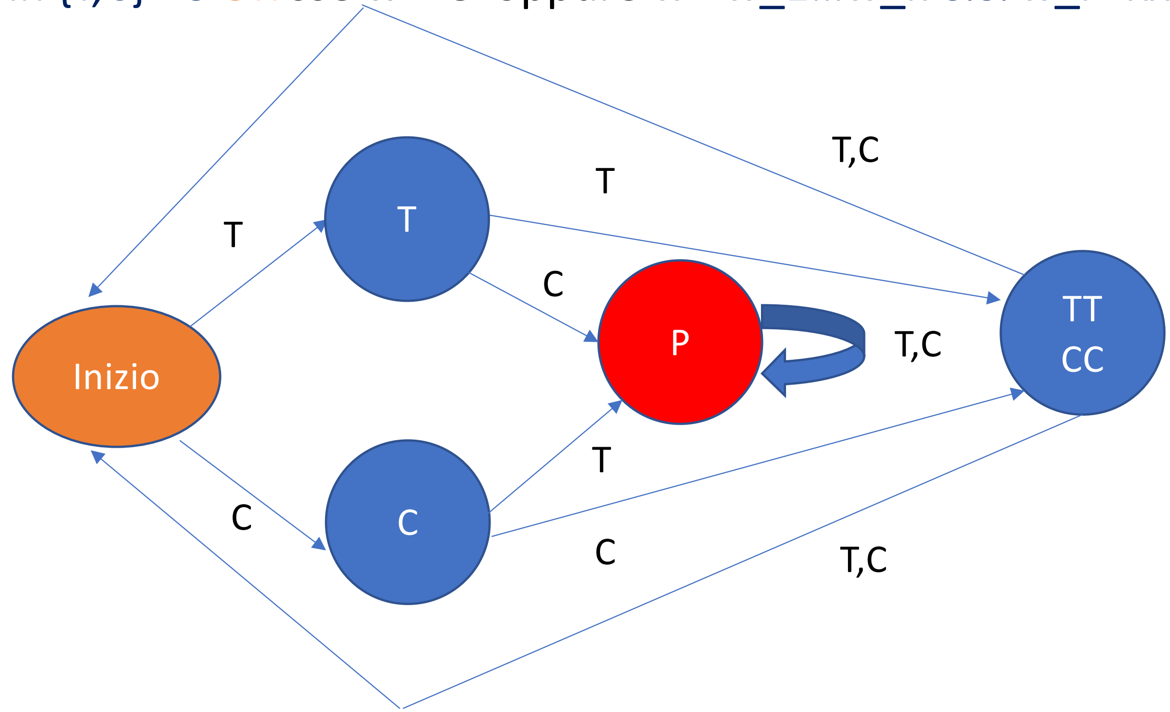
L' = {w in {T,C}* è OK sse w = ε oppure w =w_1…w_k && w_i =xxy, con x,y in {T,C}}



PROBLEMA 2

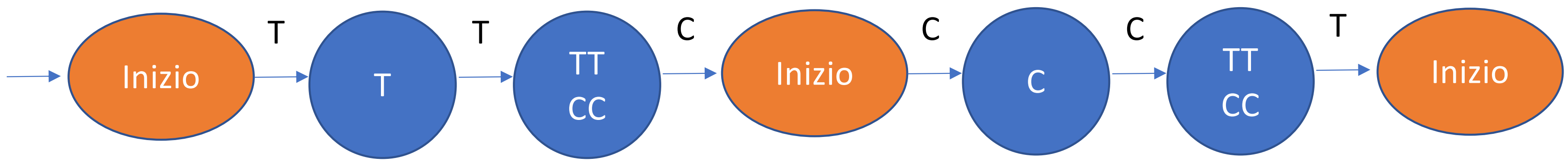
Trovare l’automa A che riconosce se una sequenza di triplette w abbia tutte le triplette in regola:

L’ = {w in {T,C}* è OK sse w = ε oppure w =w_1…w_k && w_i =xxy, con x,y in {T,C}}



Bastano meno stati

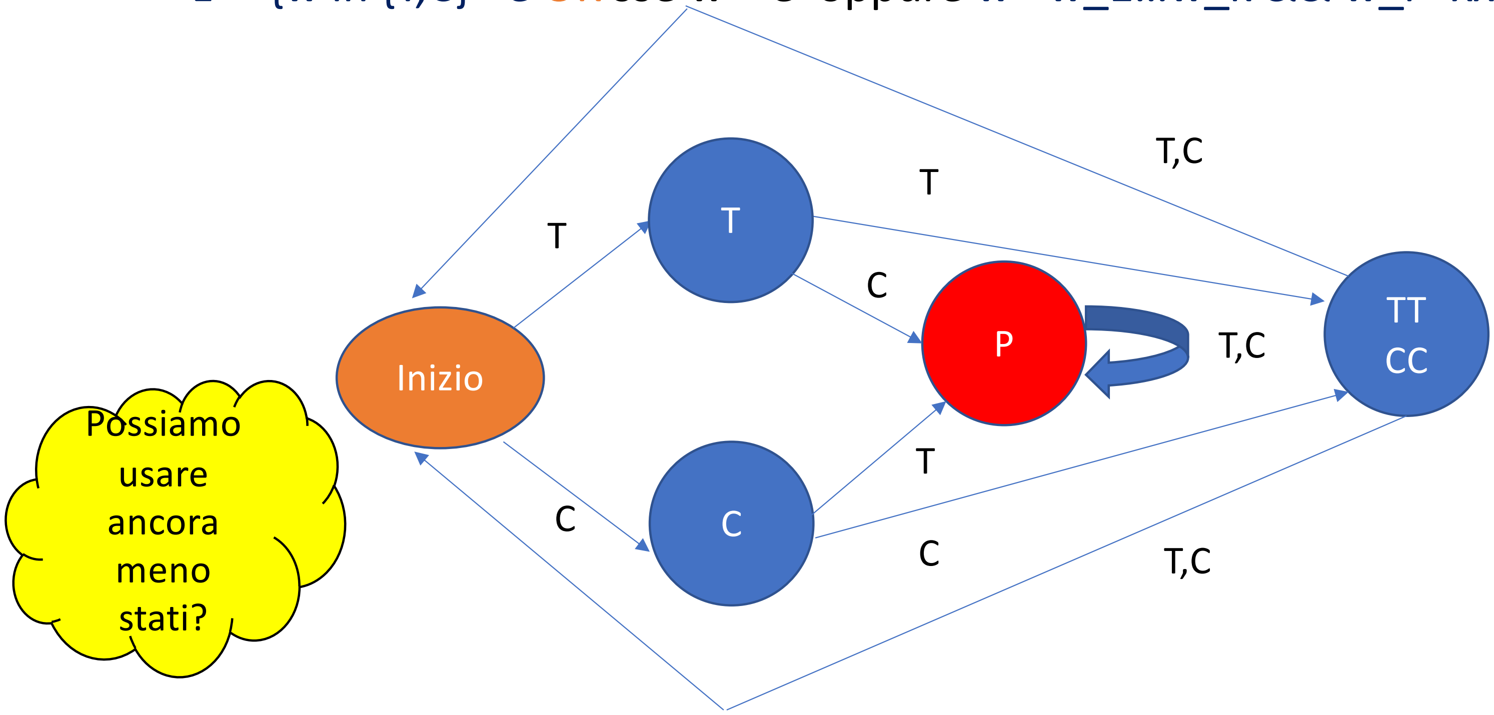
Sequenza accettata
TTC CCT



PROBLEMA 2

Trovare l'automa A che riconosce se una sequenza di triplette w abbia tutte le triplette in regola:

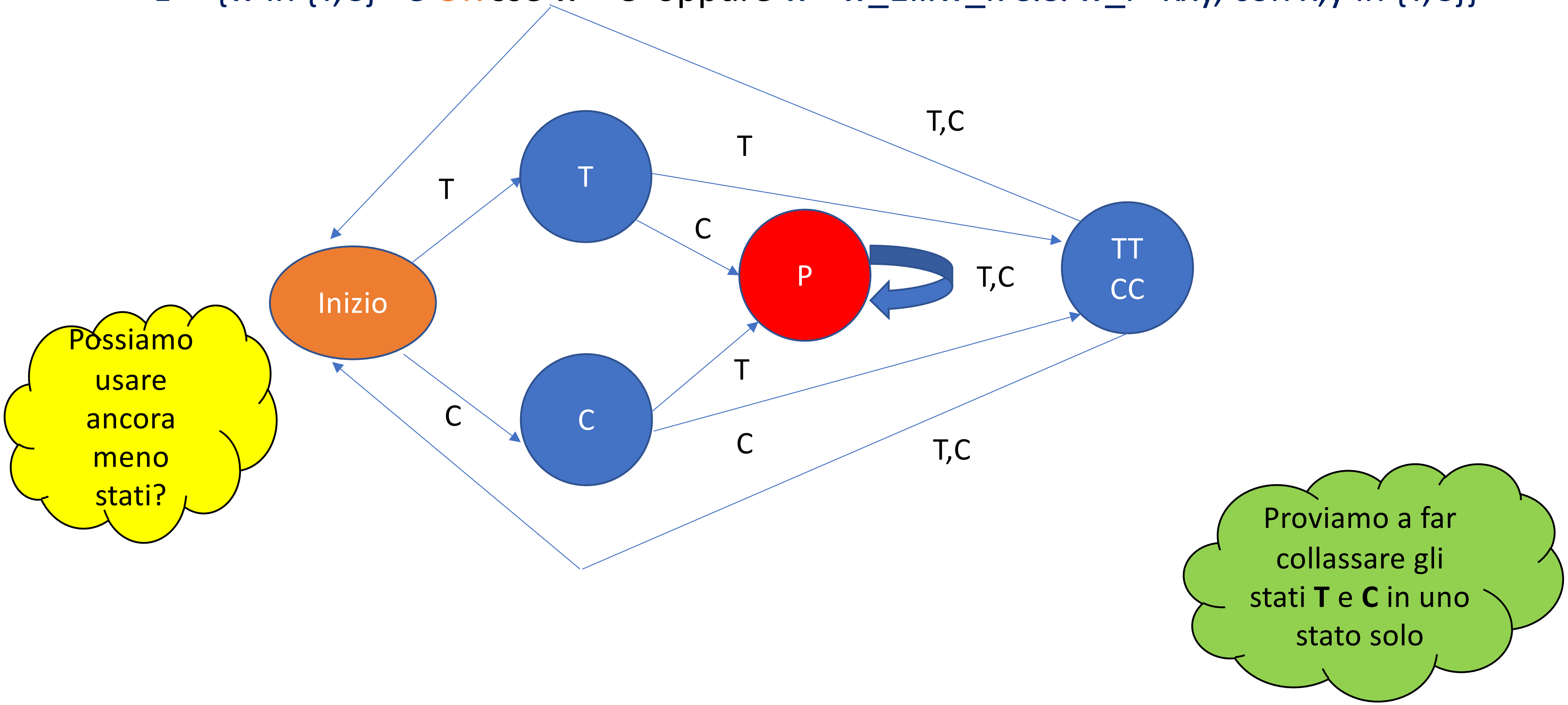
L' = {w in {T,C}* è OK sse w = ε oppure w =w_1…w_k && w_i =xxy, con x,y in {T,C}}



PROBLEMA 2

Trovare l'automa A che riconosce se una sequenza di triplette w abbia tutte le triplette in regola:

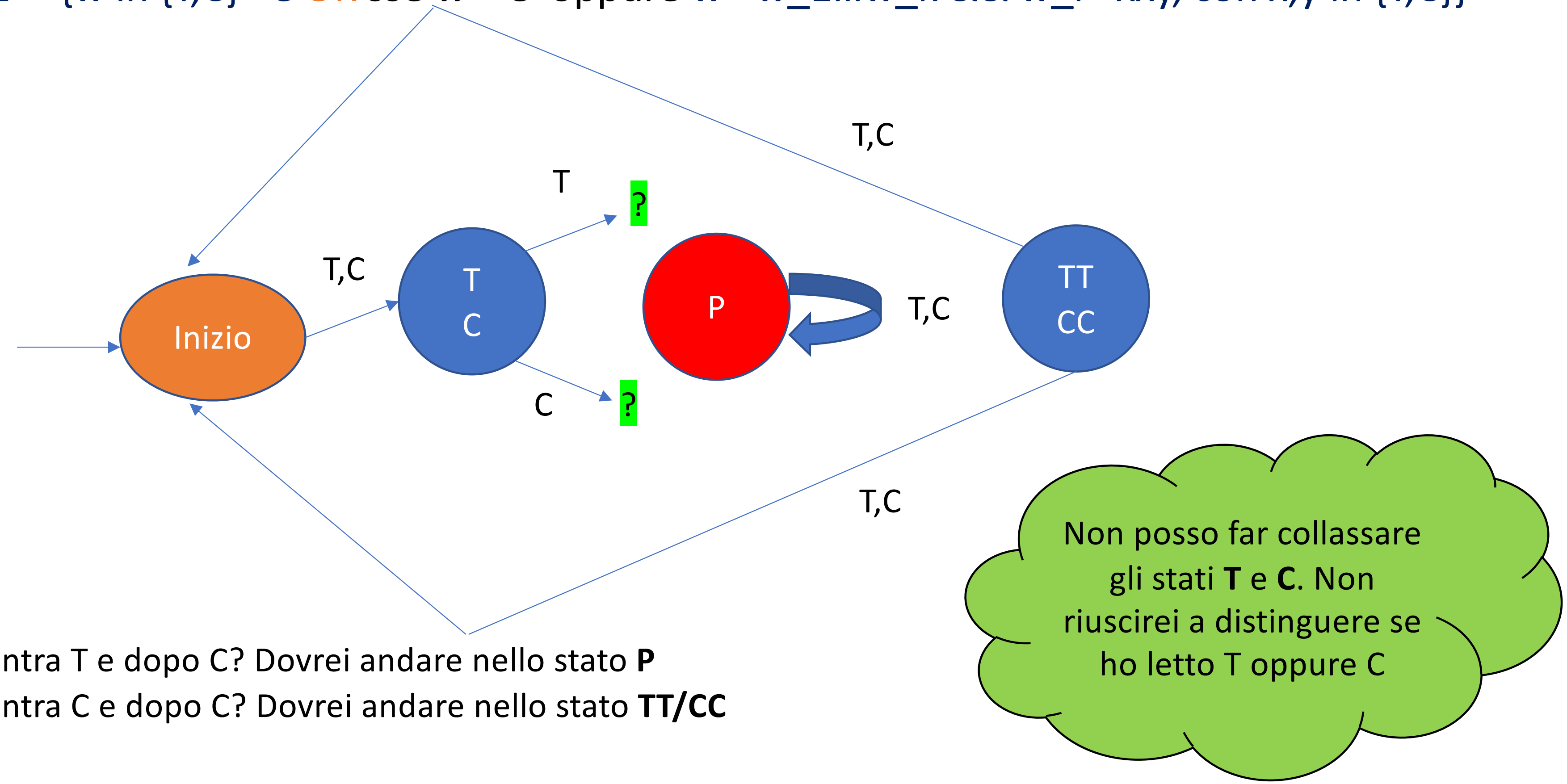
L' = {w in {T,C}* è OK sse w = ε oppure w =w_1…w_k && w_i =xxy, con x,y in {T,C}}



PROBLEMA 2

Trovare l'automa A che riconosce se una sequenza di triplette w abbia tutte le triplette in regola:

L' = {w in {T,C}* è OK sse w = ε oppure w =w_1…w_k && w_i =xxy, con x,y in {T,C}}

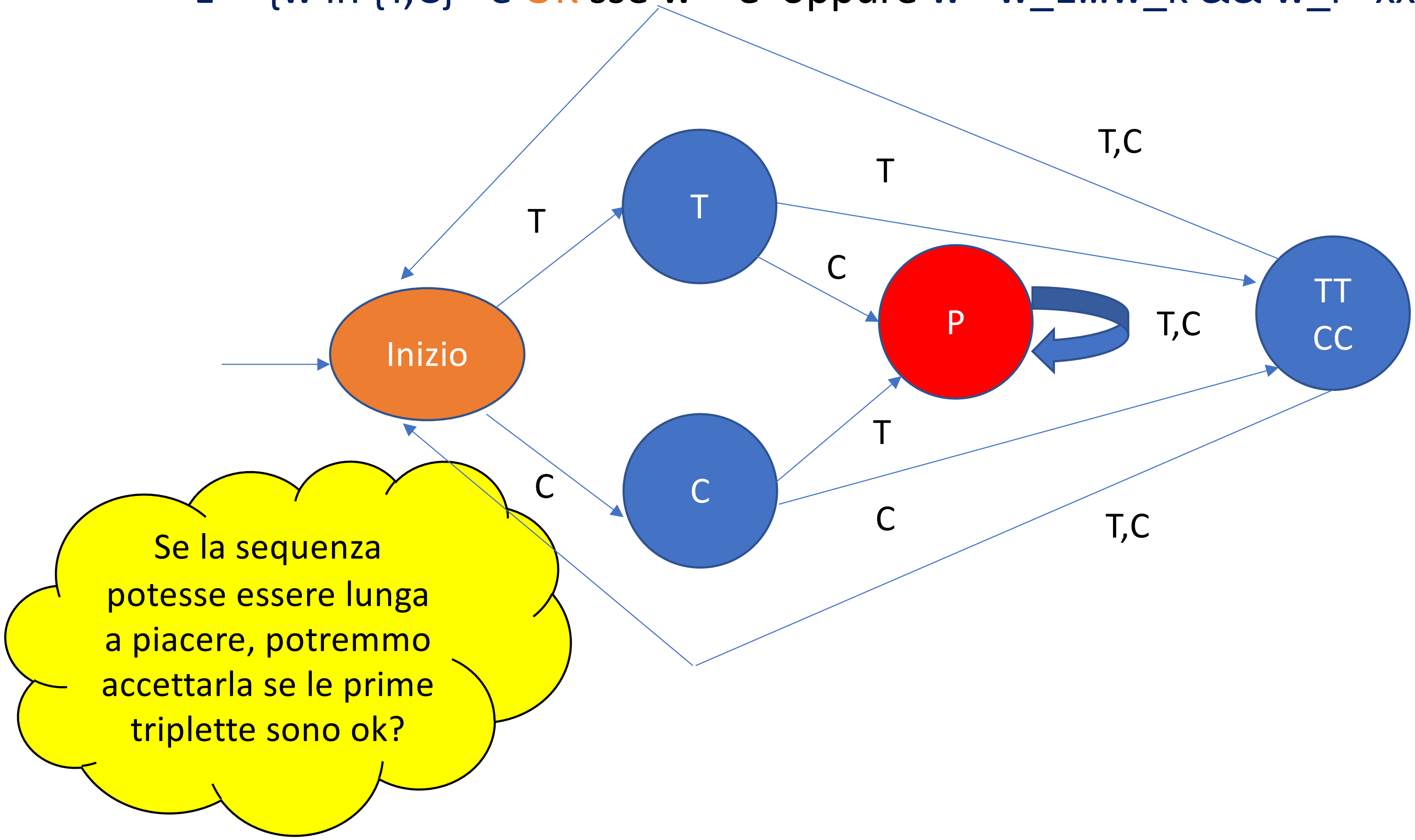


- Se entra T e dopo C? Dovrei andare nello stato **P**
- Se entra C e dopo C? Dovrei andare nello stato **TT/CC**

PROBLEMA 2

Trovare l'automa A che riconosce se una sequenza di triplette w abbia tutte le triplette in regola:

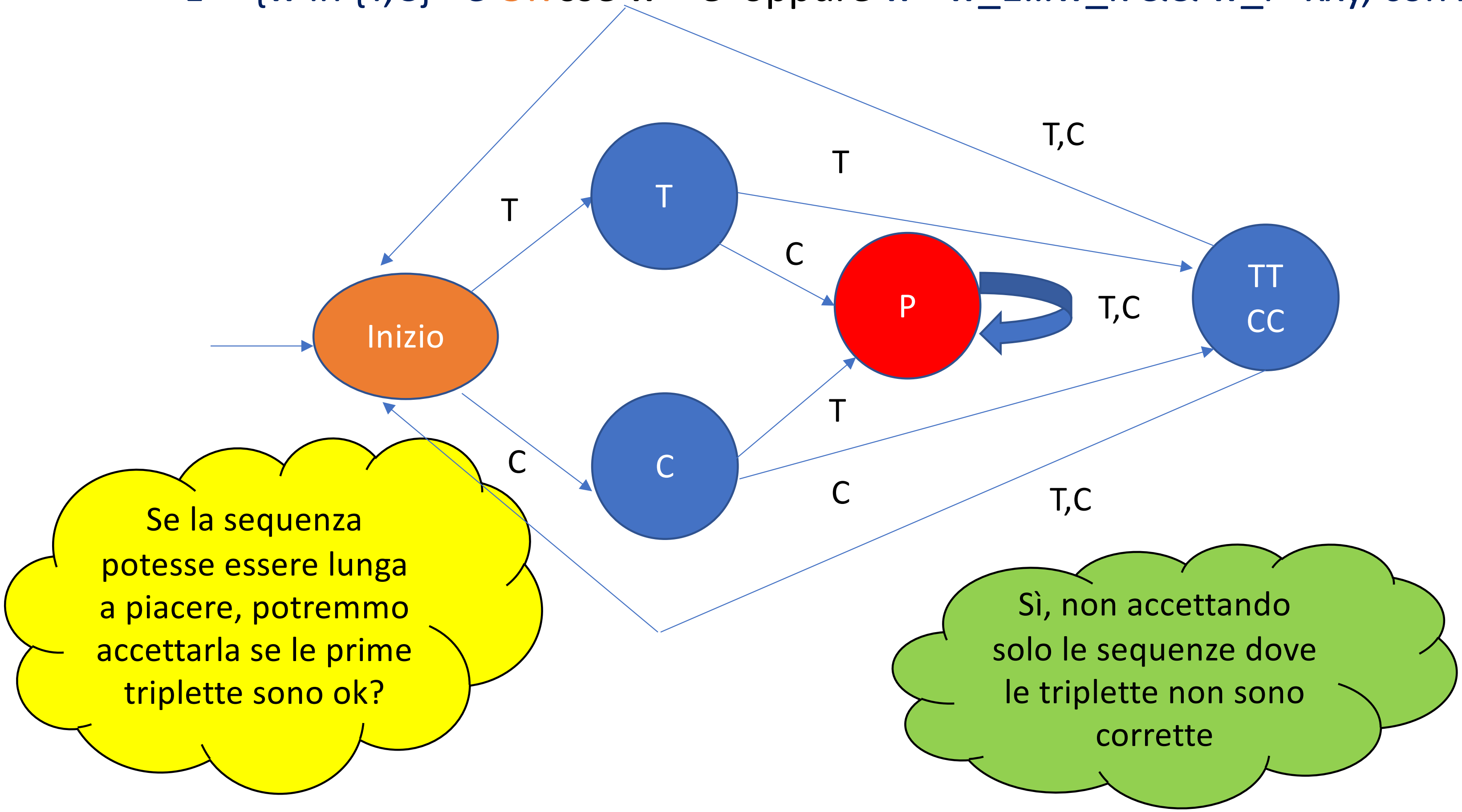
L' = {w in {T,C}* è OK sse w = ε oppure w =w_1…w_k && w_i =xxy, con x,y in {T,C}}



PROBLEMA 2

Trovare l’automa A che riconosce se una sequenza di triplette w abbia tutte le triplette in regola:

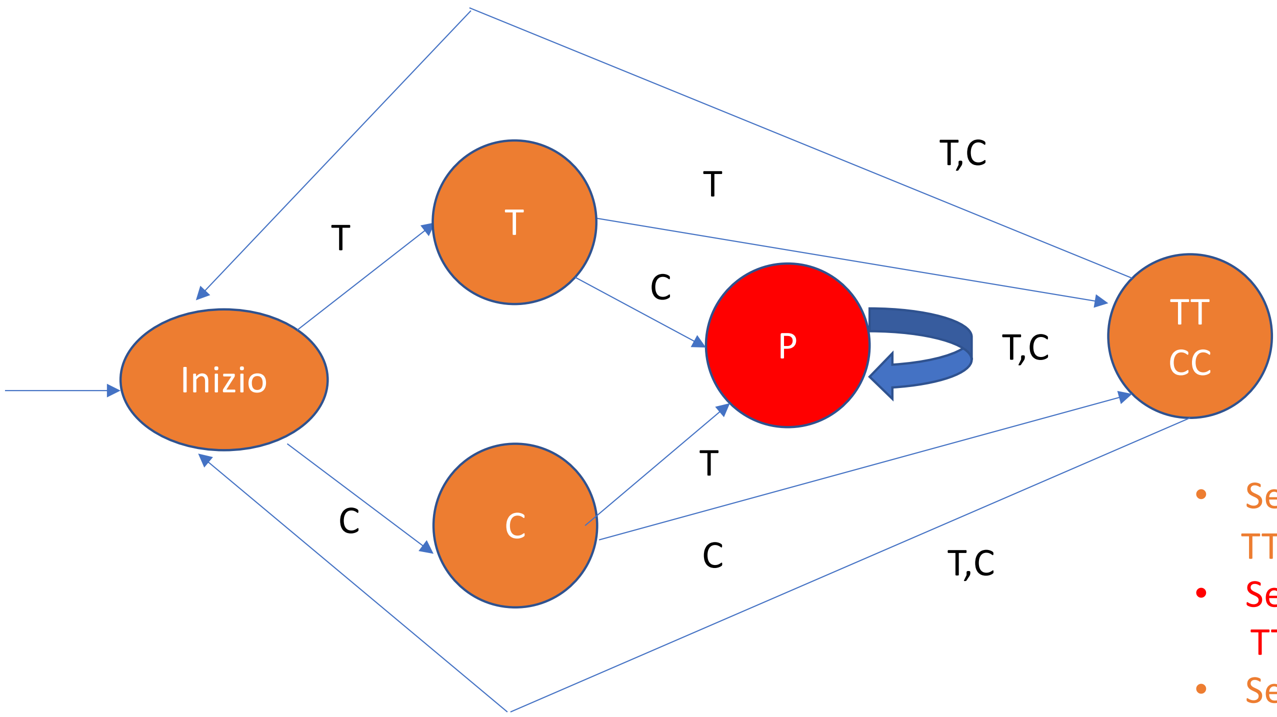
L’ = {w in {T,C}* è OK sse w = ε oppure w =w_1…w_k && w_i =xxy, con x,y in {T,C}}



PROBLEMA 3

Trovare l'automa A che riconosce se una sequenza di triplette w abbia tutte le triplette in regola:

L'' = {w in $\{T,C\}^*$ è OK sse w = ε oppure w =w_1…w_k z && w_i =xxy, con x,y in {T,C} e z in $\{T,C\}^0 \cup \{T,C\}^1 \cup \{T,C\}^2$}

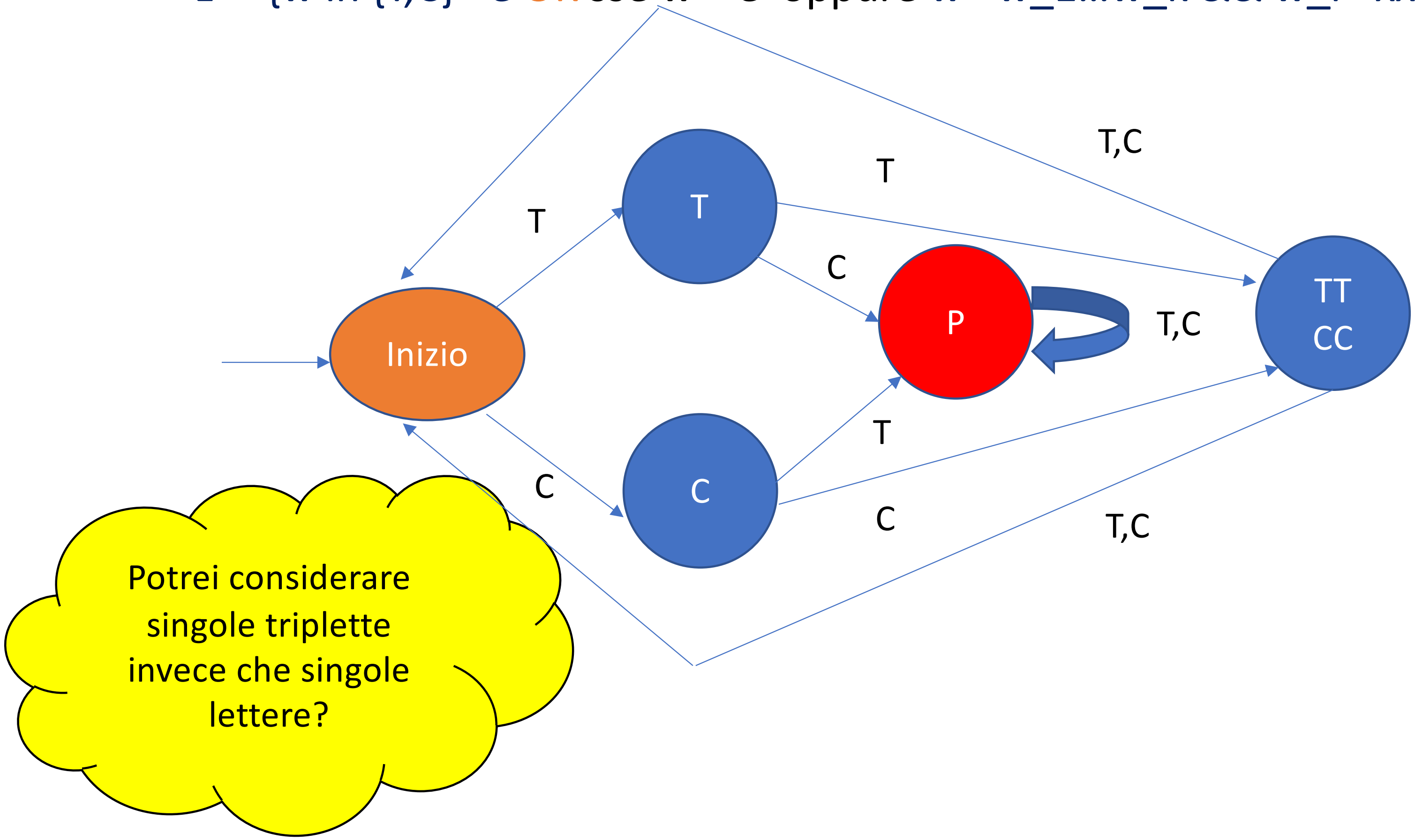


- Sequenza accettata
  TTC CCT TTC
- Sequenza non accettata
  TTC CTC TTC
- Sequenza ~~incompleta~~ accettata
  TTC CC

PROBLEMA 2

Trovare l'automa A che riconosce se una sequenza di triplette w abbia tutte le triplette in regola:

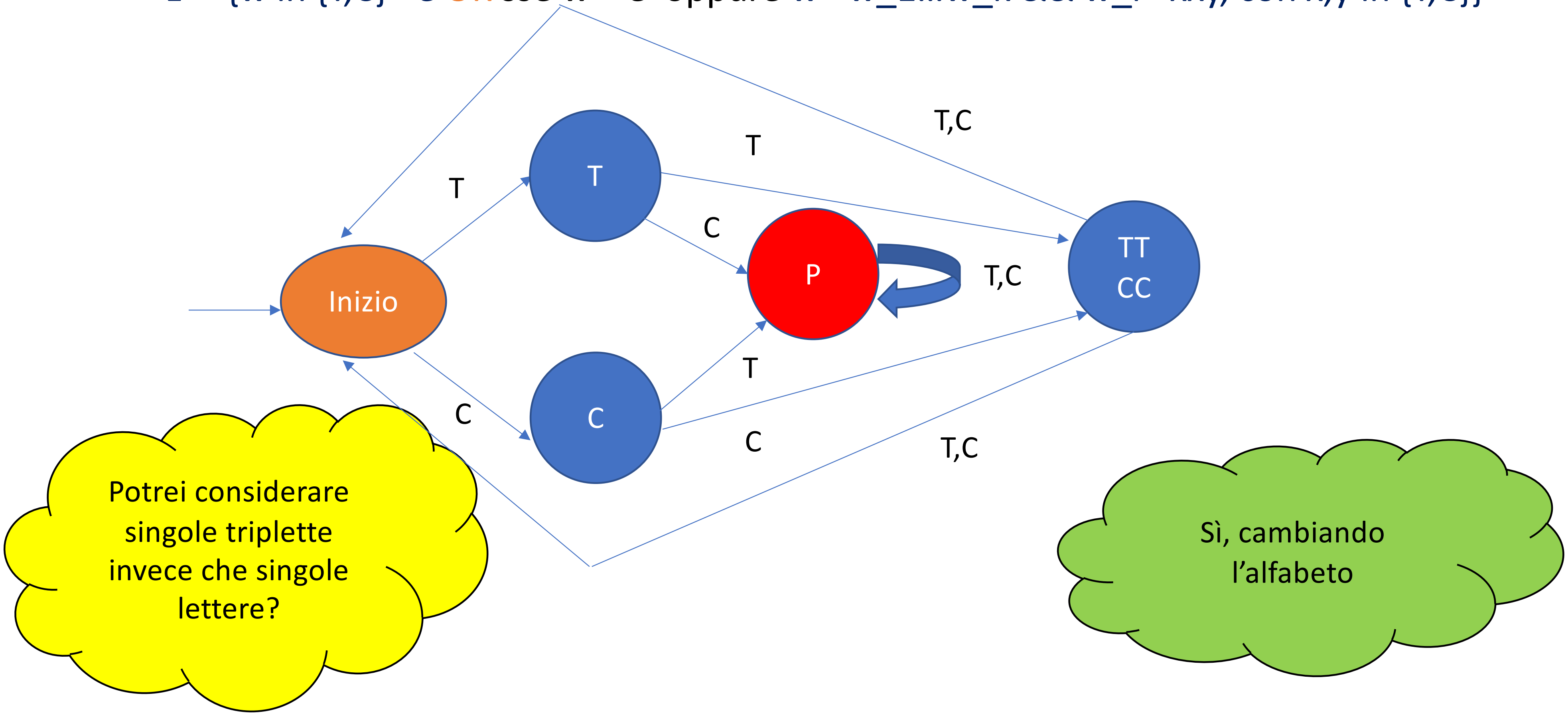
L' = {w in {T,C}* è OK sse w = ε oppure w =w_1…w_k && w_i =xxy, con x,y in {T,C}}



PROBLEMA 2

Trovare l'automa A che riconosce se una sequenza di triplette w abbia tutte le triplette in regola:

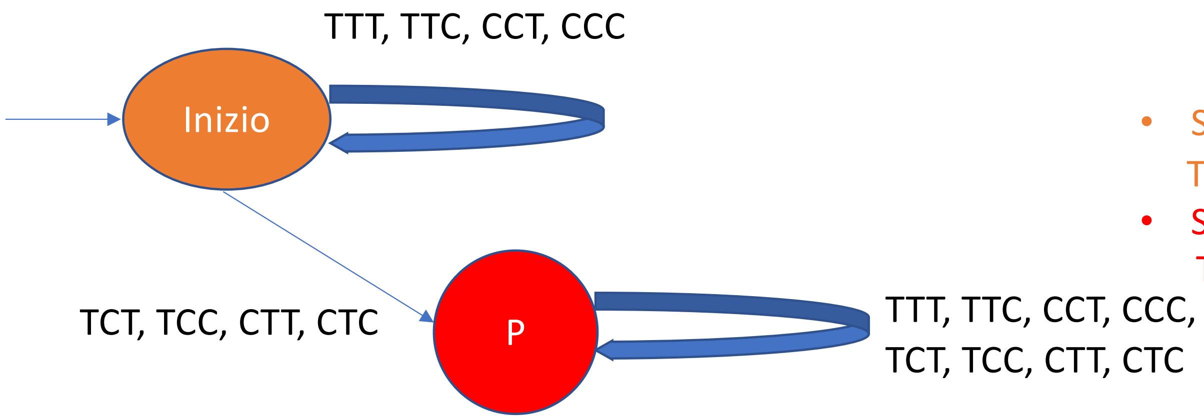
L' = {w in {T,C}* è OK sse w = ε oppure w =w_1…w_k && w_i =xxy, con x,y in {T,C}}



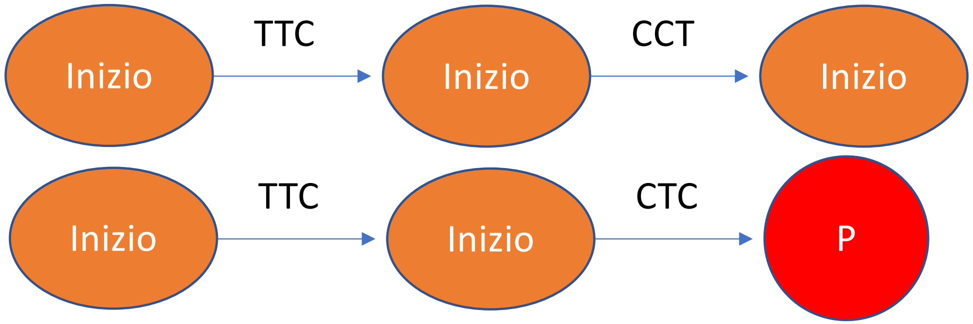
PROBLEMA 3

Trovare l’automa A che riconosce se una **sequenza di triplette** w abbia tutte le triplette in regola:

L’’ = {w in {TTT,TTC,TCT,TCC,CCC,CCT,CTC,CTT}* è OK sse w = ε oppure w in {TTT, TTC, CCT, CCC}}

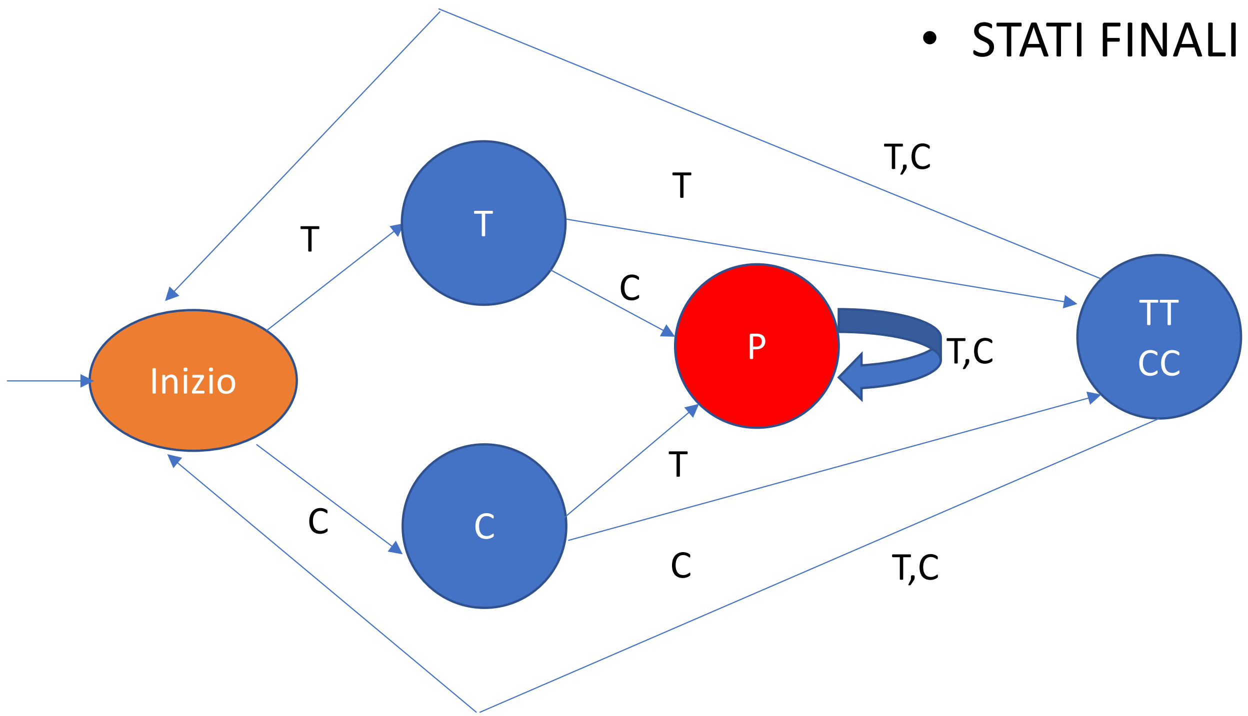


- Sequenza accettata
  TTC CCT
- Sequenza non accettata
  TTC CTC



COSA CI SERVE PER DEFINIRE UN AUTOMA?

- STATI
- ALFABETO FINITO di simboli
- FUNZIONE (stato,lettera) |--> stato’
- STATO Iniziale
- STATI FINALI

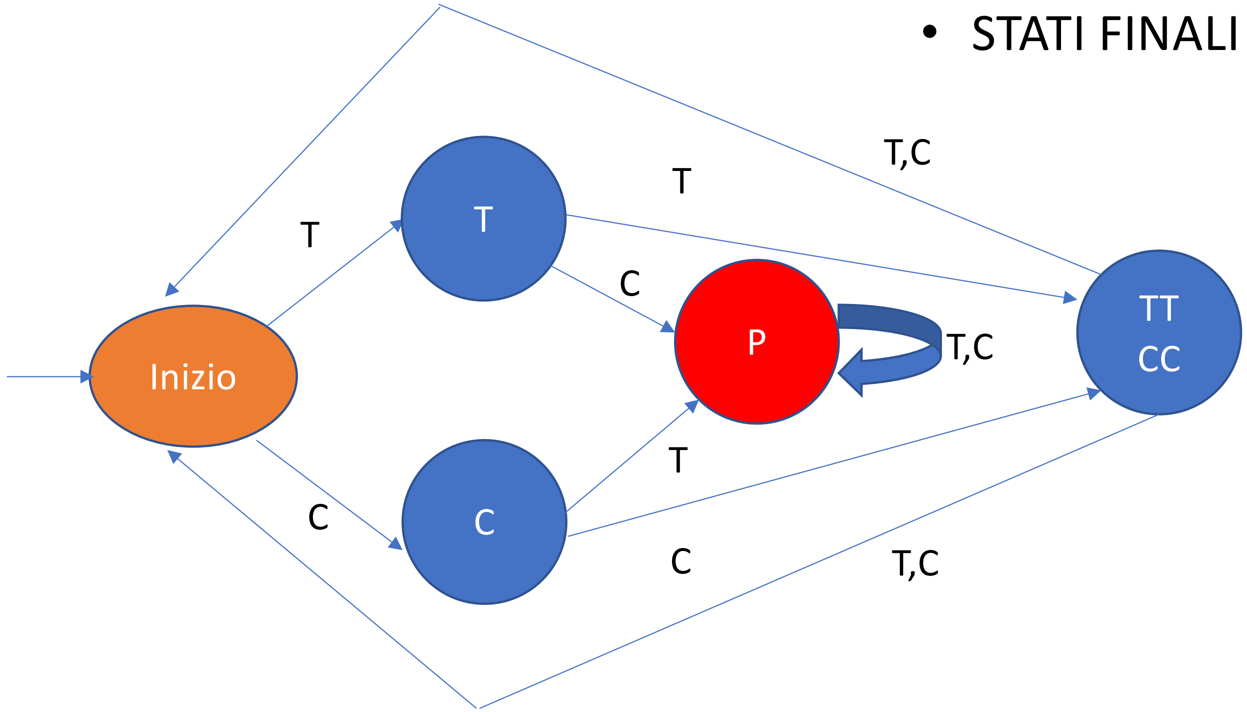


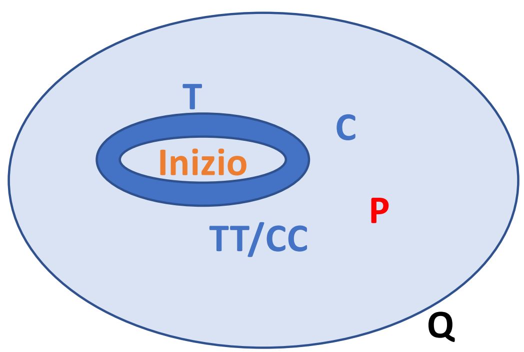
COSA CI SERVE PER DEFINIRE UN AUTOMA: $(Q, \Sigma, \delta, q_0, F)$

- STATI **Q** = {**Inizio**, **T,C,TT/CC,P**}
- ALFABETO FINITO **Σ** = {T,C}
- FUNZIONE: **δ:** Q x Σ → Q

$\delta(q, a) = q'$

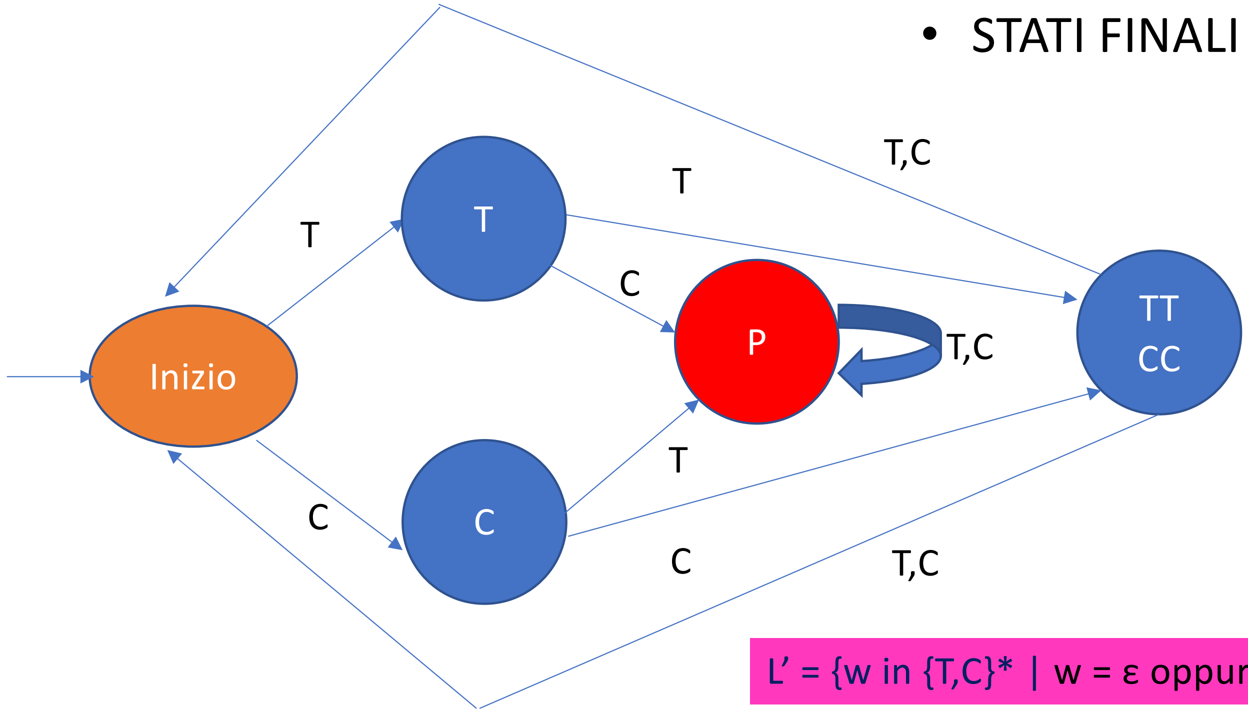
- STATO Iniziale $q_0 \in Q$ = Inizio
- STATI FINALI **F** $\subseteq$ Q= {Inizio}





COSA CI SERVE PER DEFINIRE UN AUTOMA: (Q, Σ, δ, q_0 , F)

- STATI **Q** = {**Inizio**, **T,C,TT/CC,P**}
- ALFABETO FINITO **Σ** = {T,C}
- FUNZIONE: **δ:** Q x Σ → Q (***)   δ(**q**, a) = **q**
- STATO Iniziale $\mathbf{q_0} \in Q$ = Inizio
- STATI FINALI **F** ⊆ Q= {Inizio}

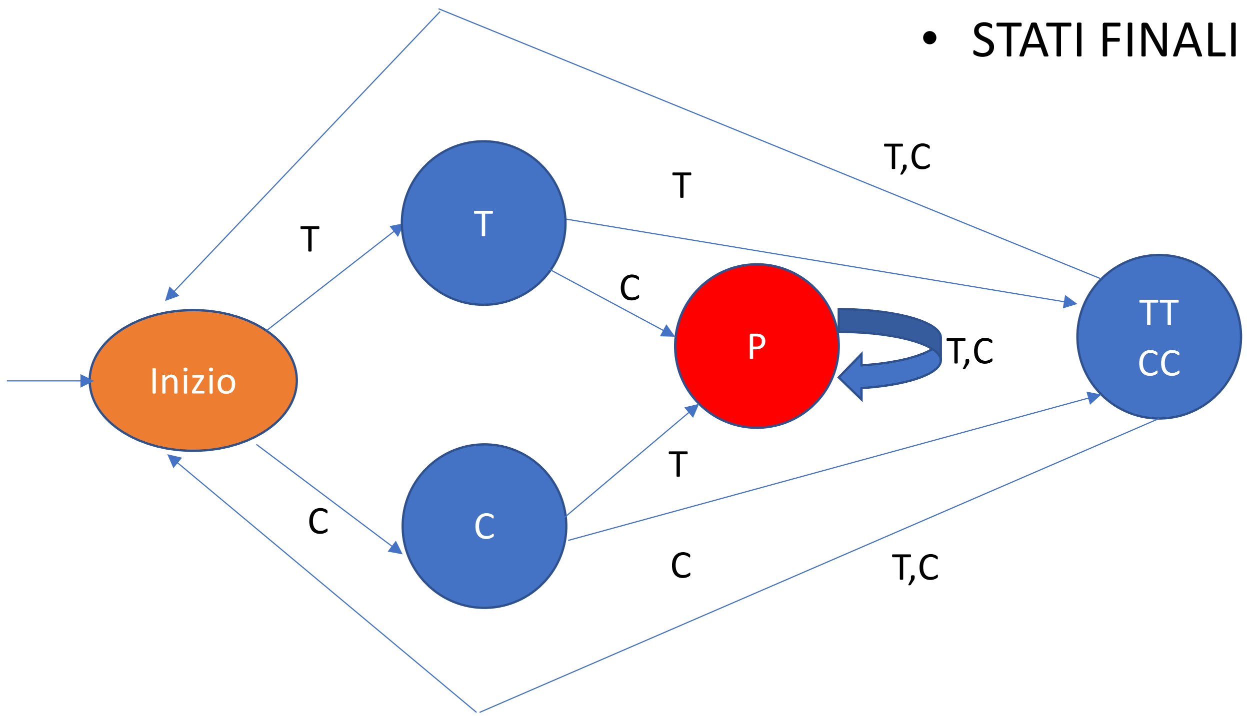


Gli automi riconoscono
linguaggi L ⊆ **Σ***
Nel nostro esempio il linguaggio L'
delle sequenze
di triplette corrette

L' = {w in {T,C}* | w = ε oppure w =w_1...w_k con w_i =xxy e x,y in {T,C}}

COSA CI SERVE PER DEFINIRE UN AUTOMA: (Q, Σ, δ, q_0 , F)

- STATI **Q** = {**Inizio**, **T,C,TT/CC,P**}
- ALFABETO FINITO **Σ** = {T,C}
- FUNZIONE: **δ:** Q x Σ → Q
- STATO Iniziale $\mathbf{q_0} \in Q$ = Inizio
- STATI FINALI **F** ⊆ Q= {Inizio}

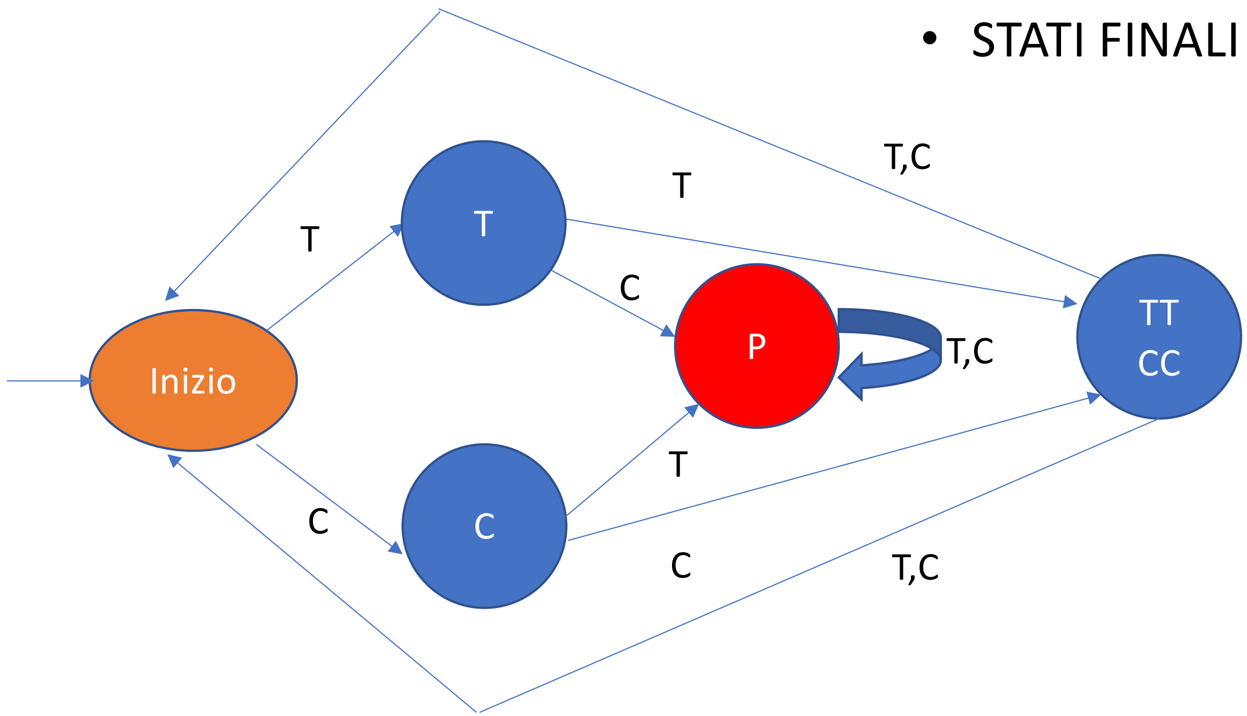


δ: Q x Σ → Q **FUNZIONE DI TRANSIZIONE**

- δ(**Inizio**, T) = **T**
- δ(**Inizio**, C) = **C**
- δ(**T**, T) = **TT/CC**
- δ(**T**, C) = **P**
- δ(**C**, C) = **TT/CC**
- δ(**C**, T) = **P**
- δ(**TT/CC**, T) = **Inizio**
- δ(**TT/CC**, C) = **Inizio**
- δ(**P**, C) = **P**
- δ(**P**, T) = **P**

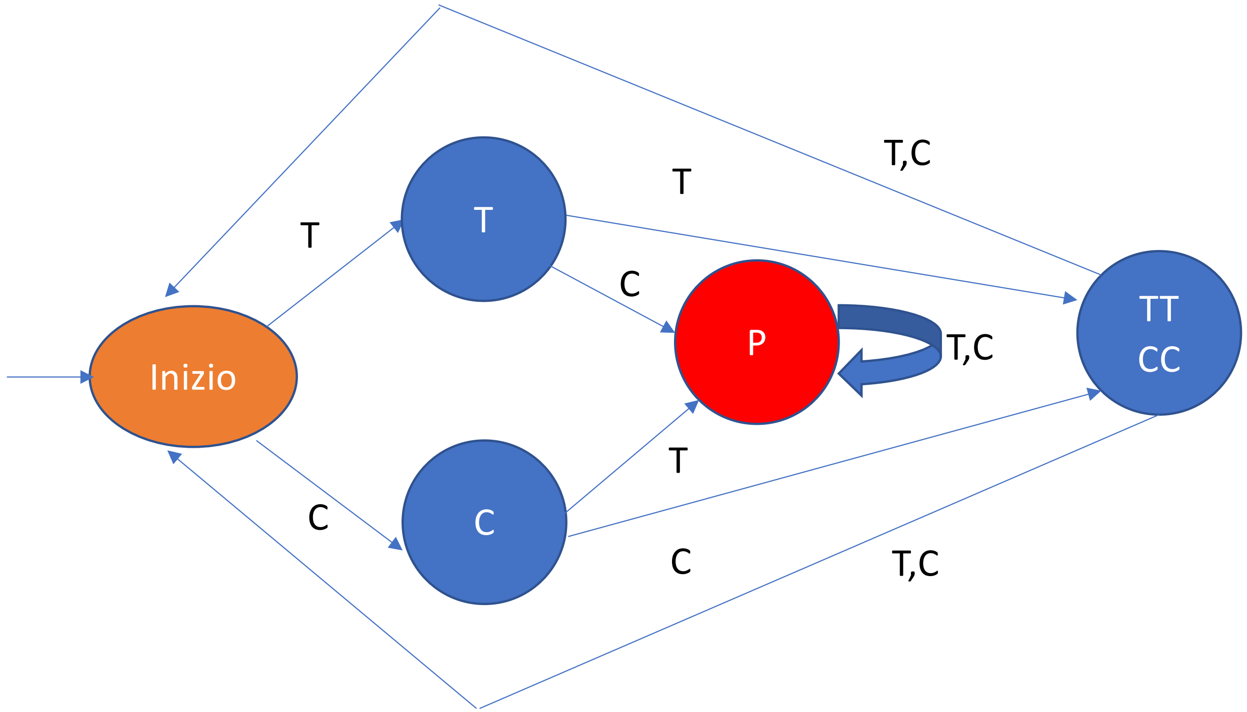
COSA CI SERVE PER DEFINIRE UN AUTOMA: (Q, Σ, δ, q_0 , F)

- STATI **Q** = {**Inizio**, **T,C,TT/CC,P**}
- ALFABETO FINITO **Σ** = {T,C}
- FUNZIONE: **δ:** Q x Σ → Q
- STATO Iniziale $\mathbf{q_0} \in Q$ = Inizio
- STATI FINALI **F** ⊆ Q= {Inizio}



δ	T	C
* → Inizio	T	C
T	TT/CC	P
C	P	TT/CC
TT/CC	Inizio	Inizio
P	P	P

FUNZIONE DI TRANSIZIONE ESTESA: definizione induttiva

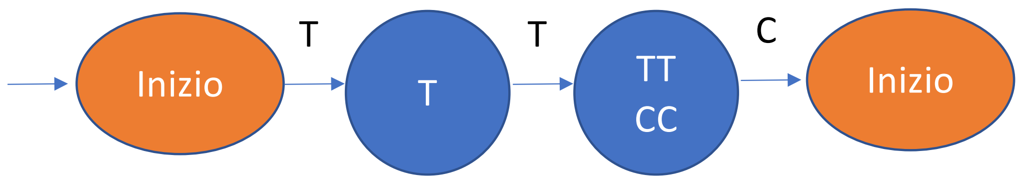


$\hat{\delta}$: $Q \times \Sigma^* \rightarrow Q$

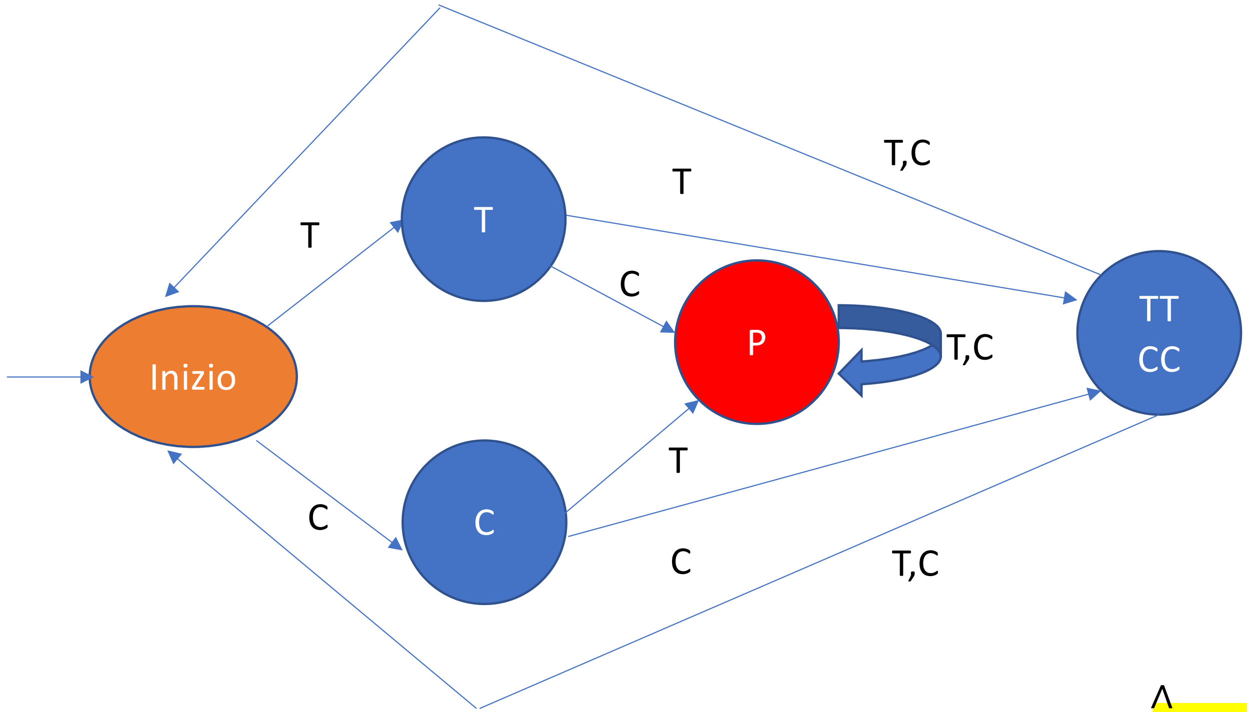
Base: $\hat{\delta}(\mathbf{q}, \varepsilon) = \mathbf{q}$

Induzione: $\hat{\delta}(\mathbf{q}, xa) = \delta(\hat{\delta}(\mathbf{q}, x), a)$
x in Σ^*, a in Σ

- La funzione di transizione estesa $\hat{\delta}$ opera su stati e stringhe (invece che su stati e simboli)
- $\hat{\delta}(q0,w)$ = p denota che p è lo stato raggiunto a partire da q0 quando si riceve in input uno a uno i simboli di w



FUNZIONE DI TRANSIZIONE ESTESA: definizione induttiva



$\hat{\delta}$**:** $Q \times \Sigma^* \rightarrow Q$

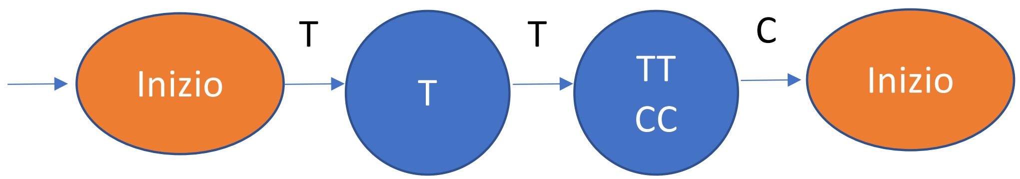
Base: $\hat{\delta}(\mathbf{q}, \varepsilon) = \mathbf{q}$

Induzione: $\hat{\delta}(\mathbf{q}, xa) = \delta(\hat{\delta}(\mathbf{q}, x), a)$

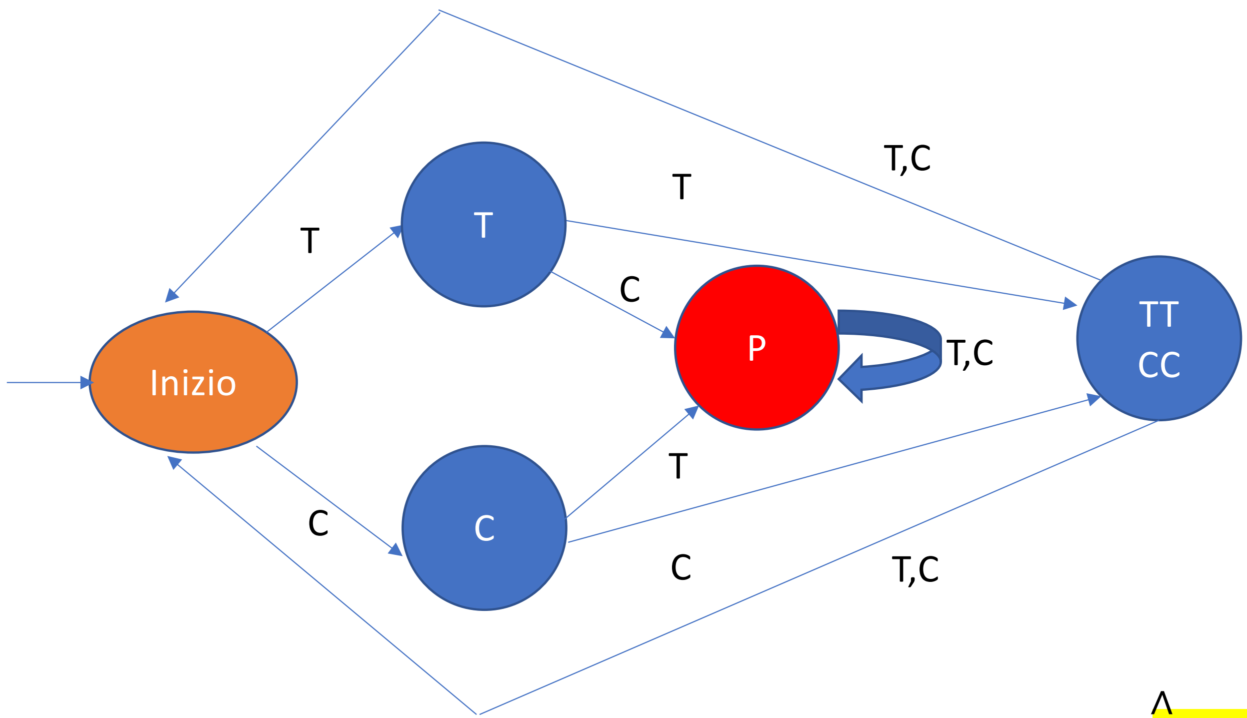
x in Σ^*, a in Σ

- La funzione di transizione estesa $\hat{\delta}$ opera su stati e stringhe (invece che su stati e simboli)
- $\hat{\delta}(q0,w) = p$ denota che p è lo stato raggiunto a partire da q0 quando si riceve in input uno a uno i simboli di w

$\hat{\delta}$(Inizio, TTC) =



FUNZIONE DI TRANSIZIONE ESTESA: definizione induttiva



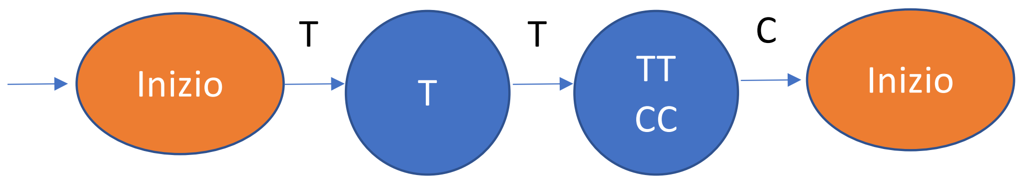
$\hat{\delta}$**:** Q x Σ* → Q

Base: $\hat{\delta}$(**q**, ε) = **q**

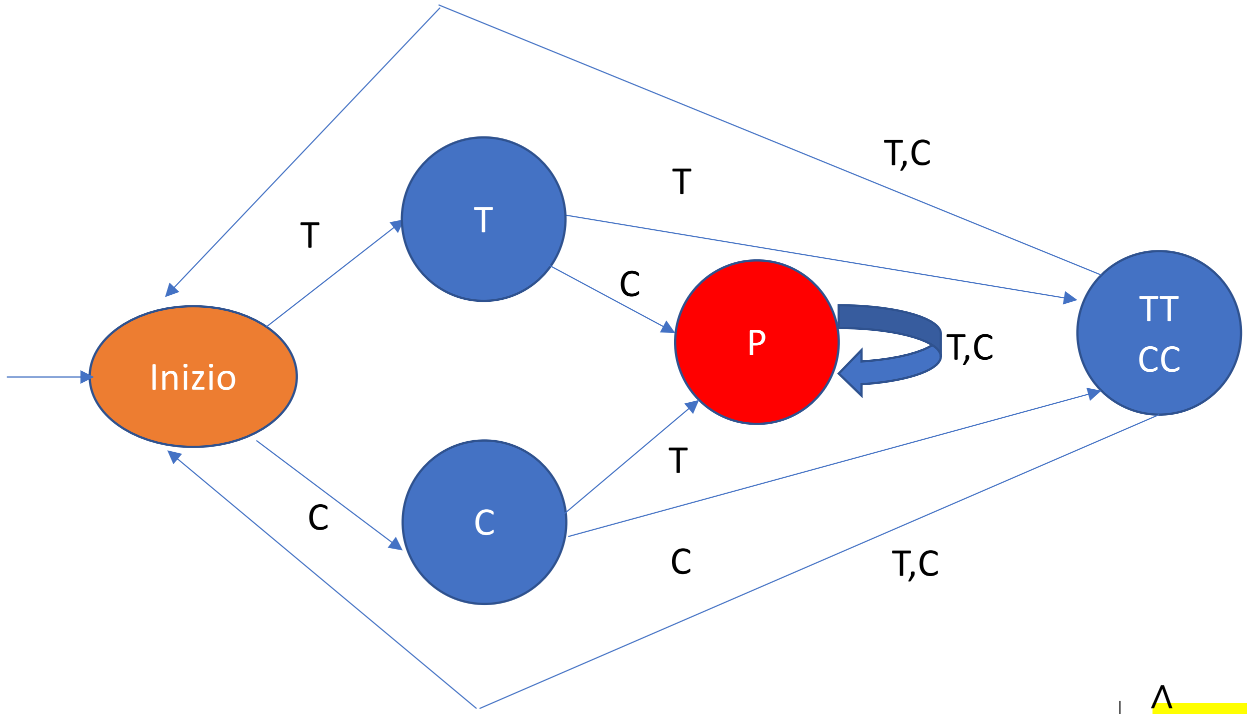
Induzione: $\hat{\delta}$(**q**, xa) = δ($\hat{\delta}$(**q**, x),a)
x in Σ*, a in Σ

- La funzione di transizione estesa $\hat{\delta}$ opera su stati e stringhe (invece che su stati e simboli)
- $\hat{\delta}$(q0,w) = p denota che p è lo stato raggiunto a partire da q0 quando si riceve in input uno a uno i simboli di w

$\hat{\delta}$(Inizio, TTC) = δ($\hat{\delta}$(Inizio, TT),C)



FUNZIONE DI TRANSIZIONE ESTESA: definizione induttiva



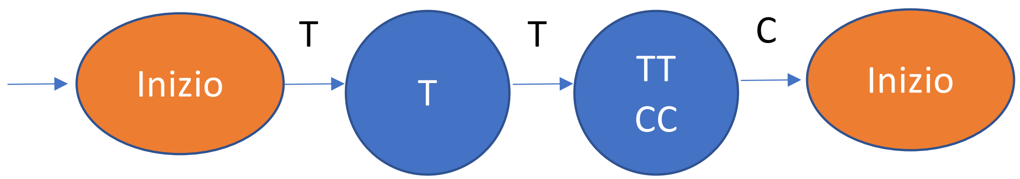
$\hat{\boldsymbol{\delta}}$**:** Q x Σ* → Q

Base: $\hat{\delta}(\mathbf{q}, \varepsilon) = \mathbf{q}$

Induzione: $\hat{\delta}(\mathbf{q}, xa) = \delta(\hat{\delta}(\mathbf{q}, x),a)$

x in Σ*, a in Σ

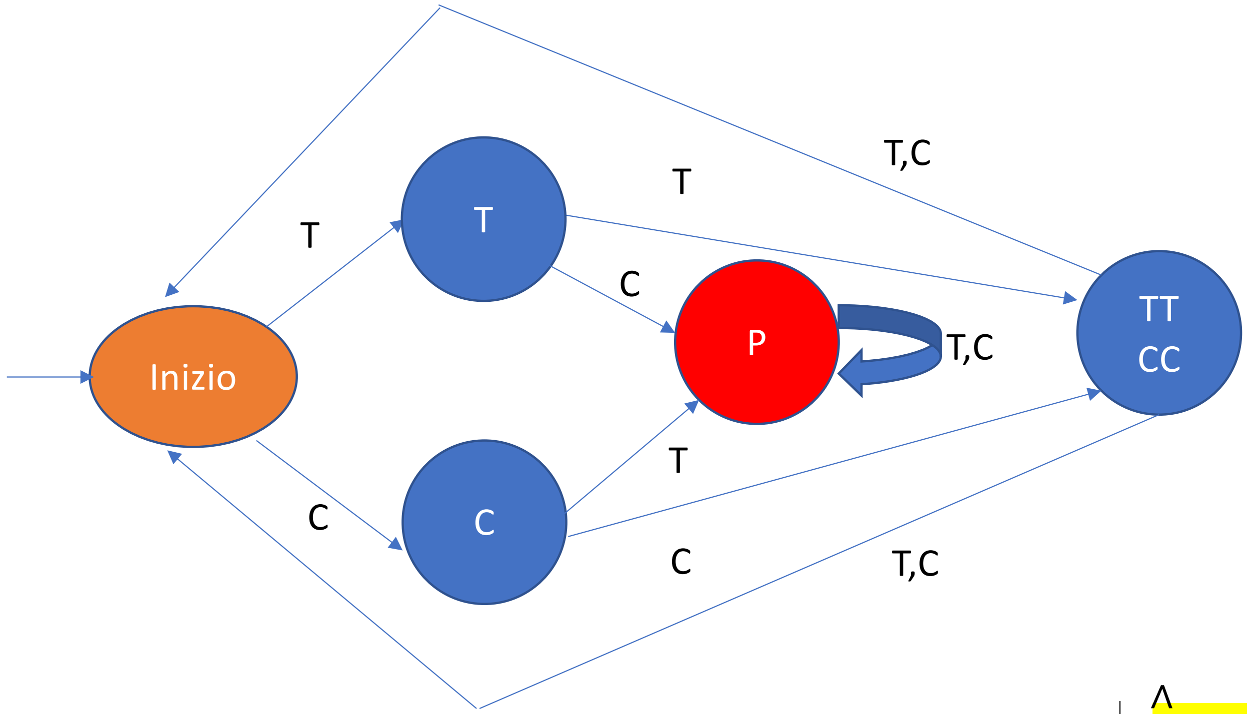
- La funzione di transizione estesa $\hat{\delta}$ opera su stati e stringhe (invece che su stati e simboli)
- $\hat{\delta}(q0,w) = p$ denota che p è lo stato raggiunto a partire da q0 quando si riceve in input uno a uno i simboli di w



$\hat{\delta}$(Inizio, TTC) = $\delta(\hat{\delta}$(Inizio, TT),C)

$\hat{\delta}$(Inizio, TT) =

FUNZIONE DI TRANSIZIONE ESTESA: definizione induttiva



$\hat{\delta}$**:** Q x Σ* → Q

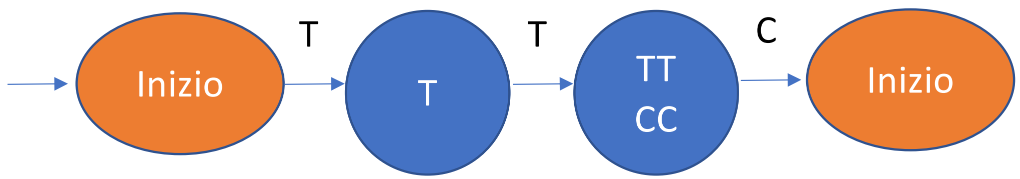
Base: $\hat{\delta}(\mathbf{q}, \varepsilon) = \mathbf{q}$

Induzione: $\hat{\delta}(\mathbf{q}, xa) = \delta(\hat{\delta}(\mathbf{q}, x), a)$
x in Σ*, a in Σ

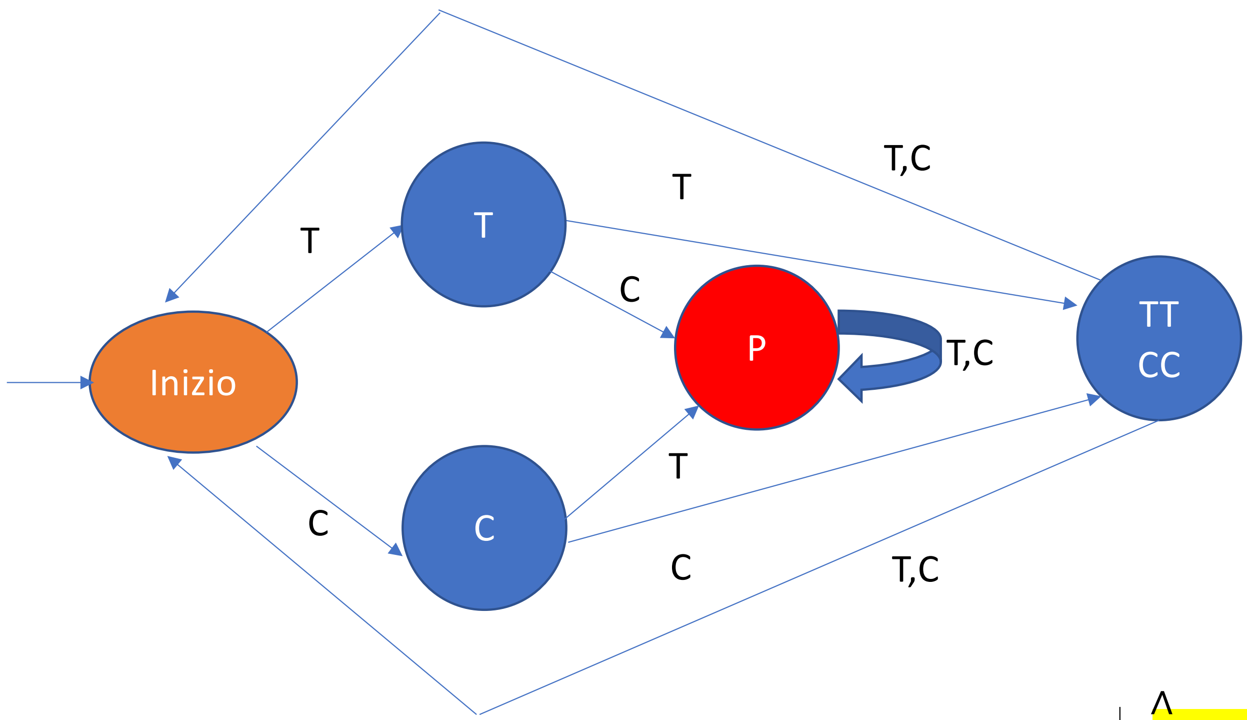
- La funzione di transizione estesa $\hat{\delta}$ opera su stati e stringhe (invece che su stati e simboli)
- $\hat{\delta}(q0,w)$ = p denota che p è lo stato raggiunto a partire da q0 quando si riceve in input uno a uno i simboli di w

$\hat{\delta}$(Inizio, TTC) = $\delta(\hat{\delta}$(Inizio, TT),C)

$\hat{\delta}$(Inizio, TT) = $\delta(\hat{\delta}$(Inizio, T),T)



FUNZIONE DI TRANSIZIONE ESTESA: definizione induttiva



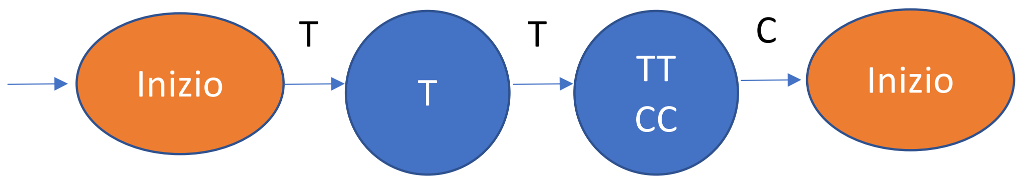
$\hat{\delta}$: $Q \times \Sigma^* \rightarrow Q$

Base: $\hat{\delta}(\mathbf{q}, \varepsilon) = \mathbf{q}$

Induzione: $\hat{\delta}(\mathbf{q}, xa) = \delta(\hat{\delta}(\mathbf{q}, x), a)$

x in Σ^*, a in Σ

- La funzione di transizione estesa $\hat{\delta}$ opera su stati e stringhe (invece che su stati e simboli)
- $\hat{\delta}(q0,w) = p$ denota che p è lo stato raggiunto a partire da q0 quando si riceve in input uno a uno i simboli di w

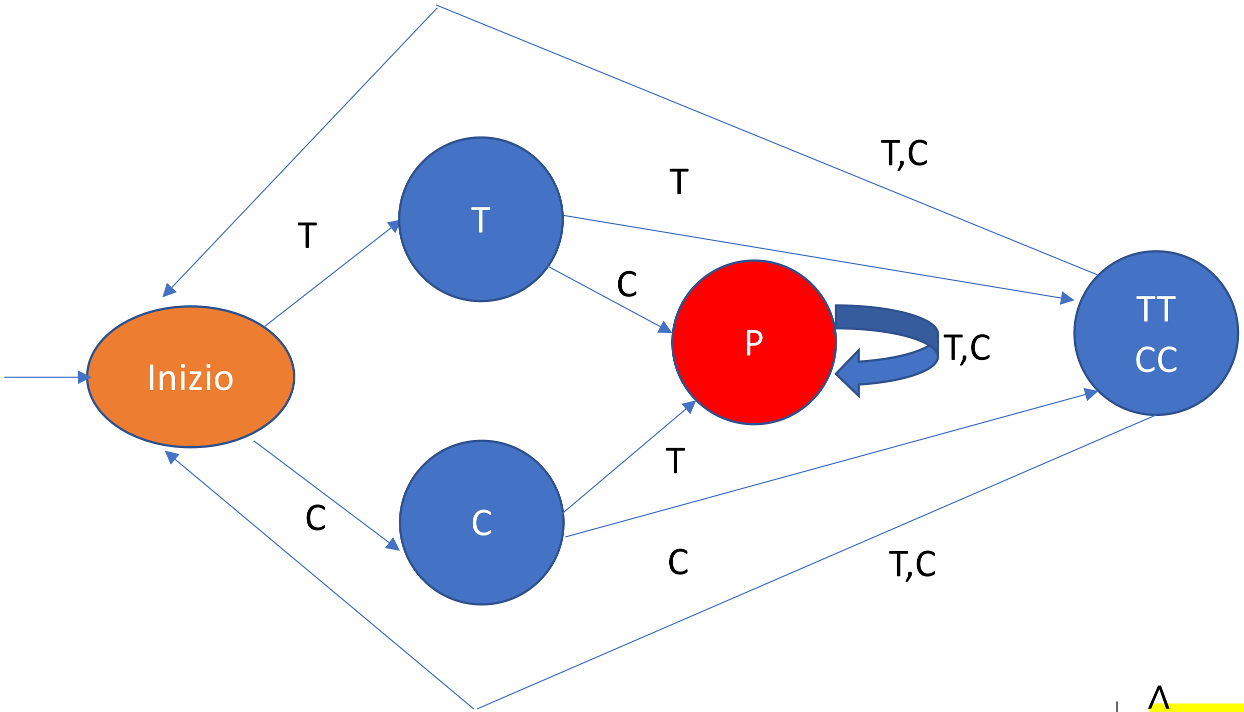


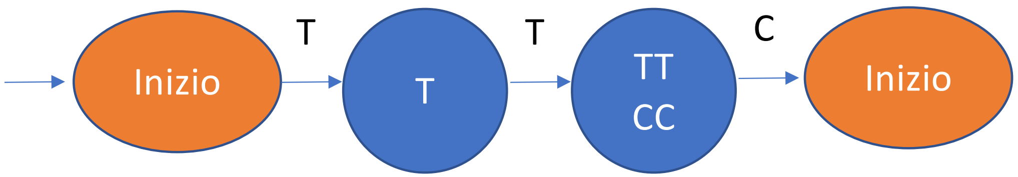
$\hat{\delta}$(Inizio, TTC) = $\delta(\hat{\delta}$(Inizio, TT),C)

$\hat{\delta}$(Inizio, TT) = $\delta(\hat{\delta}$(Inizio, T),T)

$\hat{\delta}$(Inizio, T) =

FUNZIONE DI TRANSIZIONE ESTESA: definizione induttiva





$\hat{\delta}$**:** $Q \times \Sigma^* \rightarrow Q$

Base: $\hat{\delta}(\mathbf{q}, \varepsilon) = \mathbf{q}$

Induzione: $\hat{\delta}(\mathbf{q}, xa) = \delta(\hat{\delta}(\mathbf{q}, x), a)$

x in Σ^*, a in Σ

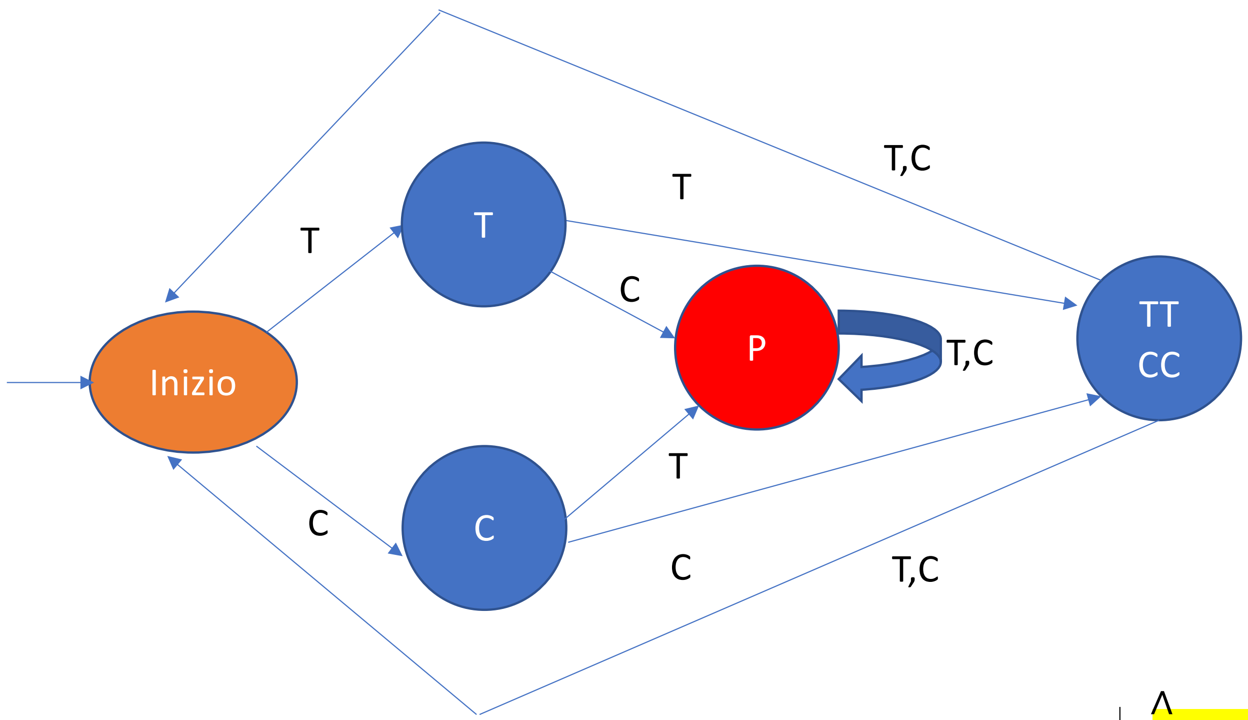
- La funzione di transizione estesa $\hat{\delta}$ opera su stati e stringhe (invece che su stati e simboli)
- $\hat{\delta}(q0,w) = p$ denota che p è lo stato raggiunto a partire da q0 quando si riceve in input uno a uno i simboli di w

$\hat{\delta}(\text{Inizio}, TTC) = \delta(\hat{\delta}(\text{Inizio}, TT), C)$

$\hat{\delta}(\text{Inizio}, TT) = \delta(\hat{\delta}(\text{Inizio}, T), T)$

$\hat{\delta}(\text{Inizio}, T) = \delta(\hat{\delta}(\text{Inizio}, \varepsilon), T)$

FUNZIONE DI TRANSIZIONE ESTESA: definizione induttiva



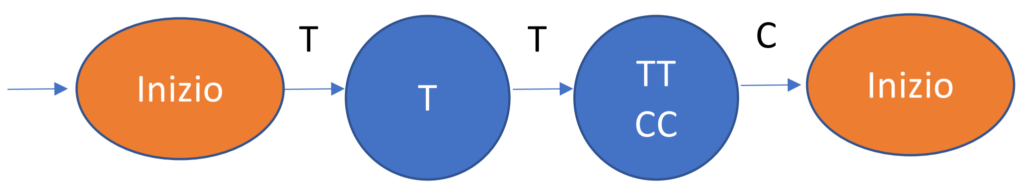
$\hat{\boldsymbol{\delta}}$**:** $Q \times \Sigma^* \rightarrow Q$

Base: $\hat{\delta}(\mathbf{q}, \varepsilon) = \mathbf{q}$

Induzione: $\hat{\delta}(\mathbf{q}, xa) = \delta(\hat{\delta}(\mathbf{q}, x), a)$

x in Σ^*, a in Σ

- La funzione di transizione estesa $\hat{\delta}$ opera su stati e stringhe (invece che su stati e simboli)
- $\hat{\delta}(q0,w) = p$ denota che p è lo stato raggiunto a partire da q0 quando si riceve in input uno a uno i simboli di w



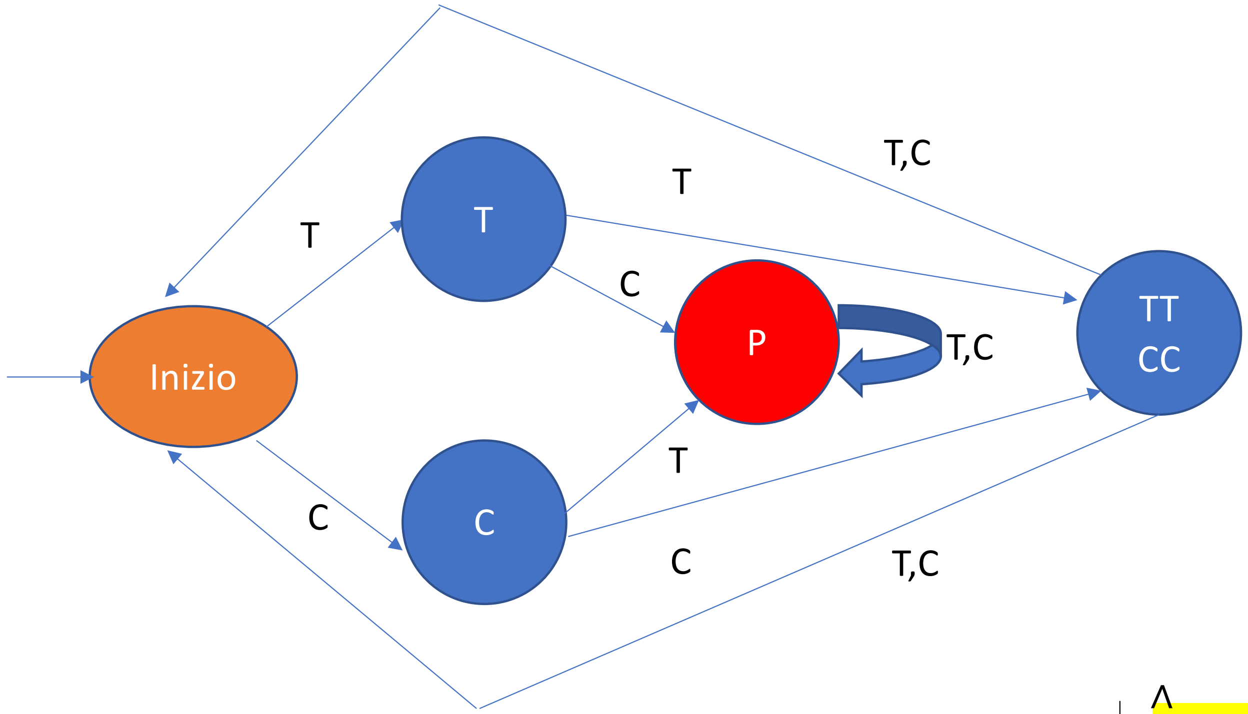
$\hat{\delta}$(Inizio, TTC) = $\delta(\hat{\delta}$(Inizio, TT),C)

$\hat{\delta}$(Inizio, TT) = $\delta(\hat{\delta}$(Inizio, T),T)

$\hat{\delta}$(Inizio, T) = $\delta(\hat{\delta}$(Inizio, ε),T)

$\hat{\delta}$(Inizio, ε) =

FUNZIONE DI TRANSIZIONE ESTESA: definizione induttiva



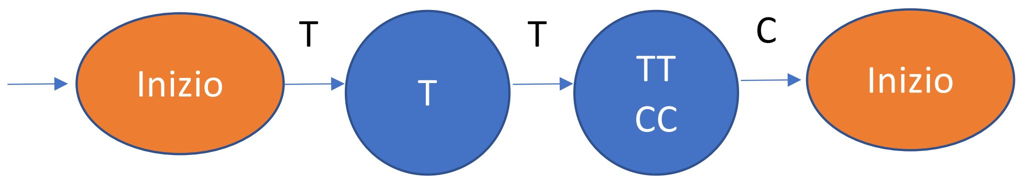
$\hat{\delta}$**:** Q x Σ* → Q

Base: $\hat{\delta}(\mathbf{q}, \varepsilon) = \mathbf{q}$

Induzione: $\hat{\delta}(\mathbf{q}, xa) = \delta(\hat{\delta}(\mathbf{q}, x), a)$

x in Σ*, a in Σ

- La funzione di transizione estesa $\hat{\delta}$ opera su stati e stringhe (invece che su stati e simboli)
- $\hat{\delta}(q0,w) = p$ denota che p è lo stato raggiunto a partire da q0 quando si riceve in input uno a uno i simboli di w



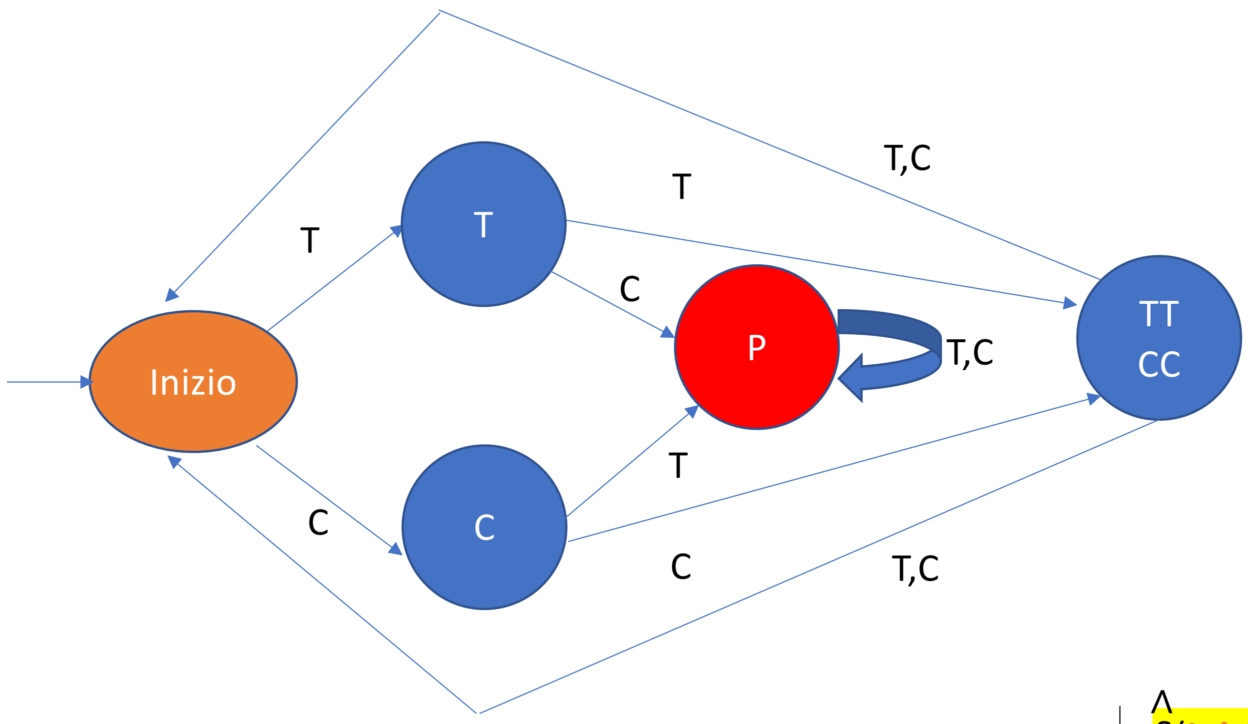
$\hat{\delta}$(Inizio, TTC) = δ($\hat{\delta}$(Inizio, TT),C)

$\hat{\delta}$(Inizio, TT) = δ($\hat{\delta}$(Inizio, T),T)

$\hat{\delta}$(Inizio, T) = δ($\hat{\delta}$(Inizio, ε),T)

$\hat{\delta}$(Inizio, ε) = Inizio

FUNZIONE DI TRANSIZIONE ESTESA: definizione induttiva

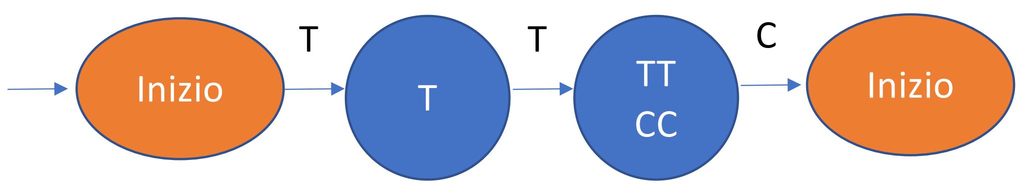


$\hat{\delta}$: $Q \times \Sigma^* \rightarrow Q$

Base: $\hat{\delta}(\mathbf{q}, \varepsilon) = \mathbf{q}$

Induzione: $\hat{\delta}(\mathbf{q}, xa) = \delta(\hat{\delta}(\mathbf{q}, x), a)$
x in Σ^*, a in Σ

- La funzione di transizione estesa $\hat{\delta}$ opera su stati e stringhe (invece che su stati e simboli)
- $\hat{\delta}(q0,w) = p$ denota che p è lo stato raggiunto a partire da q0 quando si riceve in input uno a uno i simboli di w



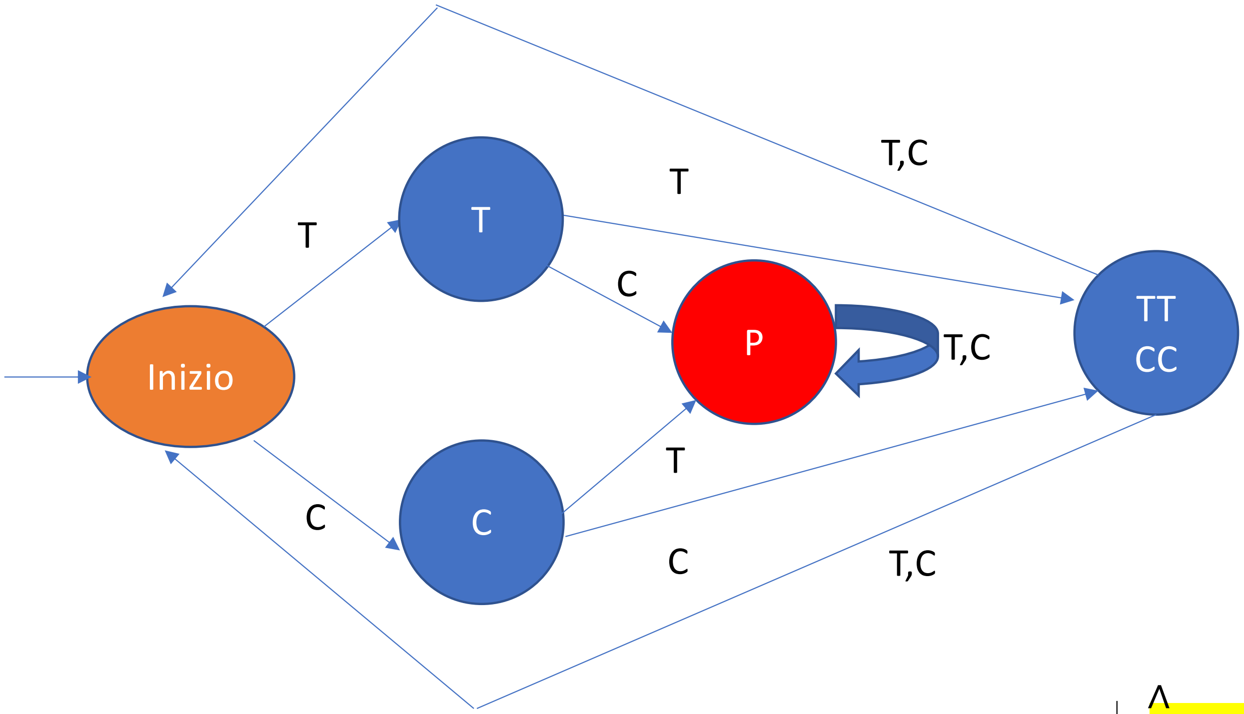
$\hat{\delta}(\text{Inizio}, TTC) = \delta(\hat{\delta}(\text{Inizio}, TT), C)$

$\hat{\delta}(\text{Inizio}, TT) = \delta(\hat{\delta}(\text{Inizio}, T), T)$

$\hat{\delta}(\text{Inizio}, T) = \delta(\hat{\delta}(\text{Inizio}, \varepsilon), T) = \delta(\text{Inizio}, T) = T$

$\hat{\delta}(\text{Inizio}, \varepsilon) = \text{Inizio}$

FUNZIONE DI TRANSIZIONE ESTESA: definizione induttiva

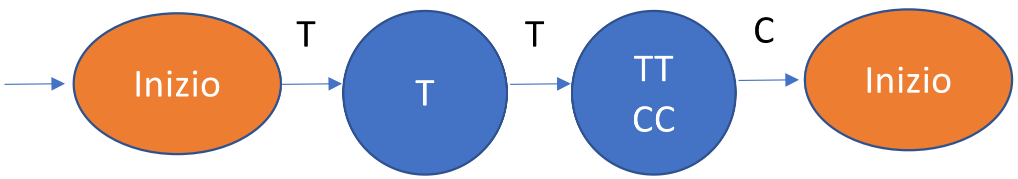


$\hat{\boldsymbol{\delta}}$**:** Q x Σ* → Q

Base: $\hat{\delta}(\mathbf{q}, \varepsilon) = \mathbf{q}$

Induzione: $\hat{\delta}(\mathbf{q}, xa) = \delta(\hat{\delta}(\mathbf{q}, x), a)$
x in Σ*, a in Σ

- La funzione di transizione estesa $\hat{\delta}$ opera su stati e stringhe (invece che su stati e simboli)
- $\hat{\delta}(q0,w) = p$ denota che p è lo stato raggiunto a partire da q0 quando si riceve in input uno a uno i simboli di w



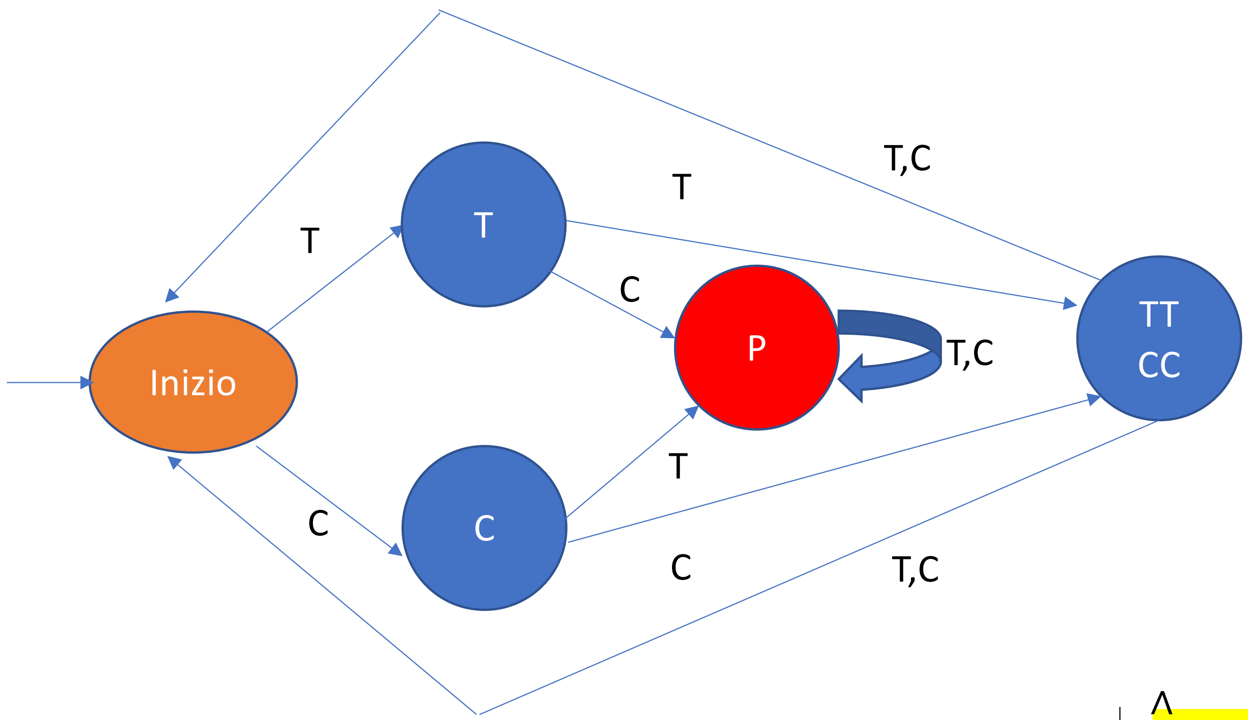
$\hat{\delta}(\text{Inizio}, TTC) = \delta(\hat{\delta}(\text{Inizio}, TT), C)$

$\hat{\delta}(\text{Inizio}, TT) = \delta(\hat{\delta}(\text{Inizio}, T), T) = \delta(T, T) = TT/CC$

$\hat{\delta}(\text{Inizio}, T) = \delta(\hat{\delta}(\text{Inizio}, \varepsilon), T) = \delta(\text{Inizio}, T) = T$

$\hat{\delta}(\text{Inizio}, \varepsilon) = \text{Inizio}$

FUNZIONE DI TRANSIZIONE ESTESA: definizione induttiva



$\hat{\delta}$: $Q \times \Sigma^* \rightarrow Q$

Base: $\hat{\delta}(\mathbf{q}, \varepsilon) = \mathbf{q}$

Induzione: $\hat{\delta}(\mathbf{q}, xa) = \delta(\hat{\delta}(\mathbf{q}, x), a)$
x in Σ^*, a in Σ

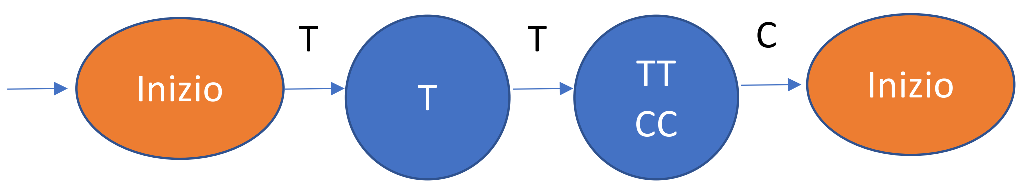
- La funzione di transizione estesa $\hat{\delta}$ opera su stati e stringhe (invece che su stati e simboli)
- $\hat{\delta}(q0,w)$ = p denota che p è lo stato raggiunto a partire da q0 quando si riceve in input uno a uno i simboli di w

$\hat{\delta}$(Inizio, TTC) = $\delta(\hat{\delta}$(Inizio, TT),C) = δ(TT/CC, T) = Inizio

$\hat{\delta}$(Inizio, TT) = $\delta(\hat{\delta}$(Inizio, T),T) = δ(T, T) = TT/CC

$\hat{\delta}$(Inizio, T) = $\delta(\hat{\delta}$(Inizio, ε),T) = δ(Inizio, T) = T

$\hat{\delta}$(Inizio, ε) = Inizio



Definizioni induttive

$\hat{\delta}$

Base: $\hat{\delta}(\mathbf{q}, \varepsilon) = \mathbf{q}$

Induzione: $\hat{\delta}(\mathbf{q}, xa) = \delta(\hat{\delta}(\mathbf{q}, x), a)$

x in Σ^*, a in Σ

$\hat{\delta}$(Inizio, TTC) = δ($\hat{\delta}$(Inizio, TT),C) = δ(TT/CC, T) = Inizio

$\hat{\delta}$(Inizio, TT) = δ($\hat{\delta}$(Inizio, T),T) = δ(T, T) = TT/CC

$\hat{\delta}$(Inizio, T) = δ($\hat{\delta}$(Inizio, ε),T) = δ(Inizio, T) = T

$\hat{\delta}$(Inizio, ε) = Inizio

Fattoriale

Base: 0! = 1

Induzione: n! = n x (n-1)!

3! = 3 x 2! = 3 x 2 = 6

2! = 2 x 1! = 2 x 1 = 2

1! = 1 x 0! = 1 x 1 = 1

0! = 1