How to achieve Reachability in Broadcast Networks?

An ongoing work

Journée Pavedys

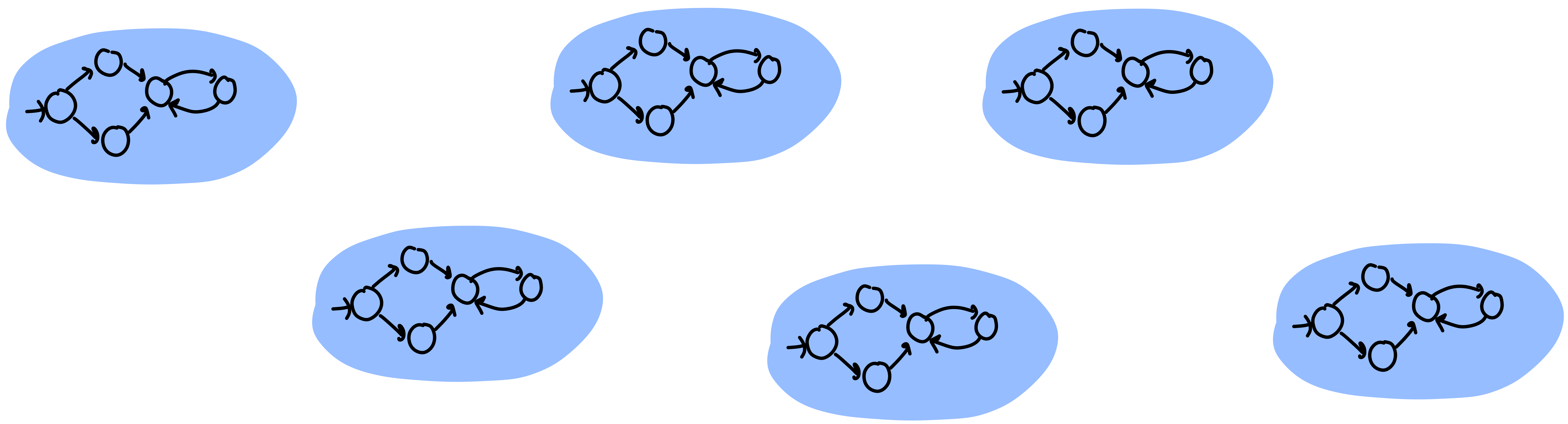
14 octobre 2024, Grenoble

Lucie Guillou
IRIF
Paris, France

Arnaud Sangnier
DIBRIS
Genova, Italy

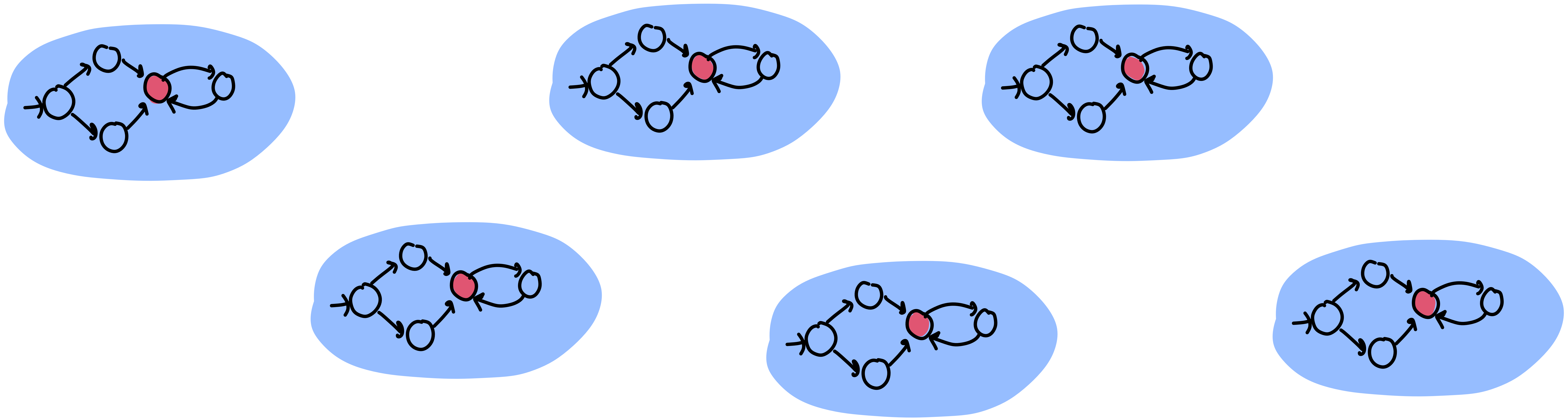
Tali Sznajder
LIP6
Paris, France

Parameterized Broadcast Networks



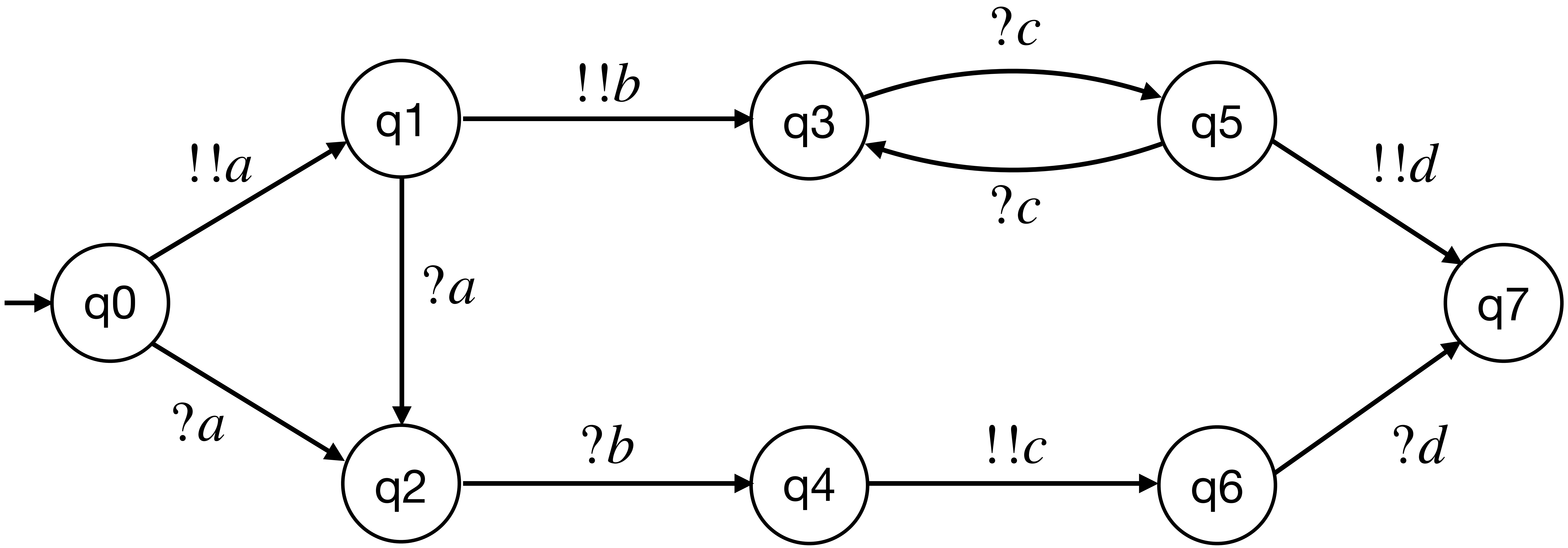
- Unknown number of agents
- Each agent follows a protocol given as a finite-state machine
- Synchronous Communication (Broadcast)
- Interleaving Semantics

The Reachability Question

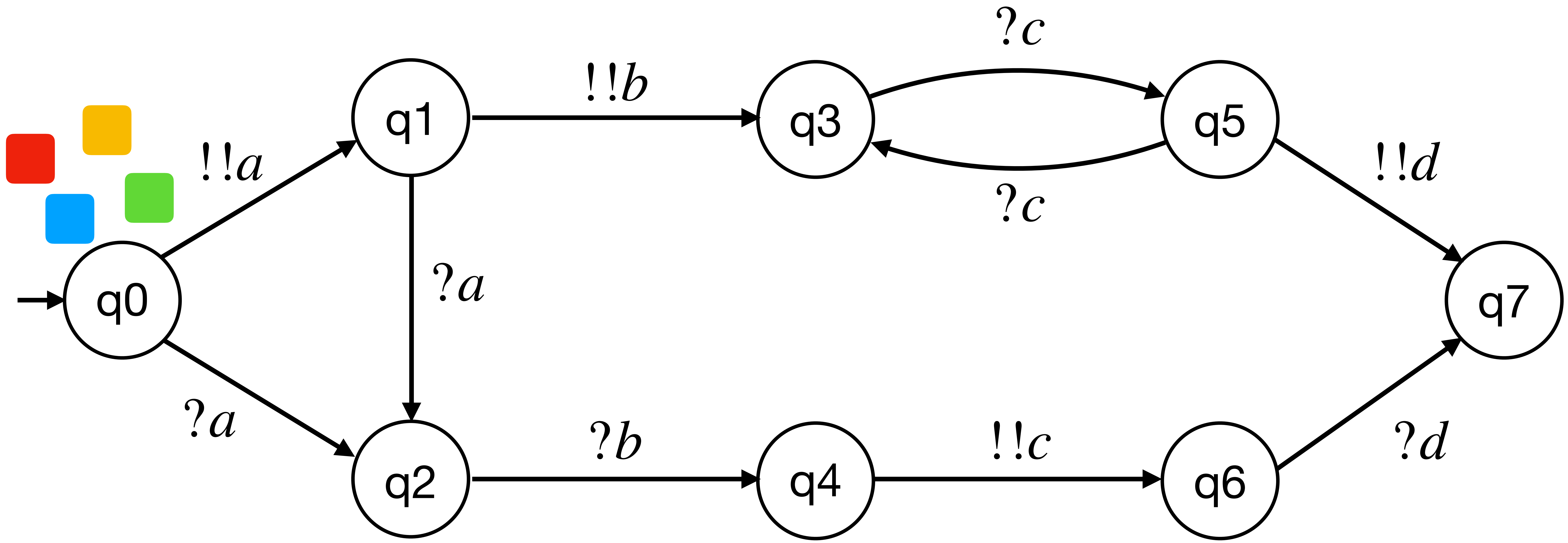


- Is there a number of agents such that there exists a run leading to a bad configuration?

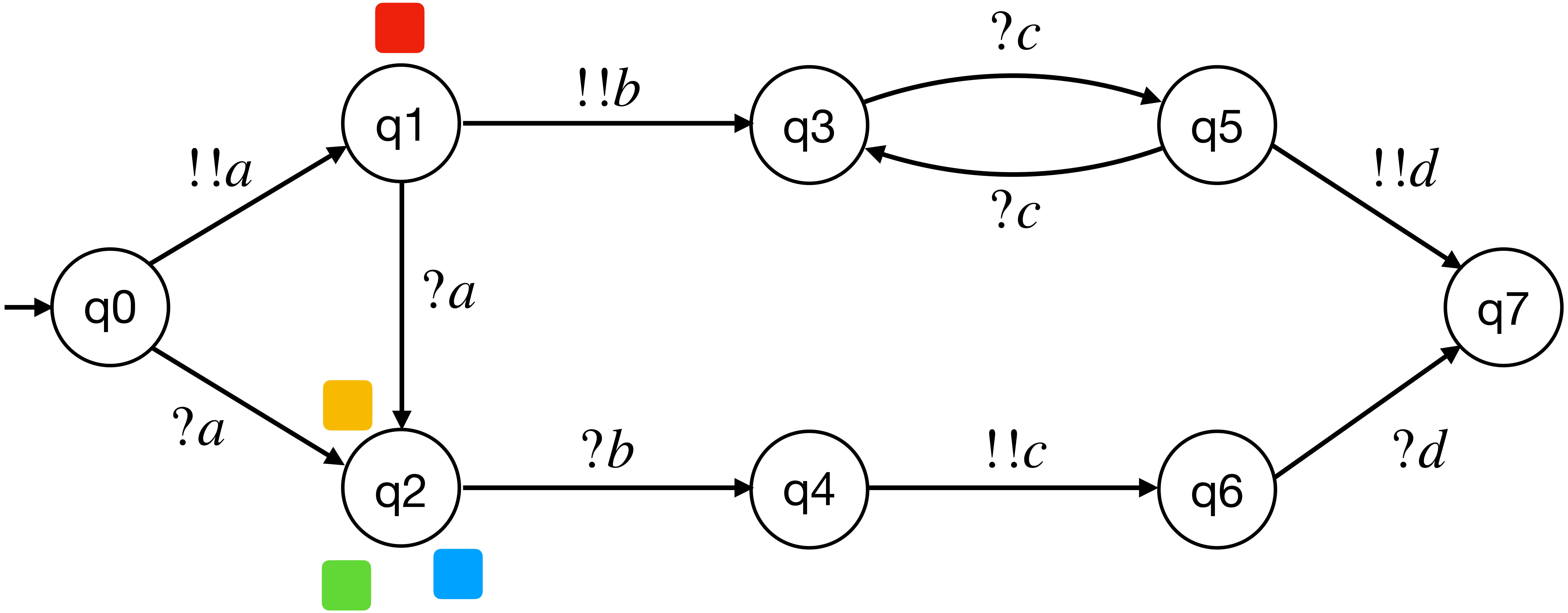
Broadcast Protocols



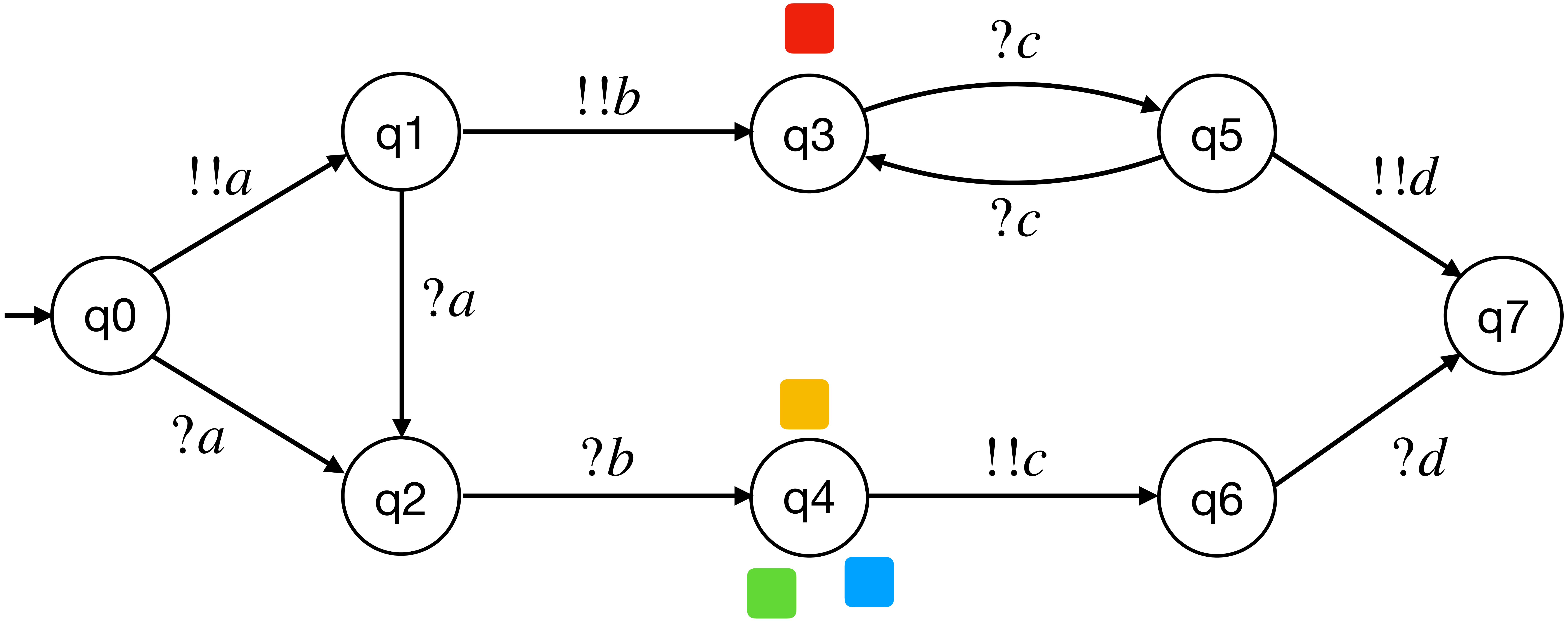
Example of an execution



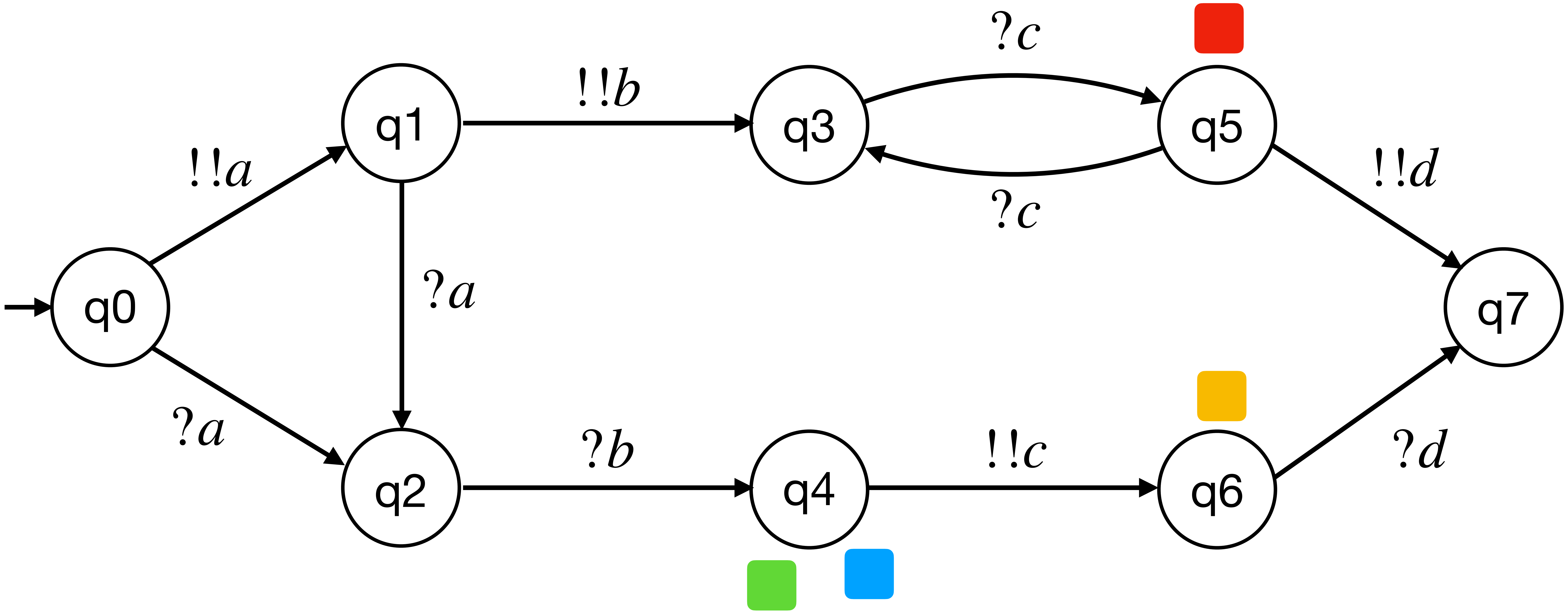
Example of an execution



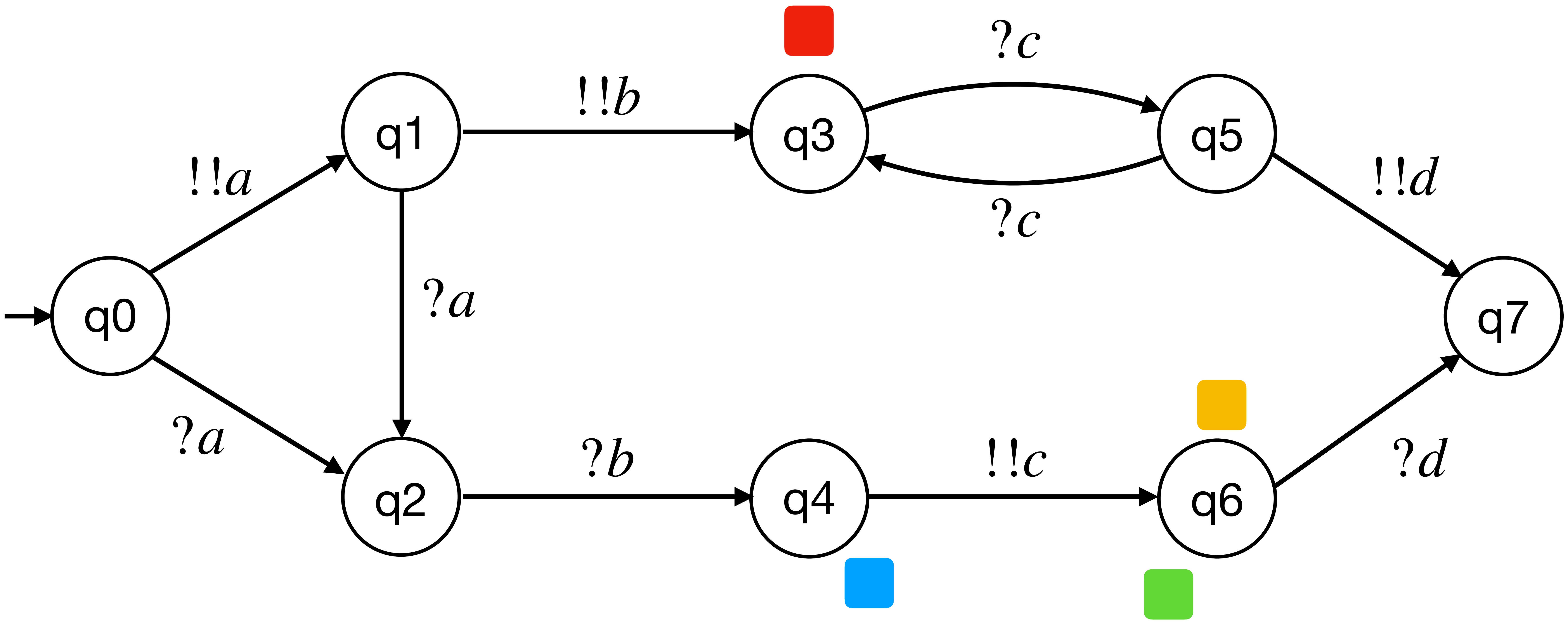
Example of an execution



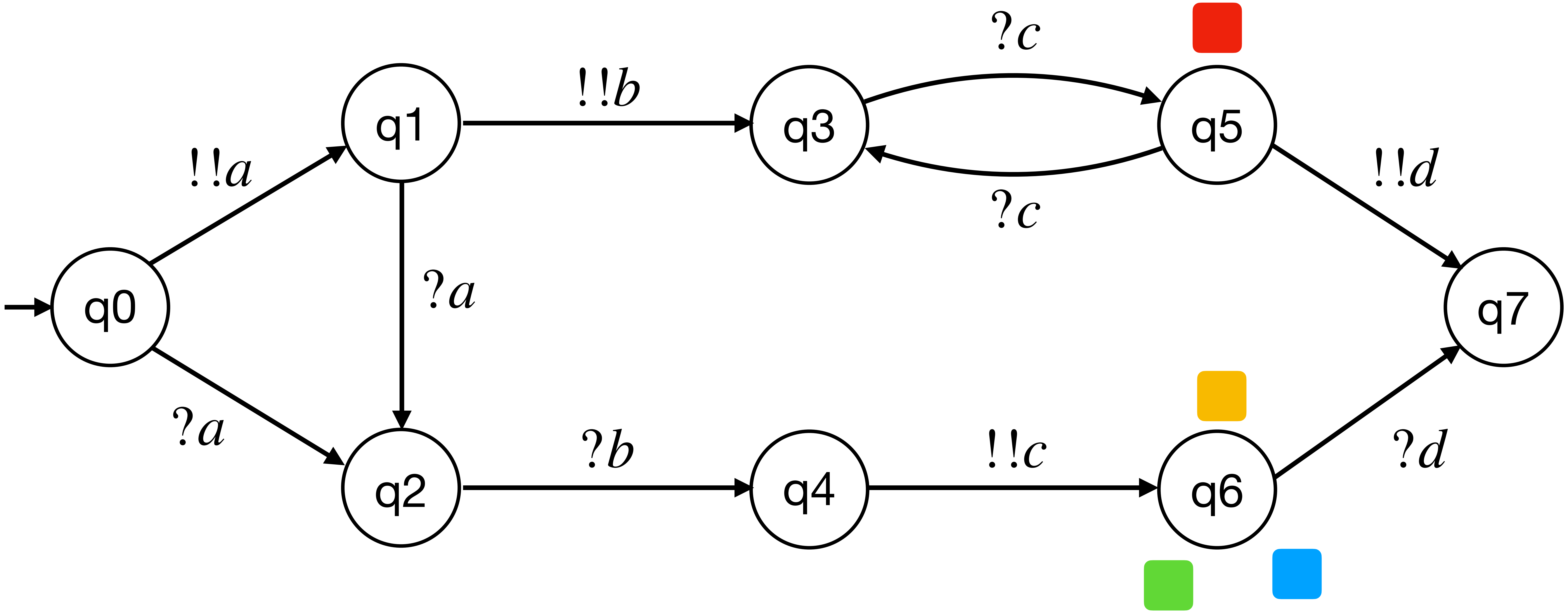
Example of an execution



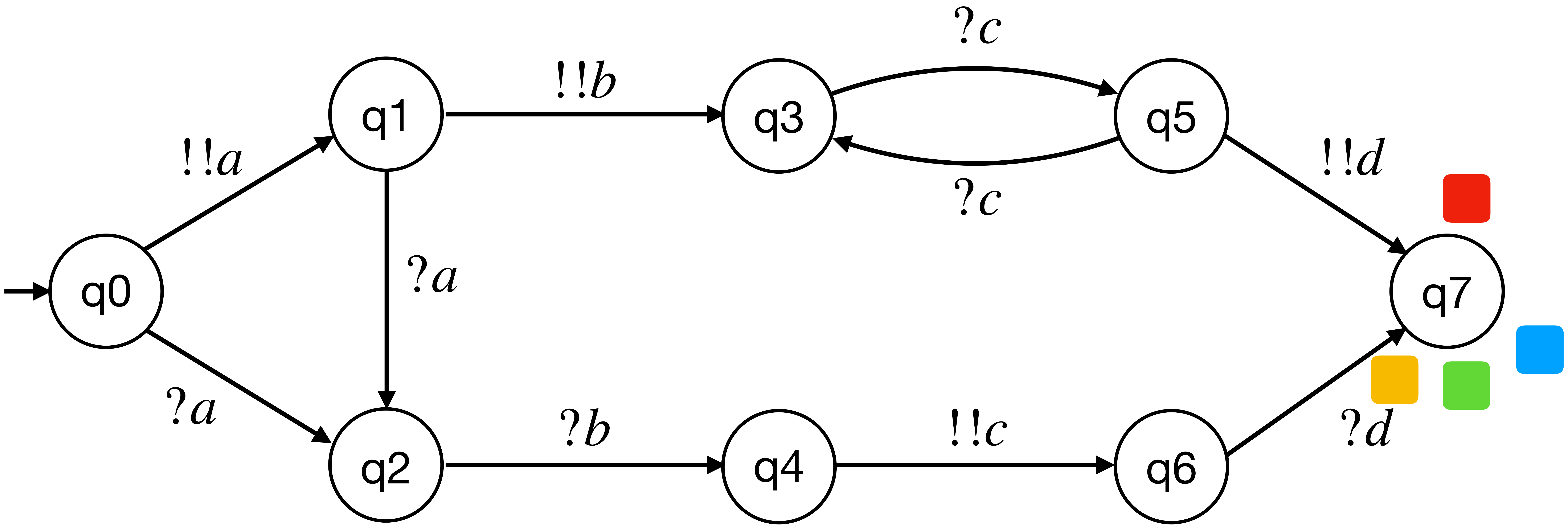
Example of an execution



Example of an execution



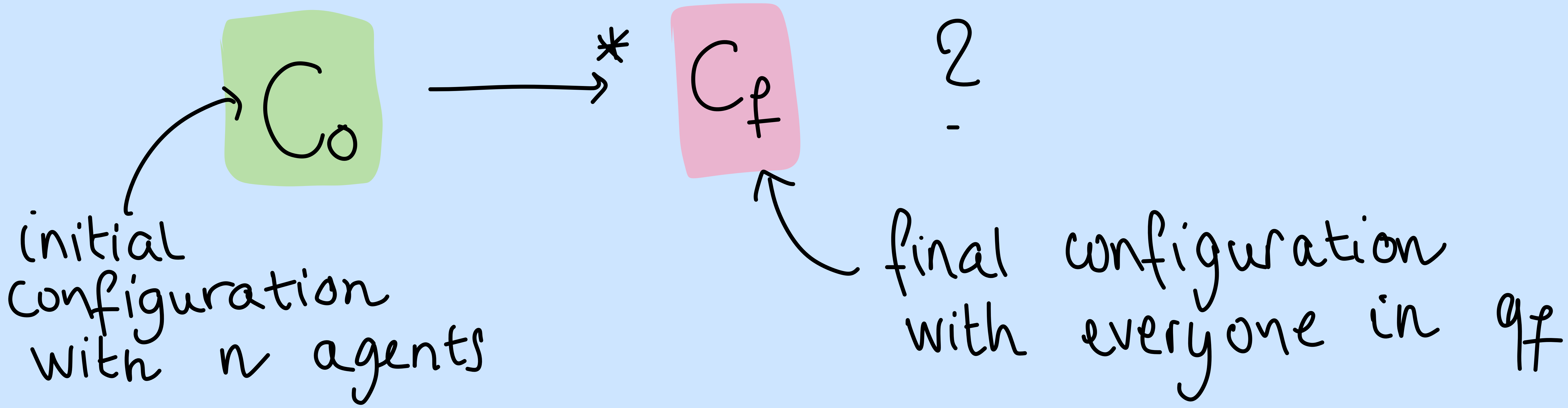
Example of an execution



The Reachability Problem Formalized

REACH($\mathscr{P}, q_f$):

Is there a number of agents $n \in \mathbb{N}$ such that:



The Reachability Problem Formalized

REACH($\mathscr{P}, q_f$ **):**

Is there a number of agents $n \in \mathbb{N}$ such that:

initial
configura
with n agents

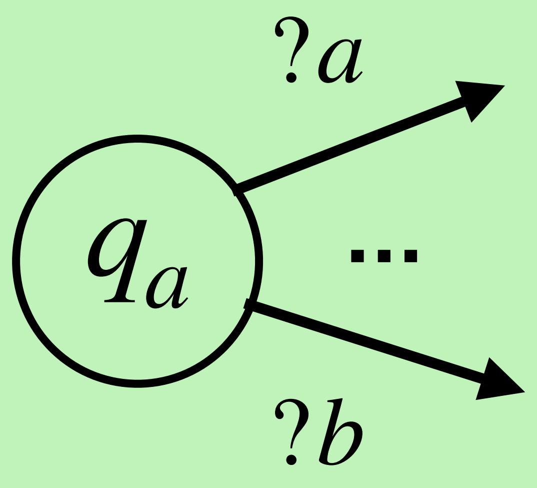
Undecidable

final configuration
with everyone in q_f

A Restriction on Protocols: Wait-Only

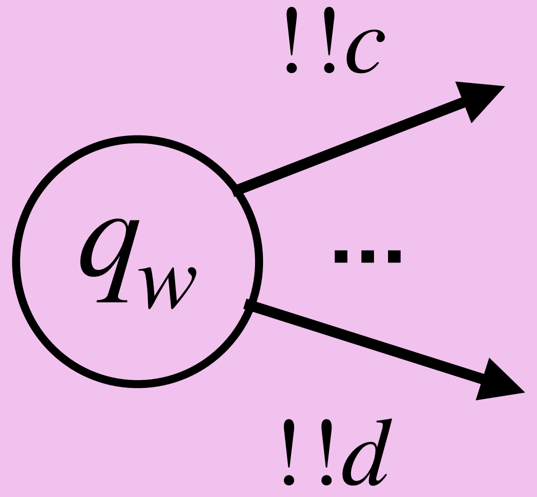
Each state is either:

a waiting state



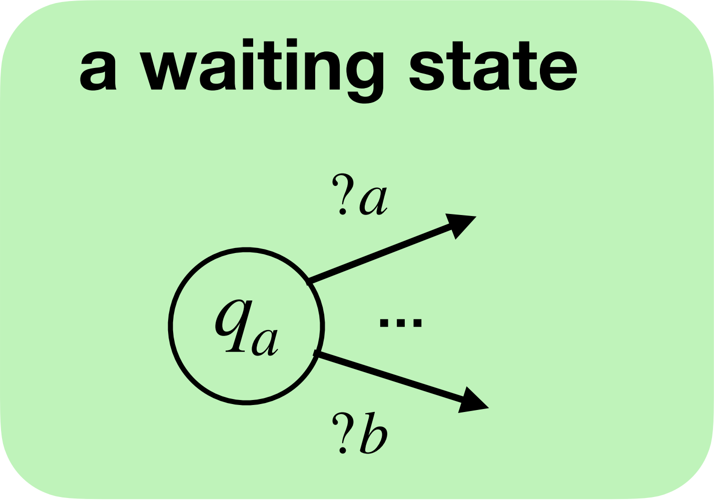
or

an action state

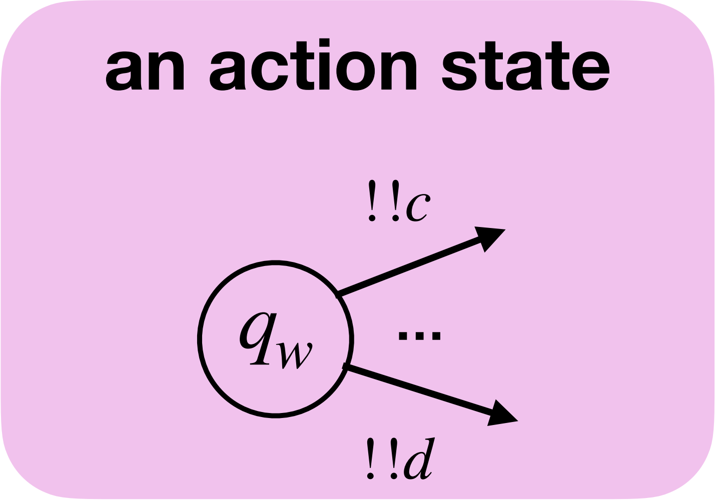


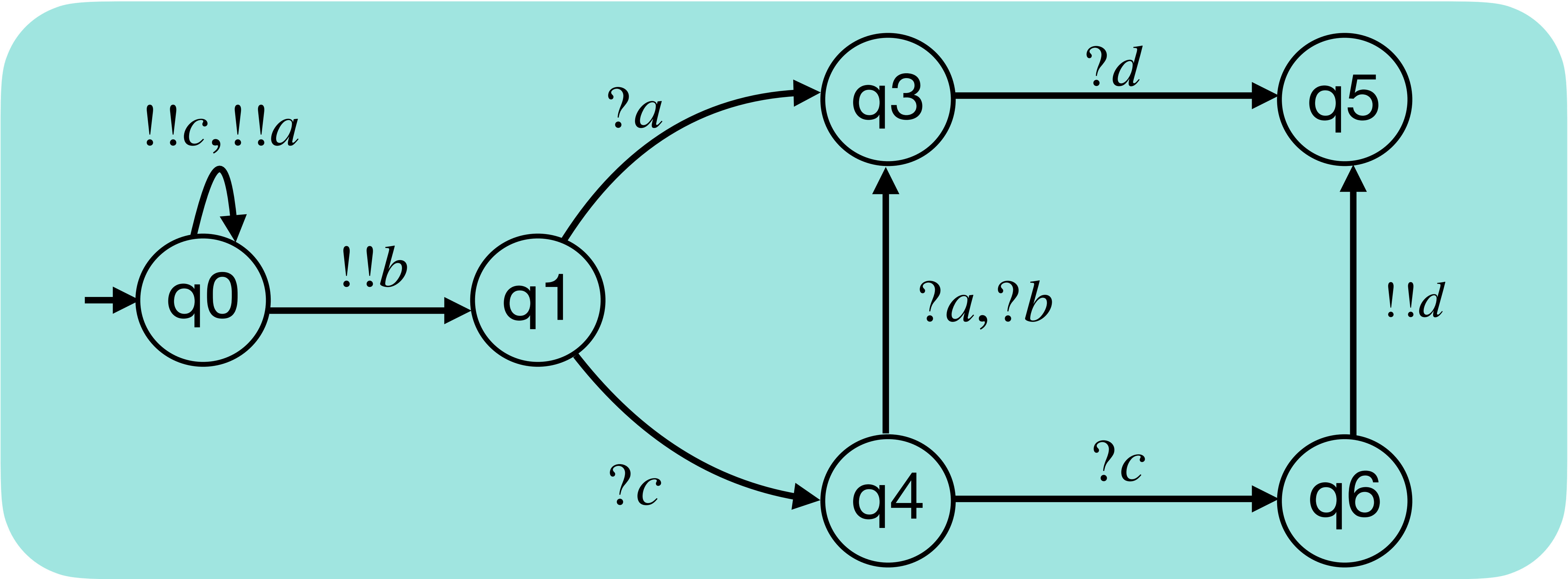
A Restriction on Protocols: Wait-Only

Each state is either:



or



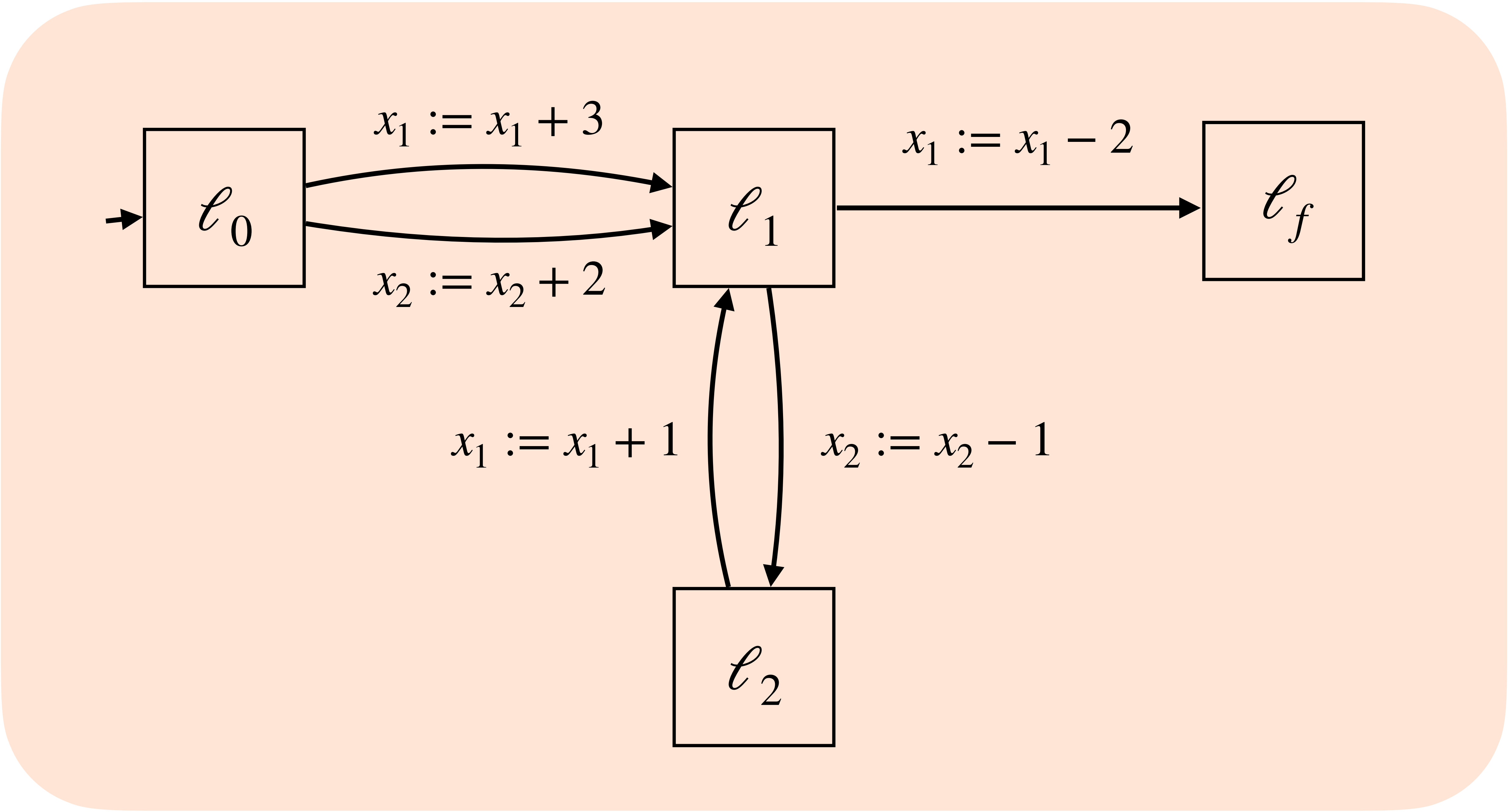


A Wait-Only Protocol

The initial state is always an action state

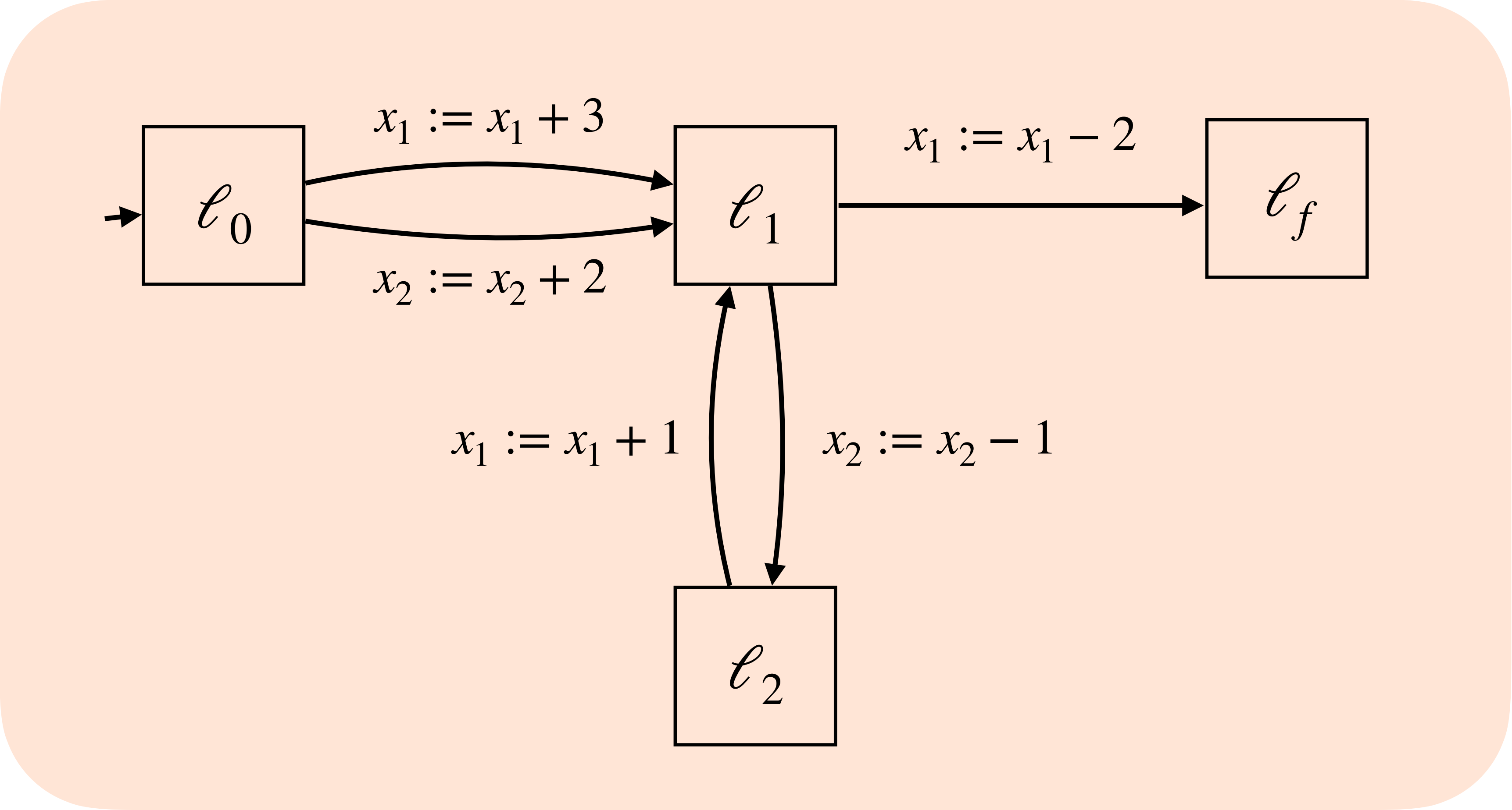
Vector Addition Systems with States

Vector Addition Systems with States



A VASS with two counters x_1, x_2

Vector Addition Systems with States



A VASS with two counters x_1, x_2

	ℓ_0	ℓ_1	ℓ_2	ℓ_1	ℓ_2	ℓ_1	ℓ_f
x_1	0	0	0	1	1	2	0
x_2	0	2	1	1	0	0	0

Reachability in VASS

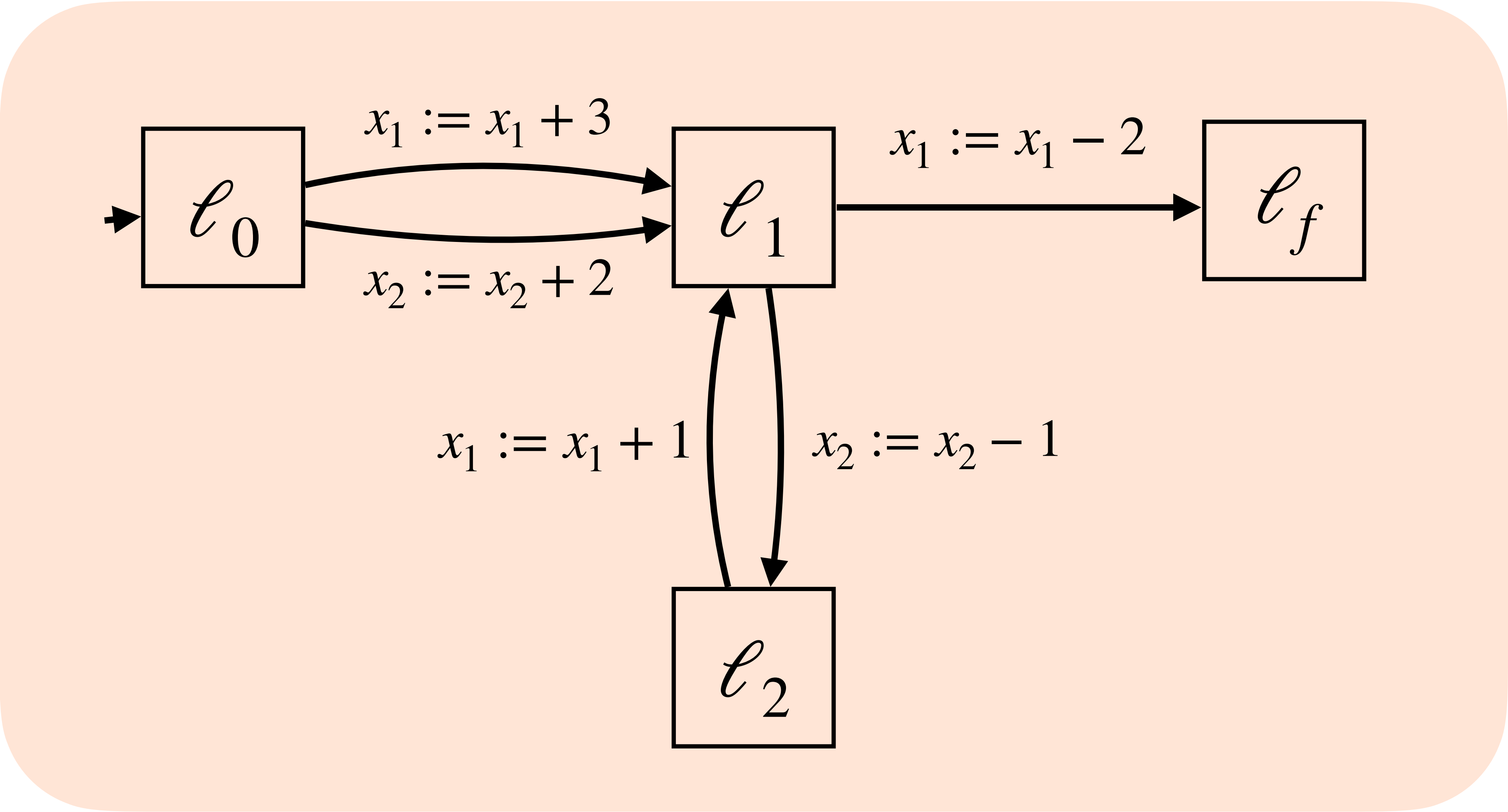
Given a VASS, can we reach $(\ell_f, 0, 0)$ from $(\ell_0, 0, 0)$?

Reachability in VASS

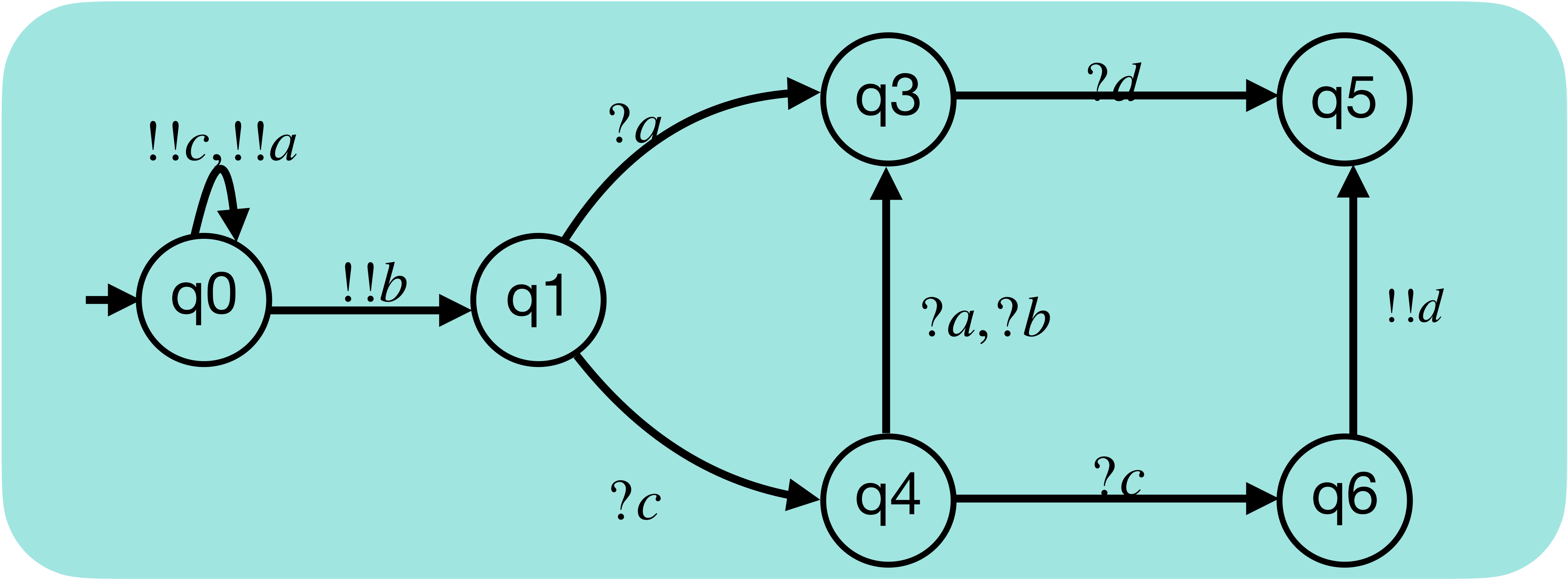
Given a VASS, can we reach $(\ell_f, 0, 0)$ from $(\ell_0, 0, 0)$?

Decidable but Ackermann-hard

[LerouxSchmitz19] [Leroux'21, CwerwinskiOrlikowski'21]

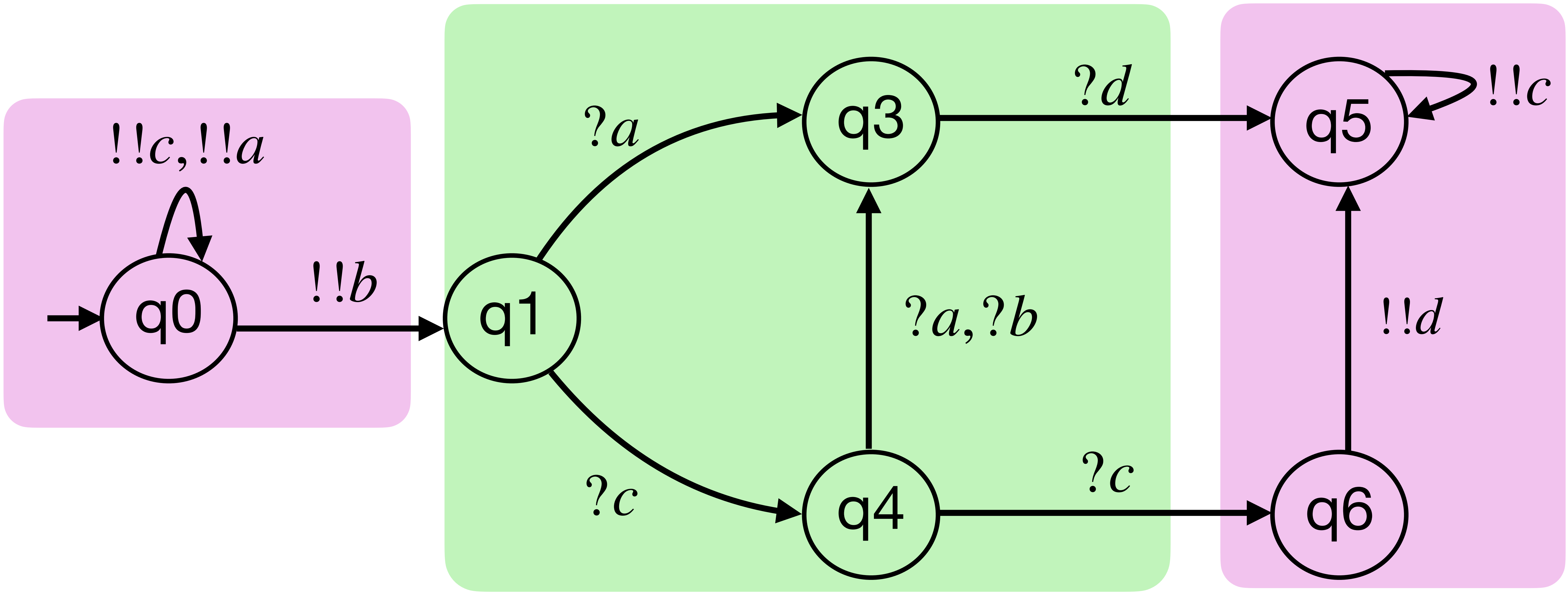


A VASS with two counters x_1, x_2

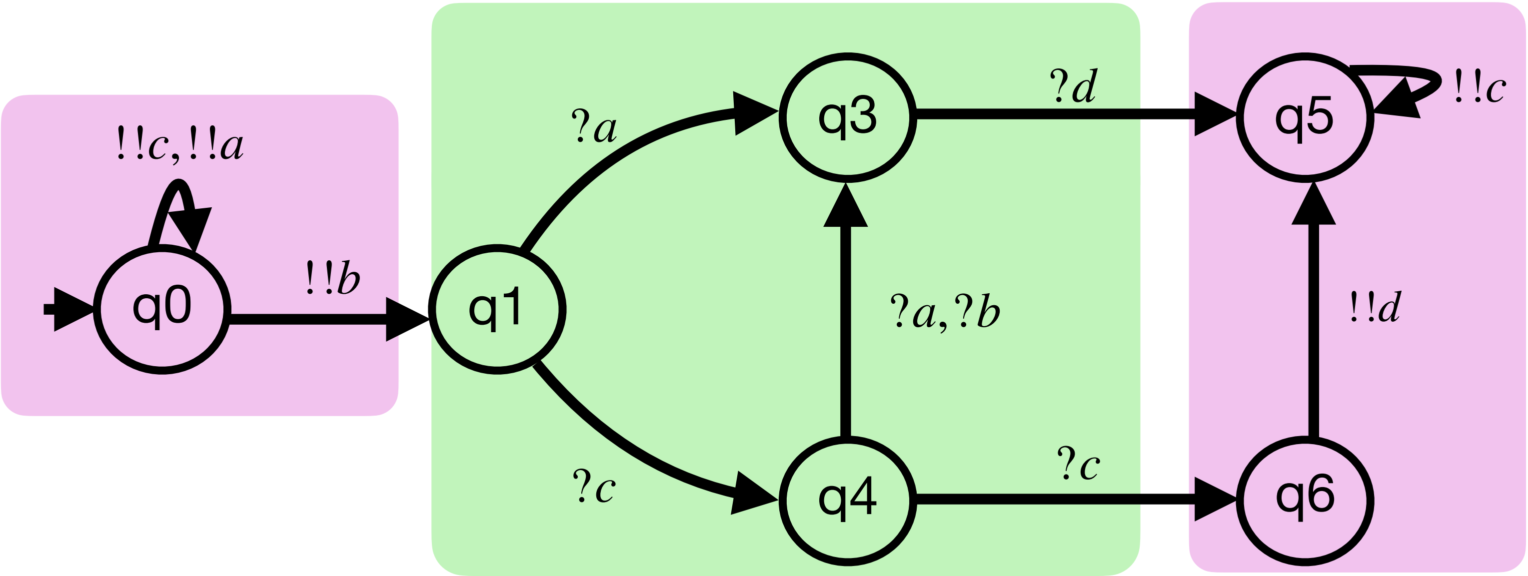


A Wait-Only Protocol

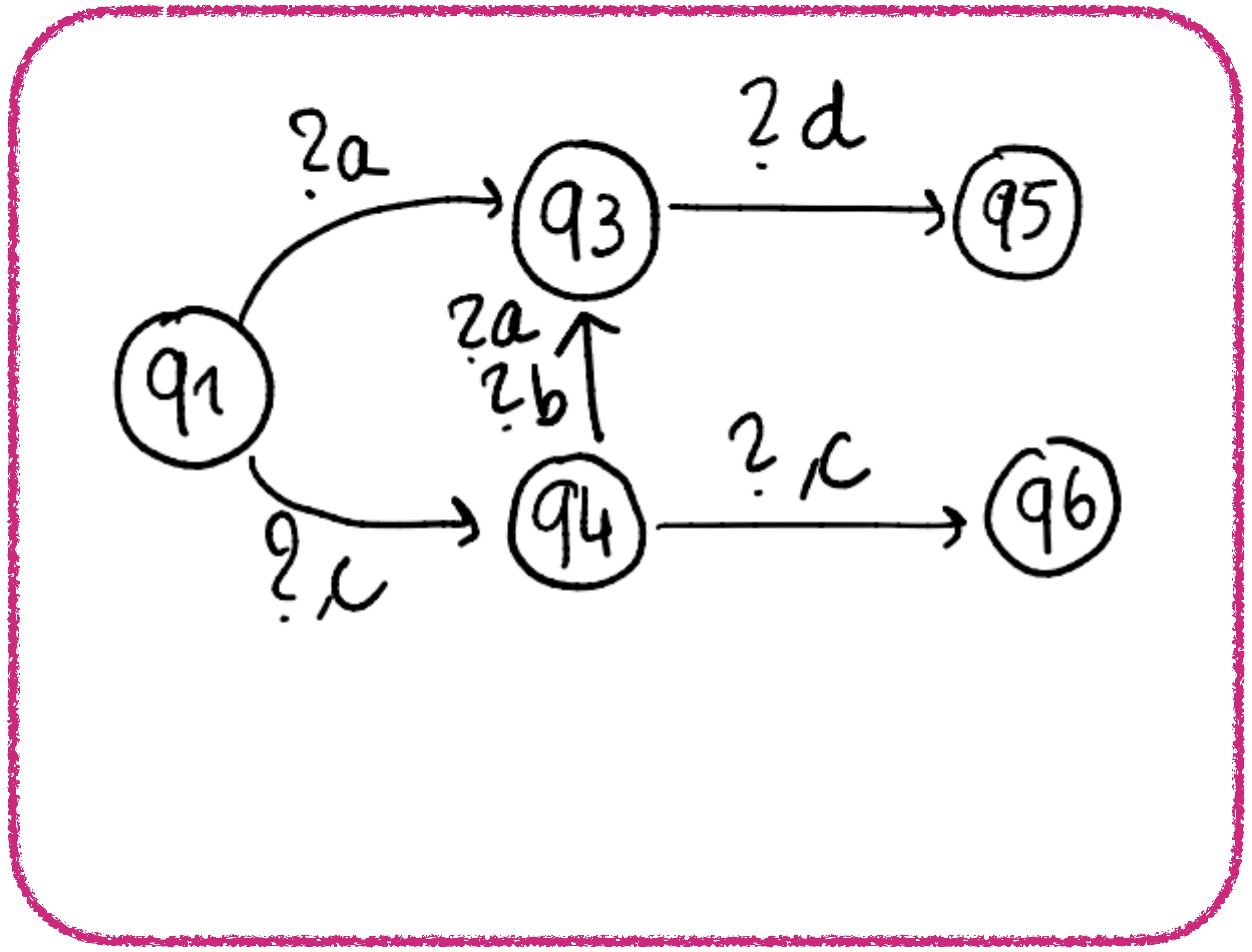
Reductions everywhere!



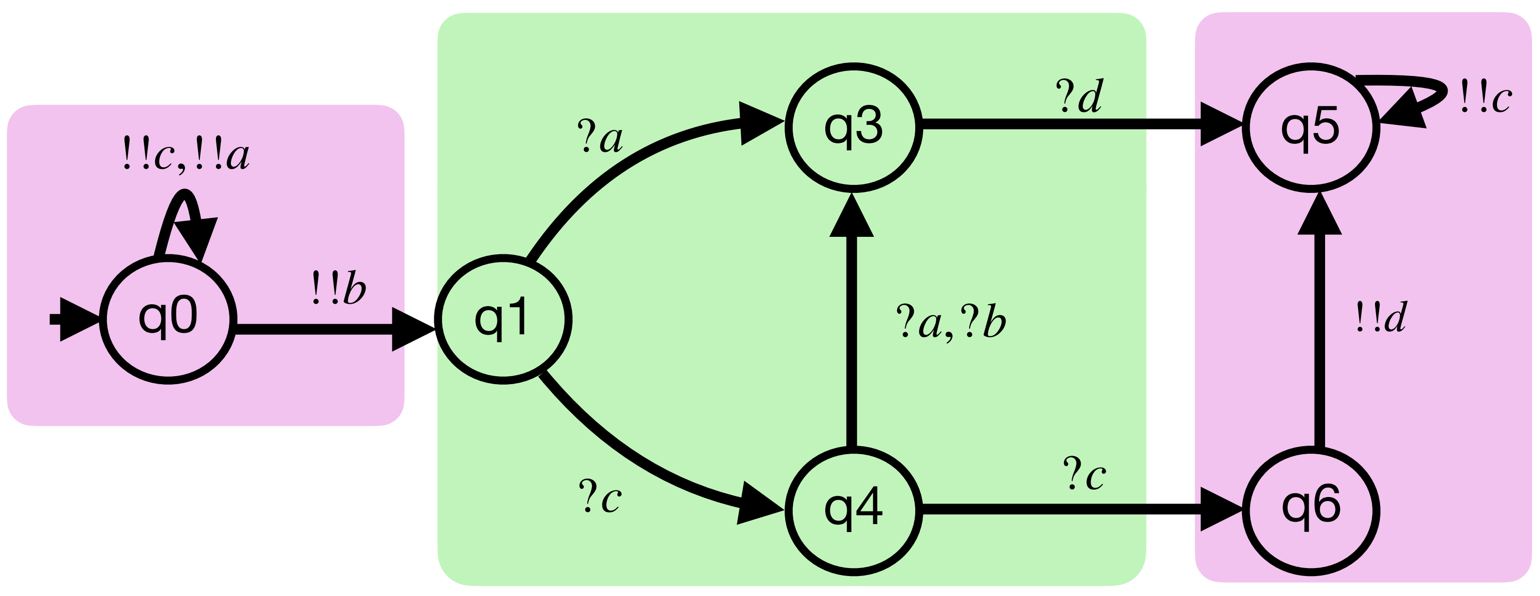
Goal: everyone on q5



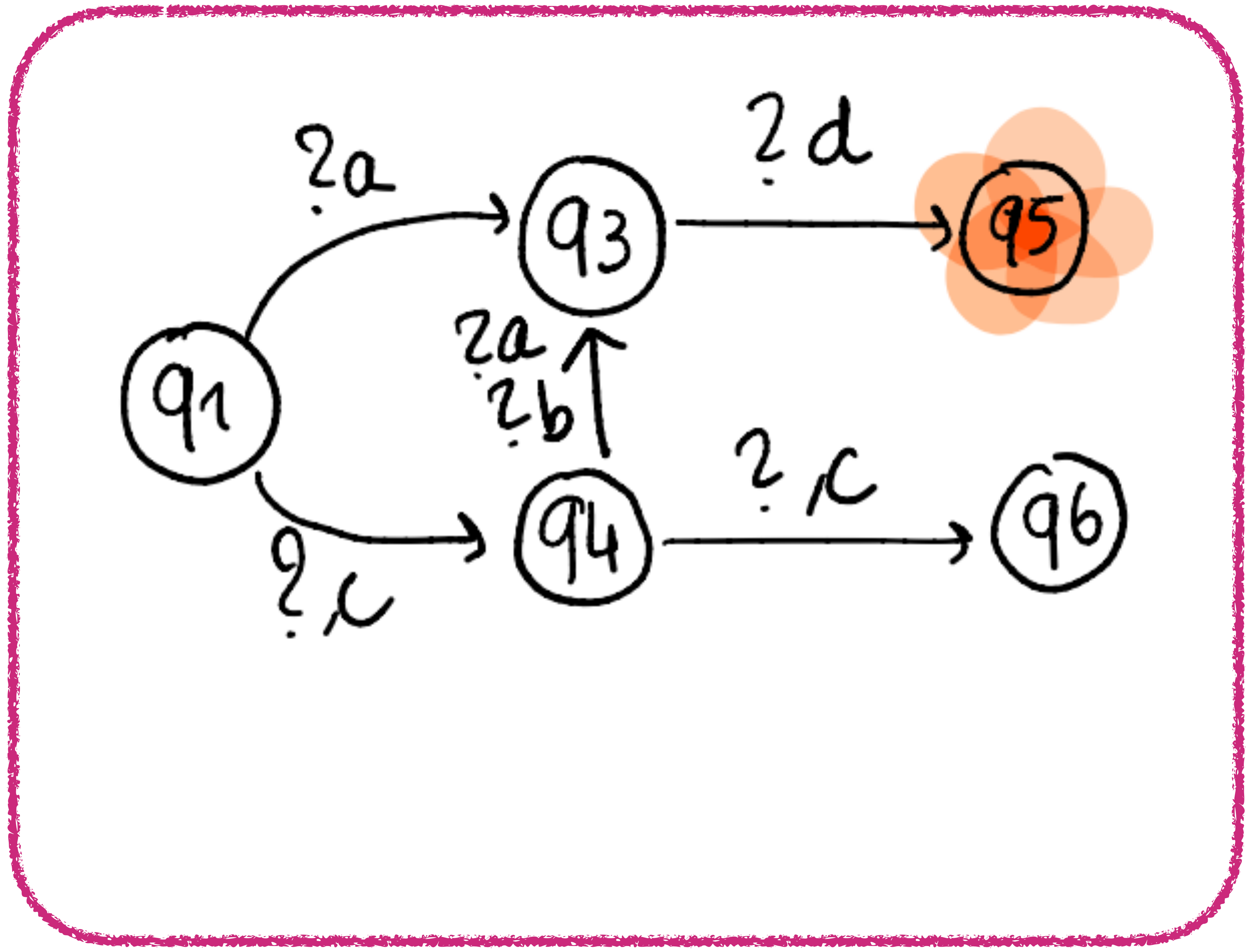
A Summary



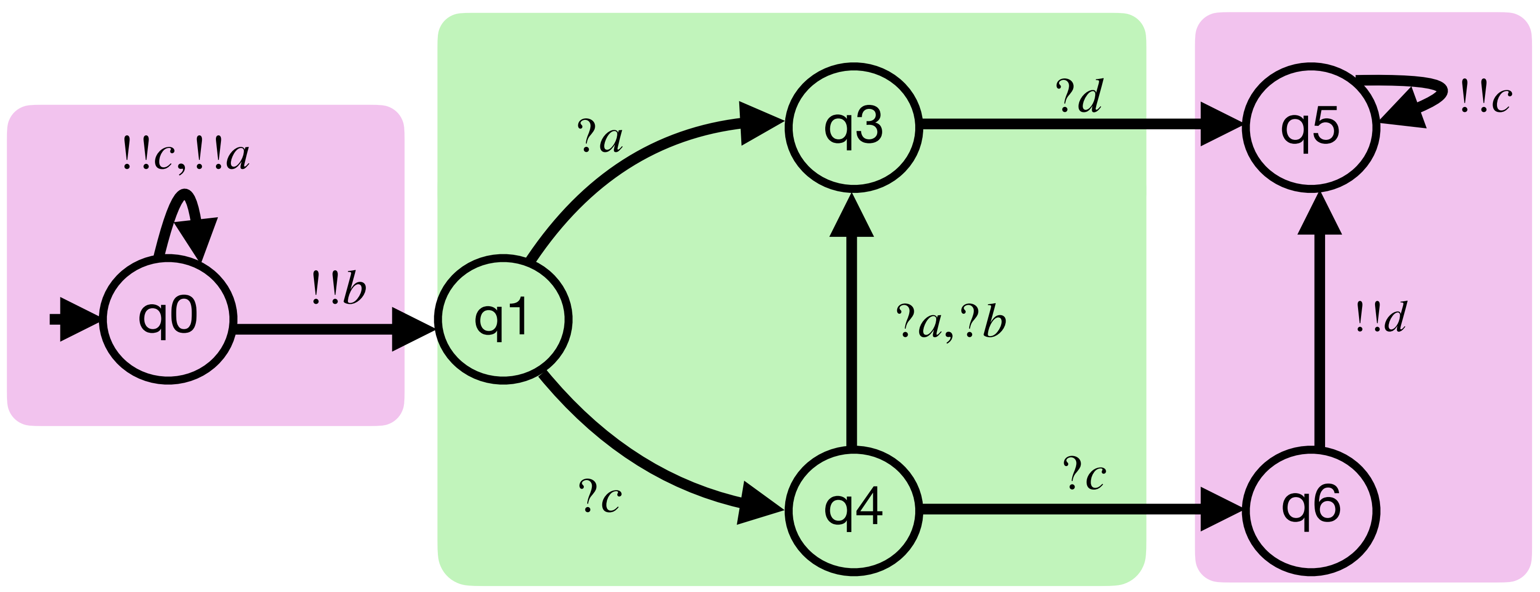
+ one counter



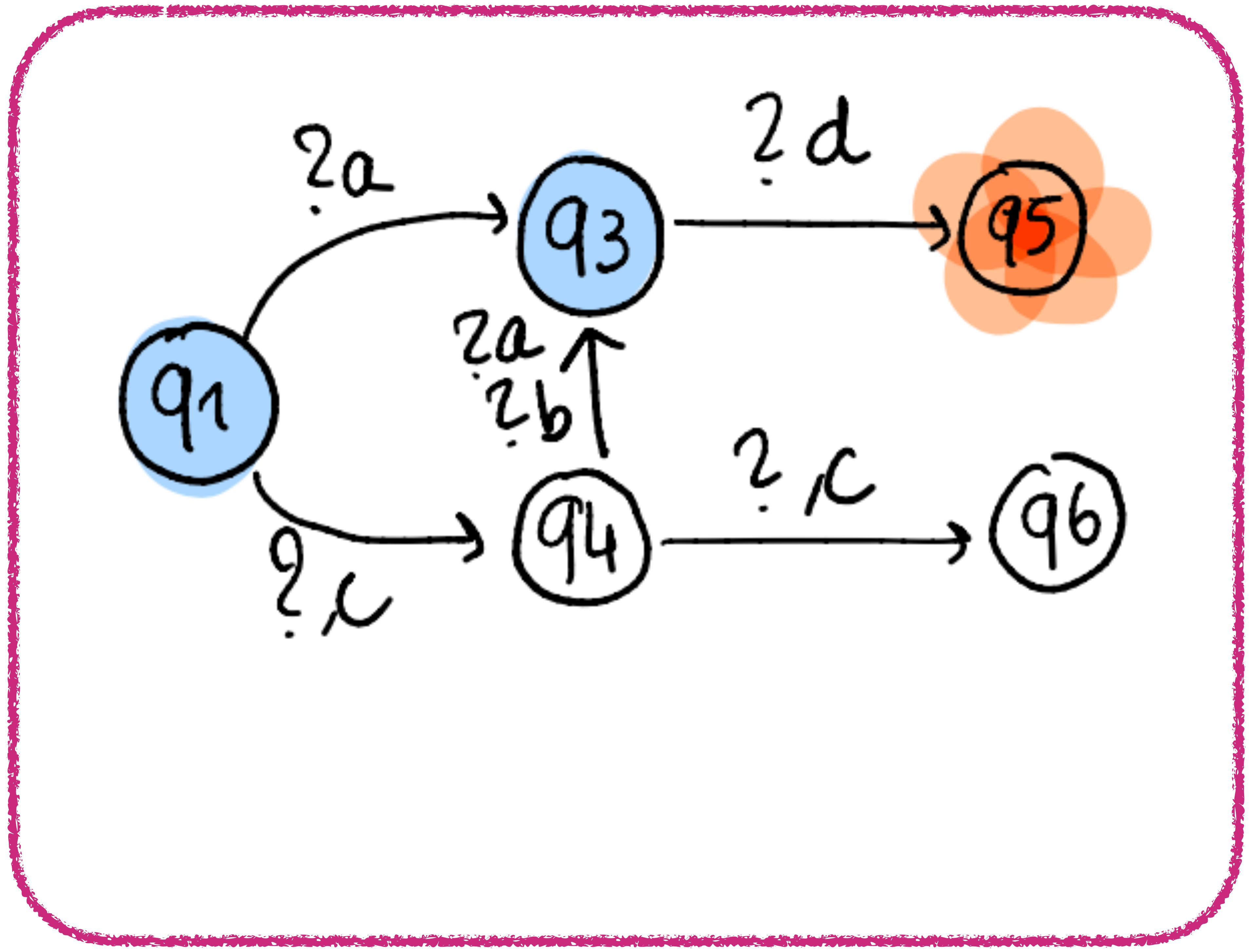
A Summary



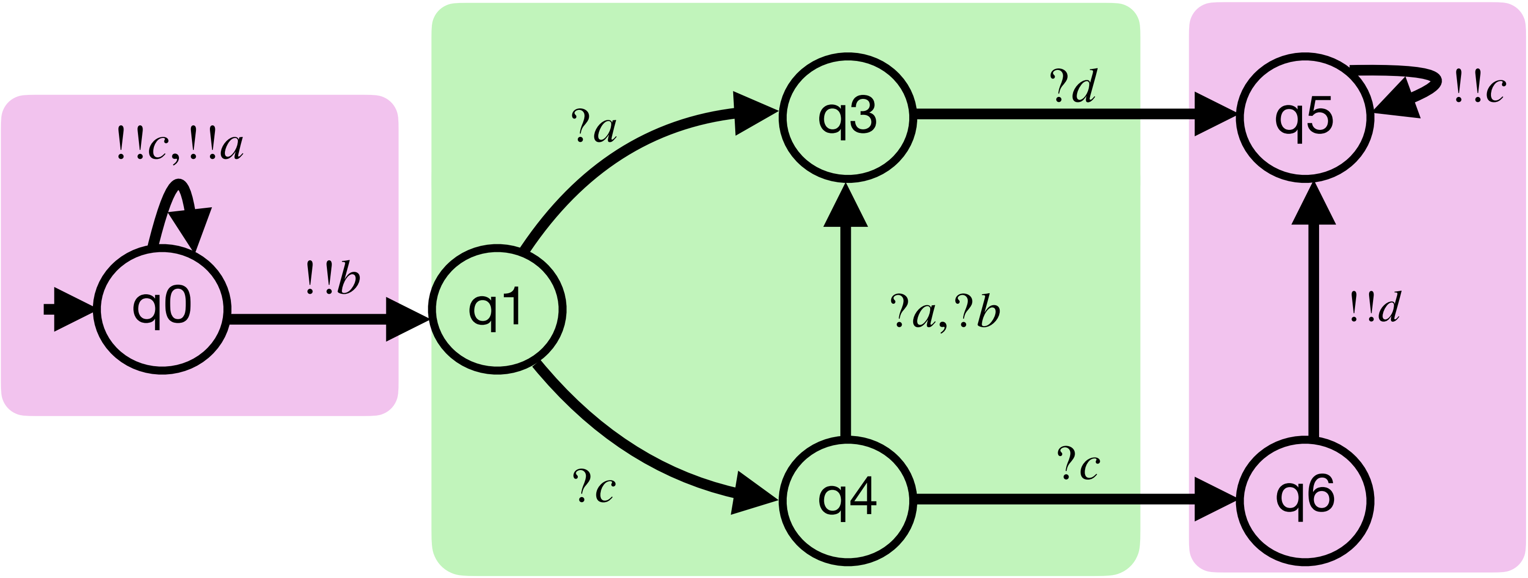
+ one counter



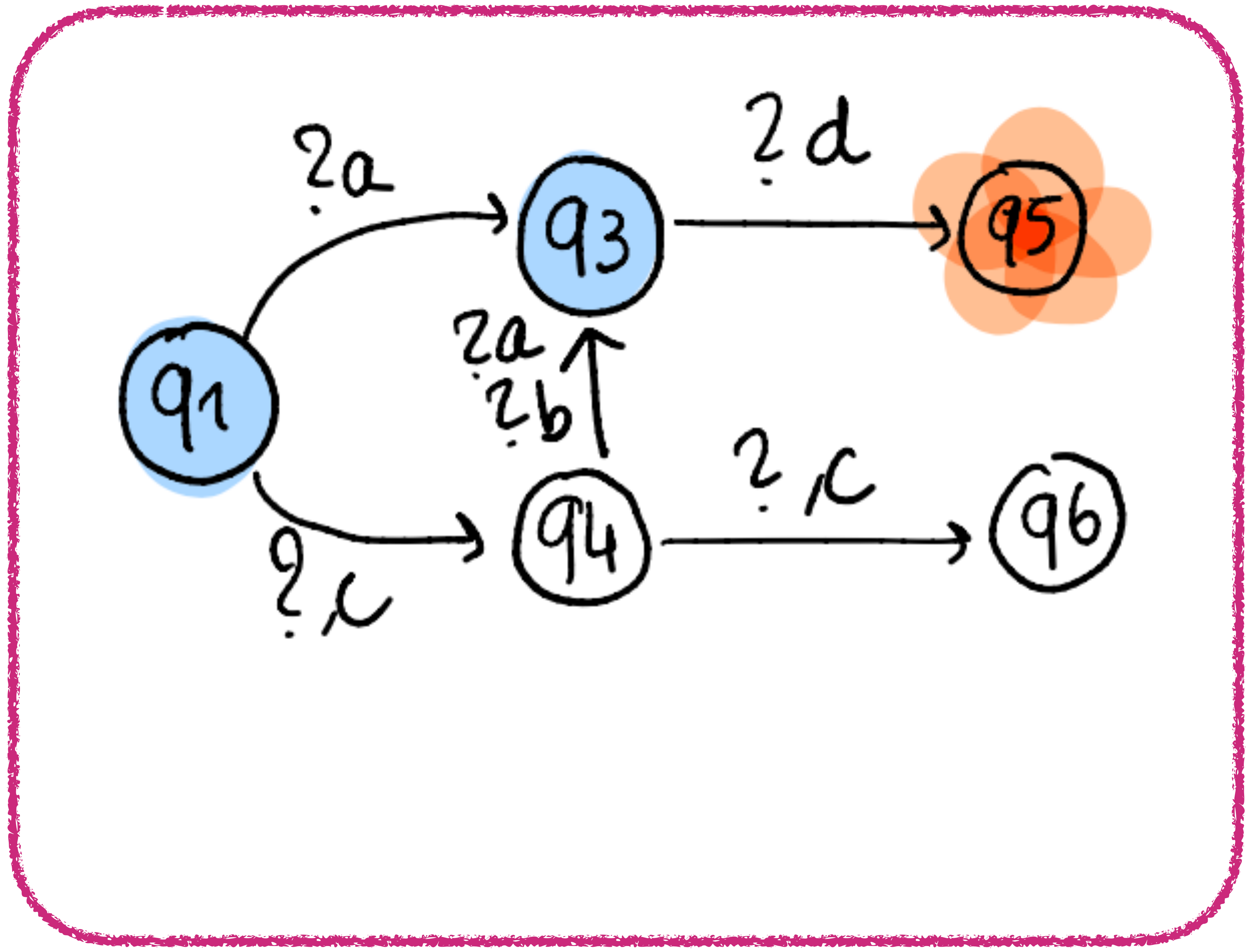
A Summary



+ one counter

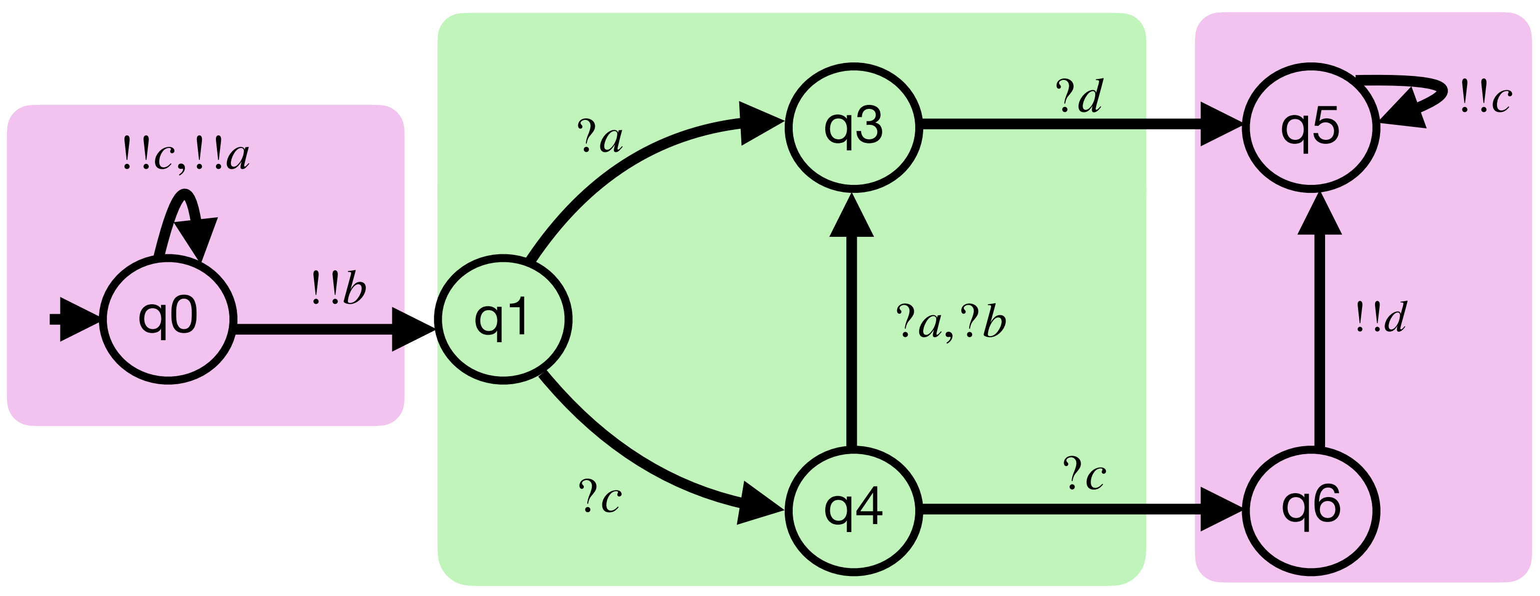


A Summary

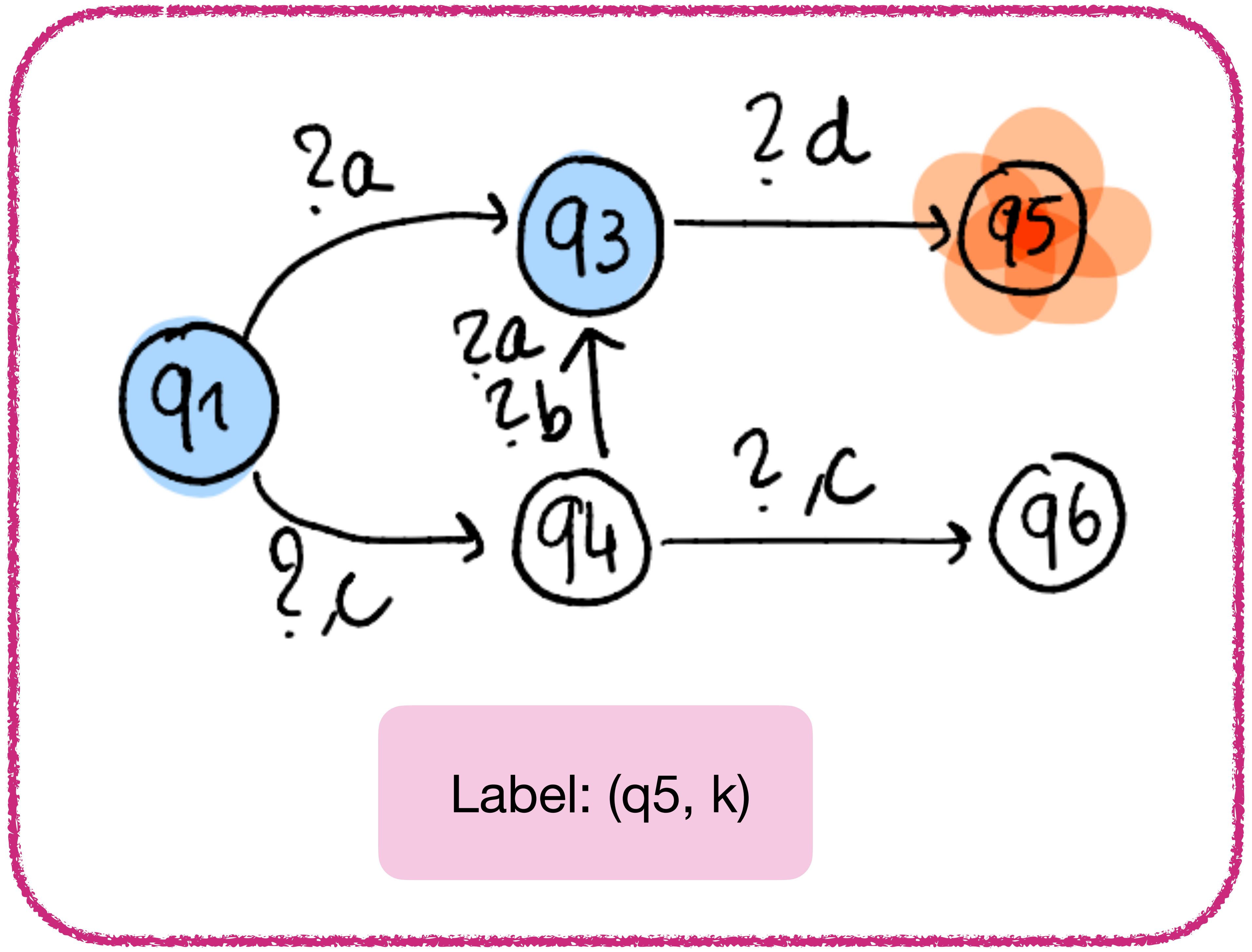


+ one counter

- Some processes are present on q1 and q3,
- the next action state they will reach is q5 and
- they will reach q5 at the same time

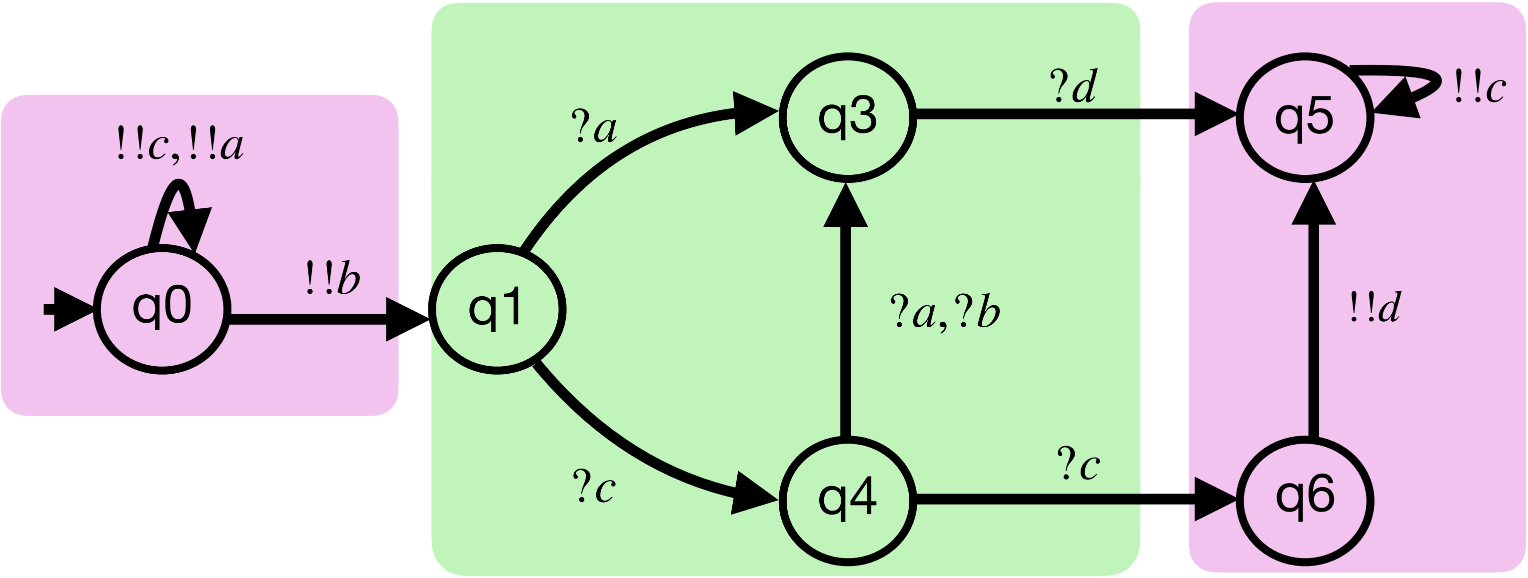


A Summary

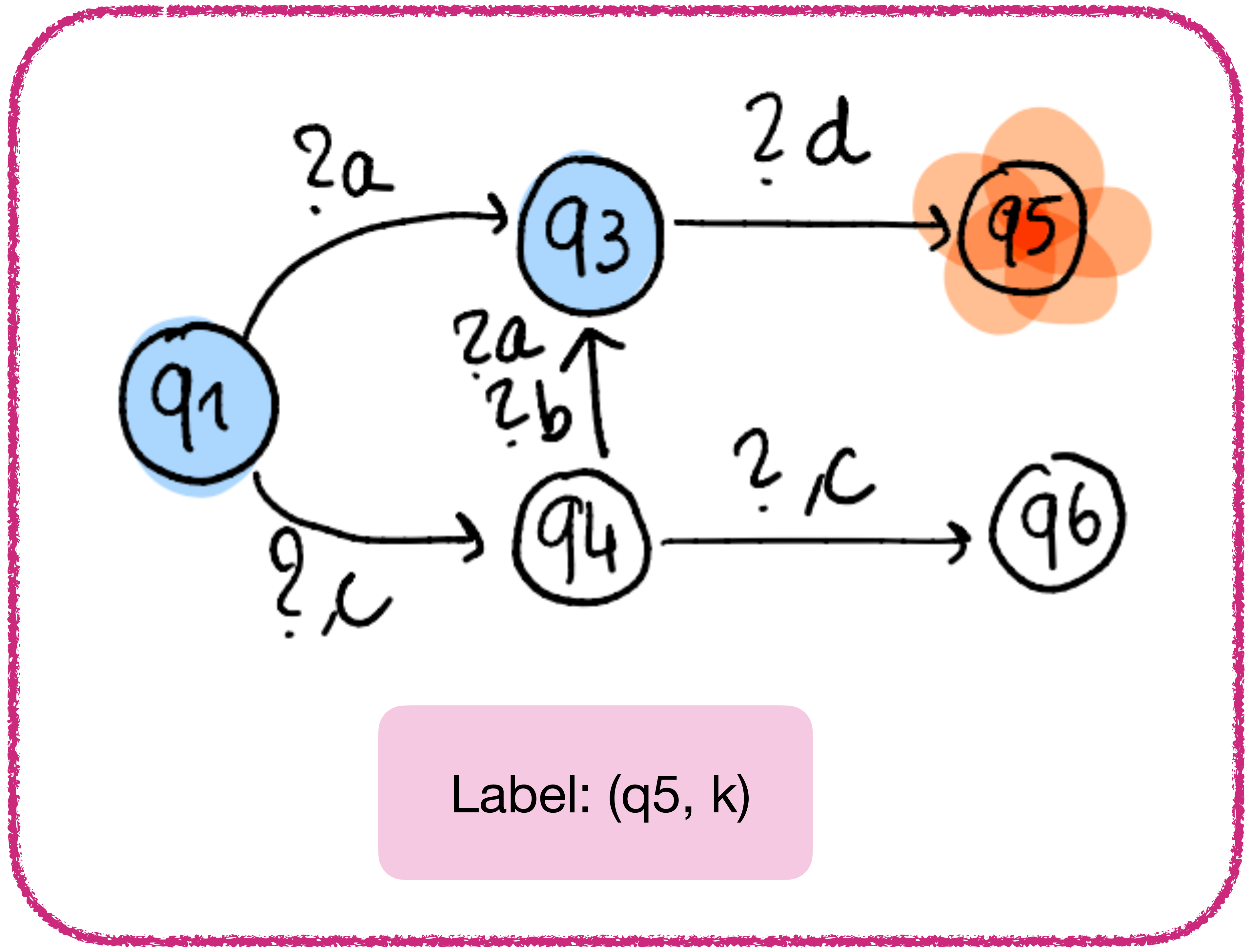


+ one counter

- Some processes are present on q1 and q3,
- the next action state they will reach is q5 and
- they will reach q5 at the same time



A Summary

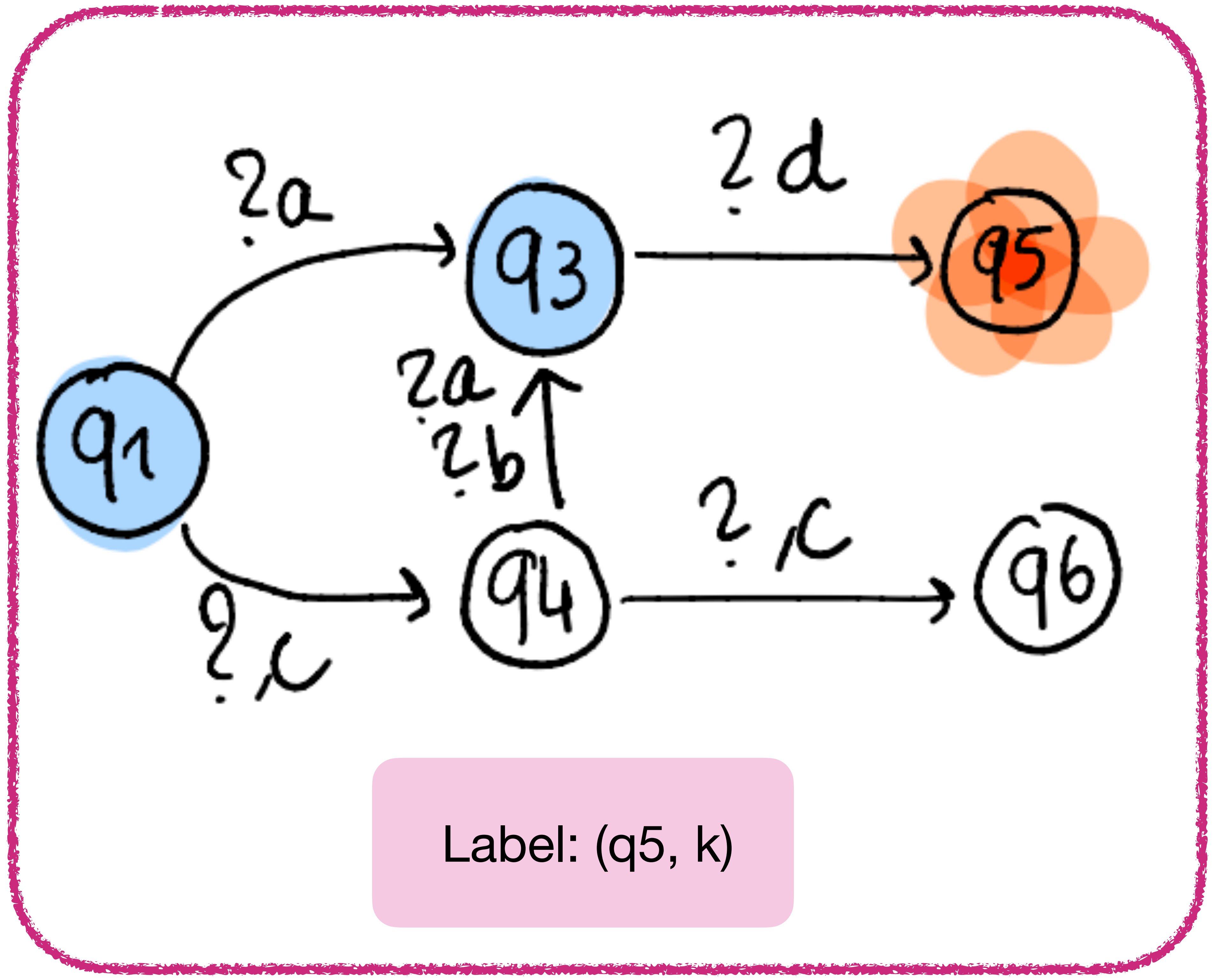


$1 \leq k \leq$ #(waiting states)

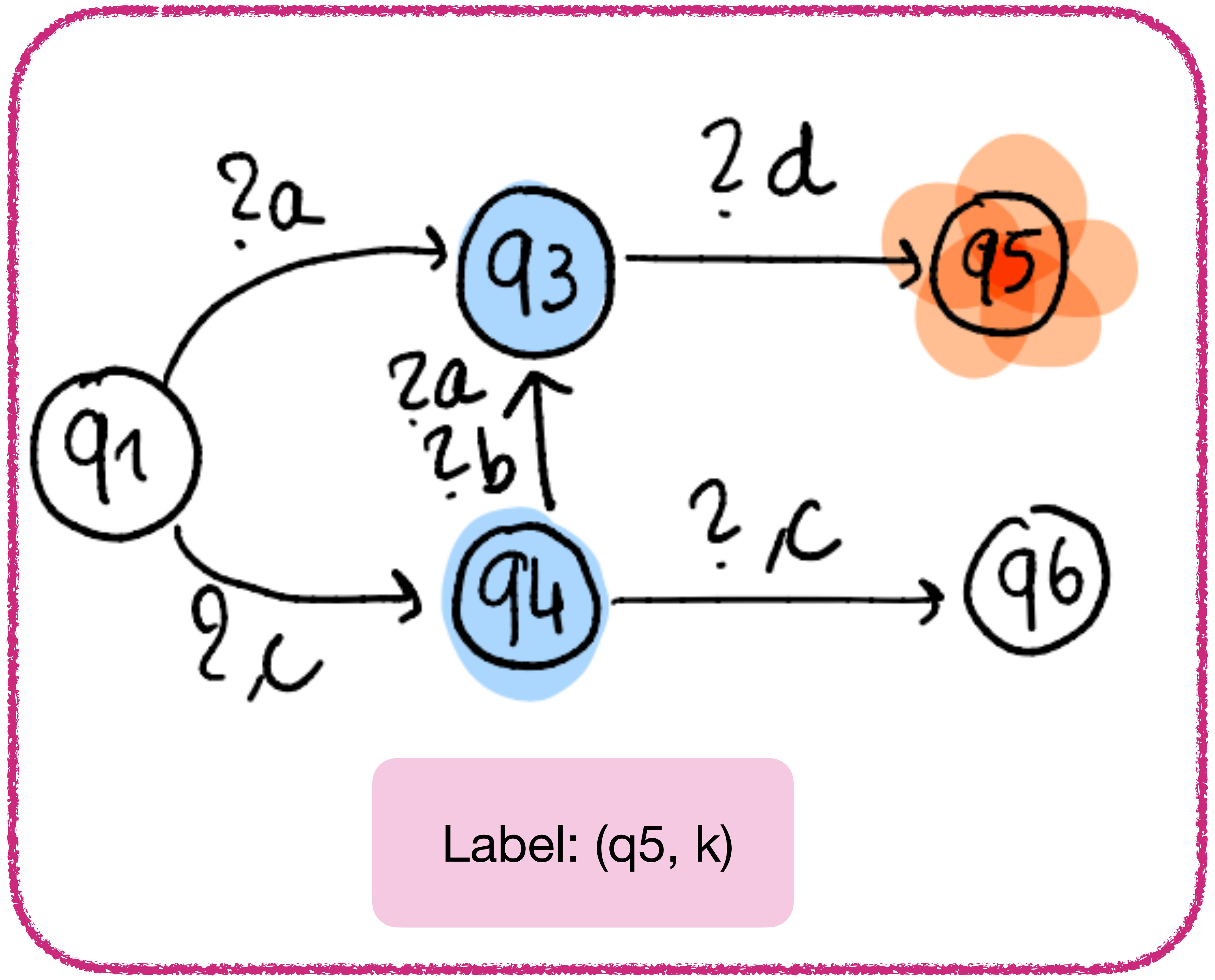
+ one counter

- Some processes are present on q1 and q3,
- the next action state they will reach is q5 and
- they will reach q5 at the same time

Broadcast and Summary

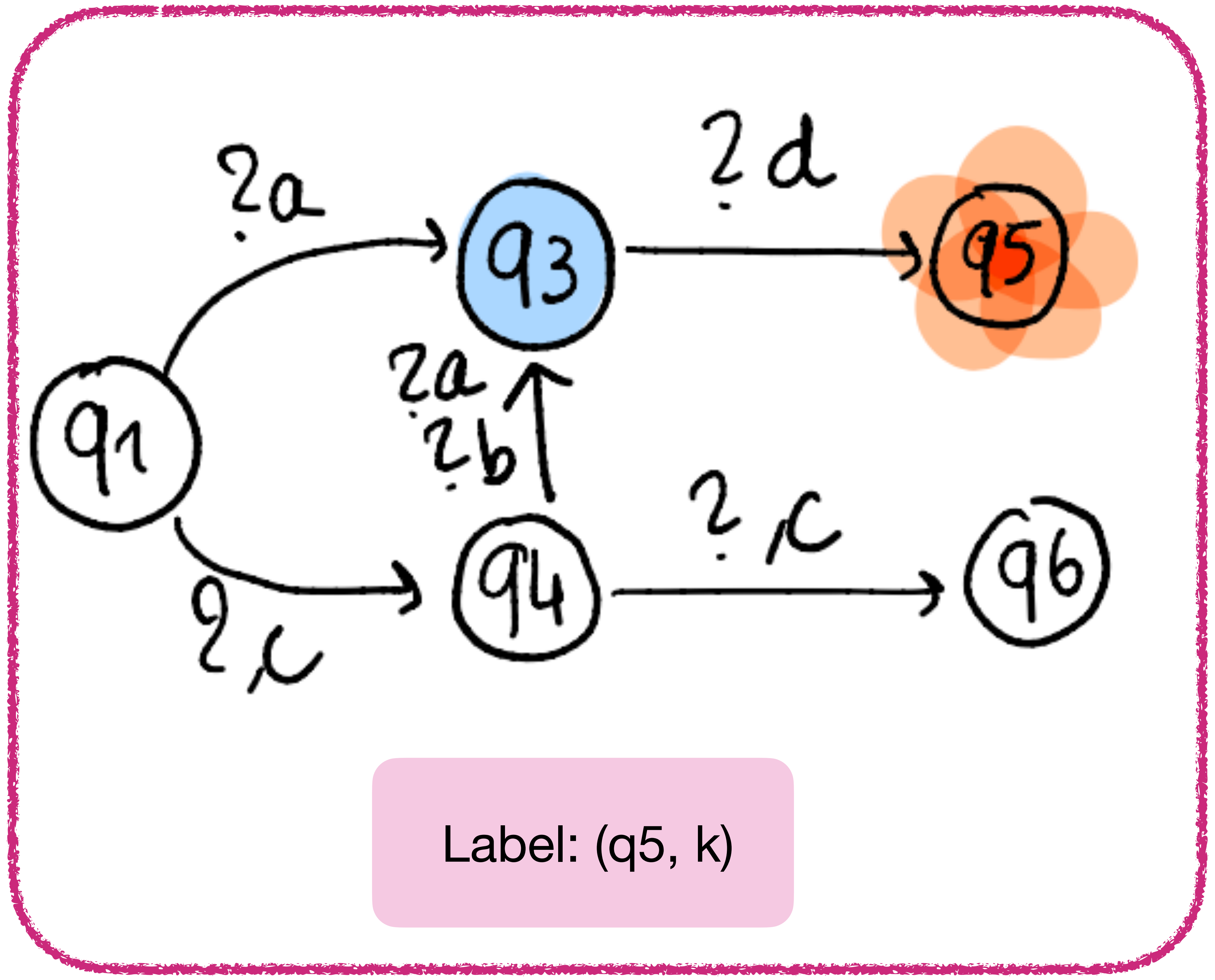


Broadcast and Summary



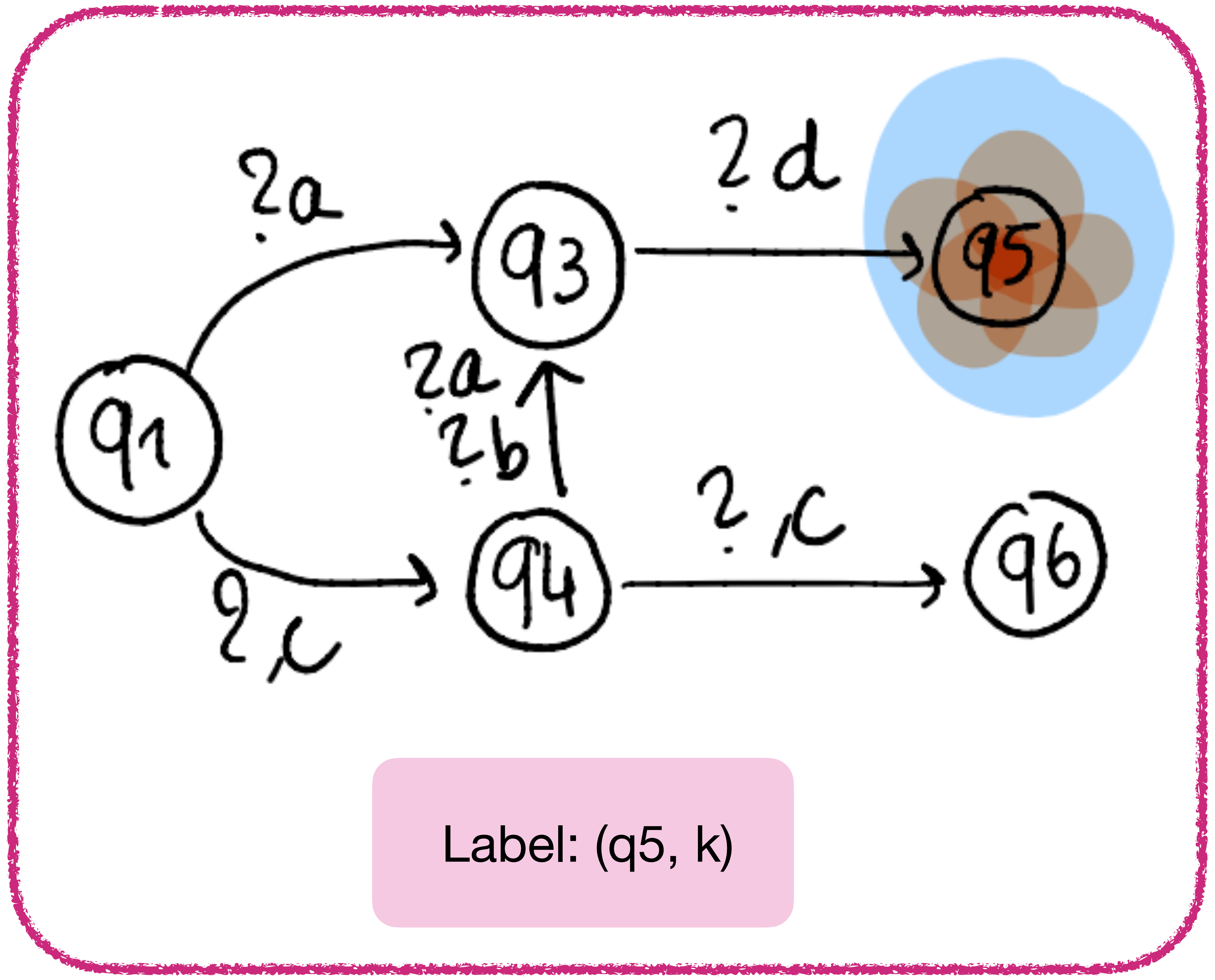
!!c

Broadcast and Summary



!!a

Broadcast and Summary

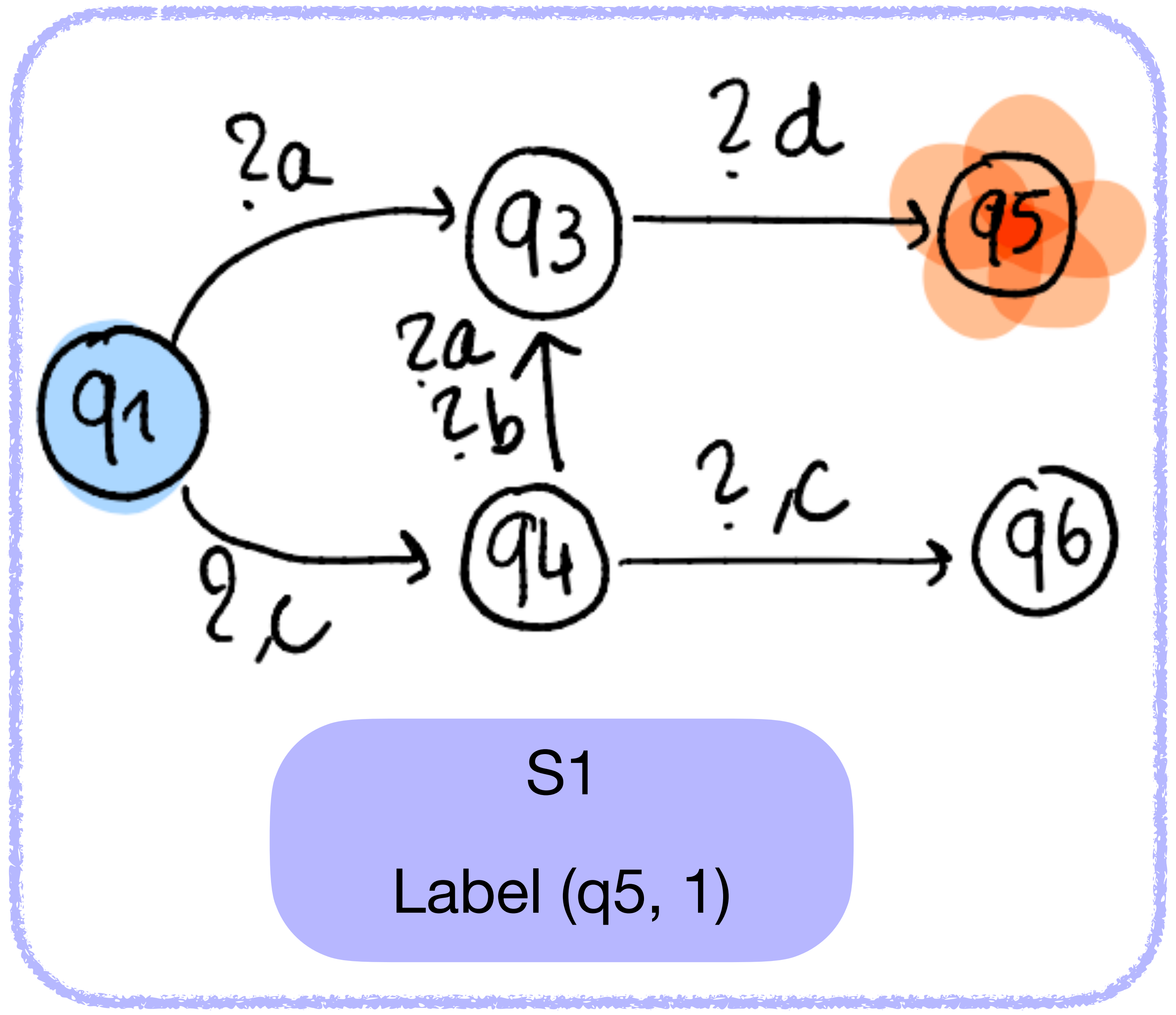


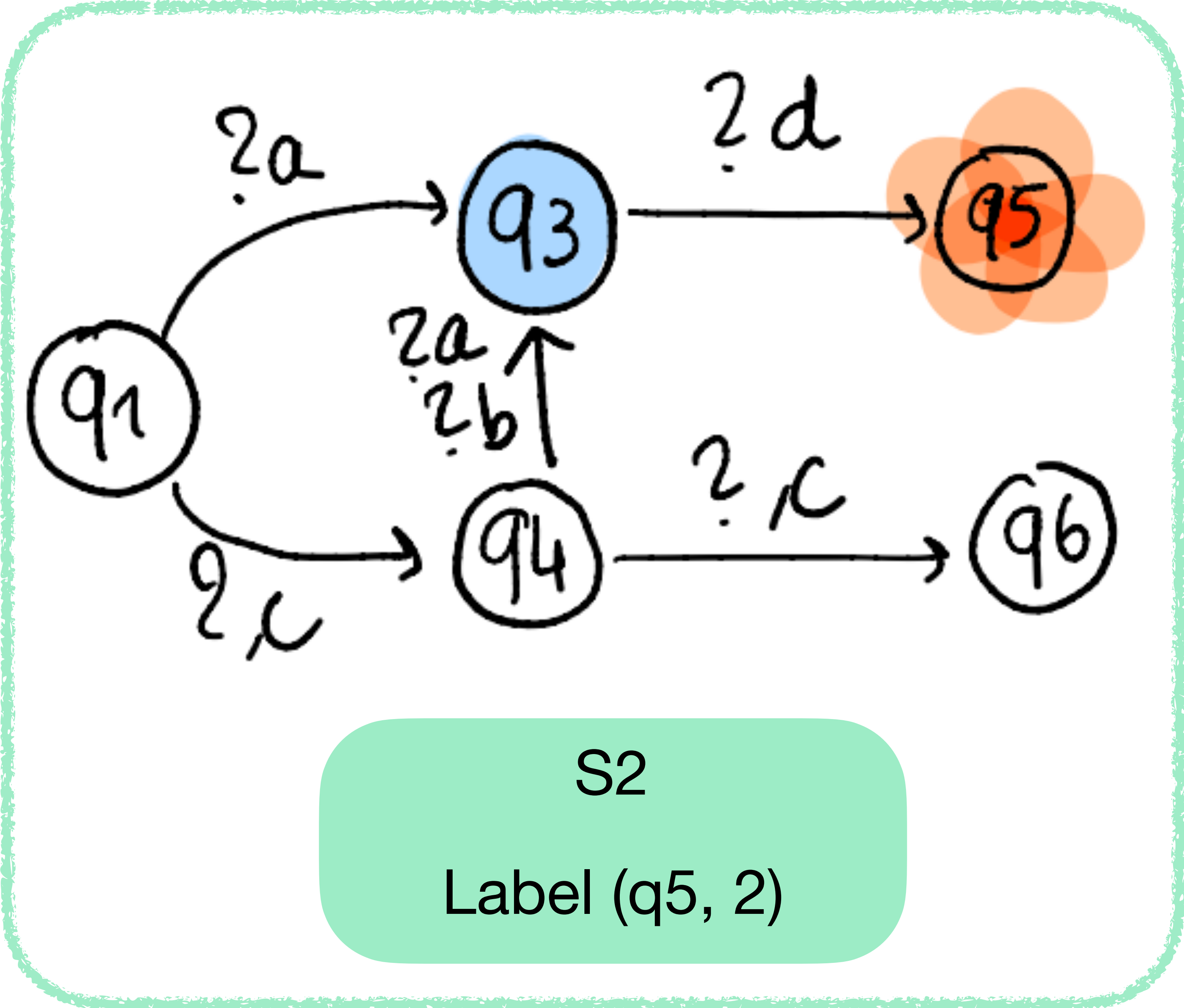
!!d

Summaries in VASS

- Location = **coherent** set of summaries
- Counters = one counter per action states + one counter per summary label
- In the VASS, we keep track of processes on action states, and guess some summaries for the processes on waiting states

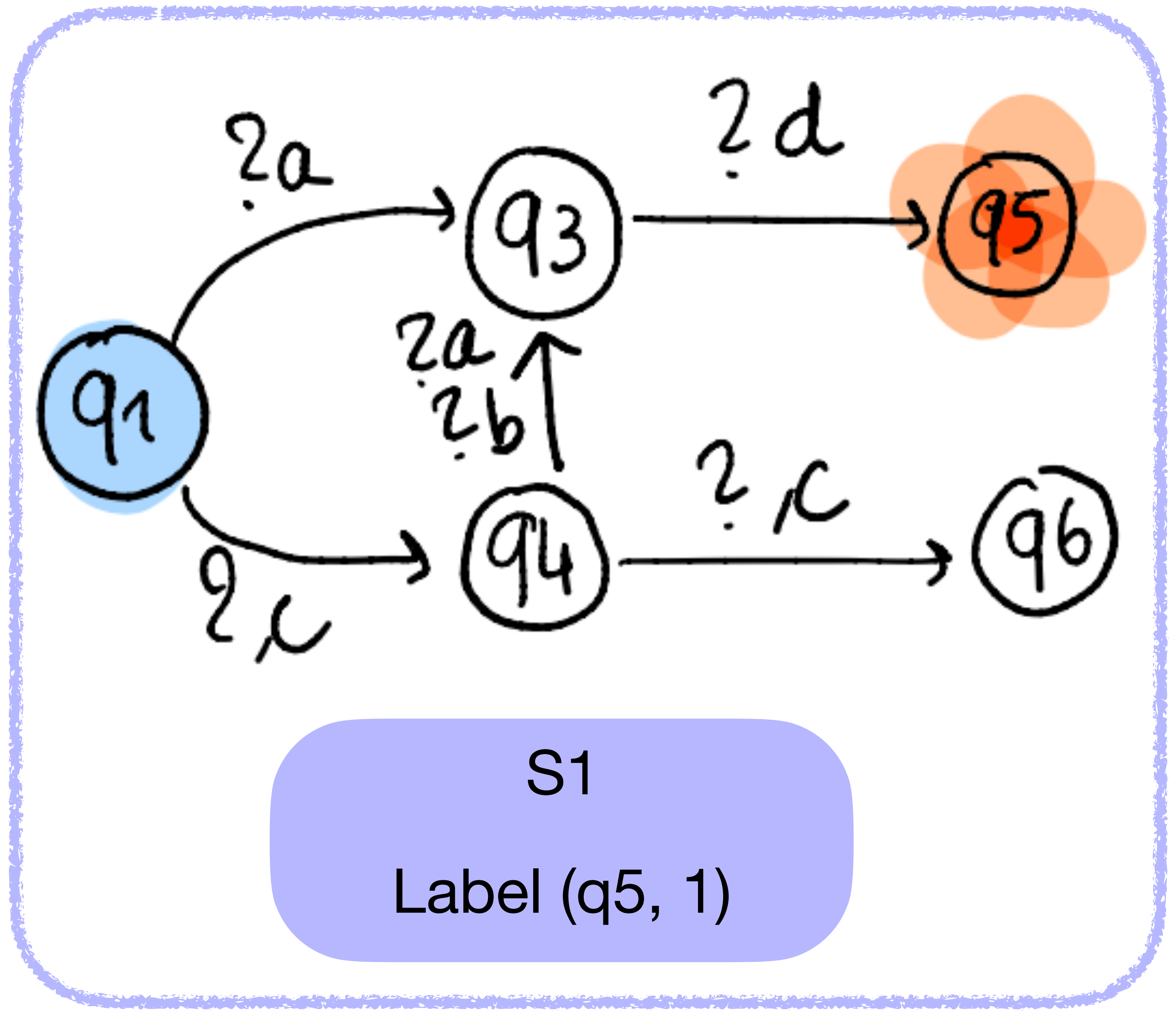
Coherent* sets of Summaries

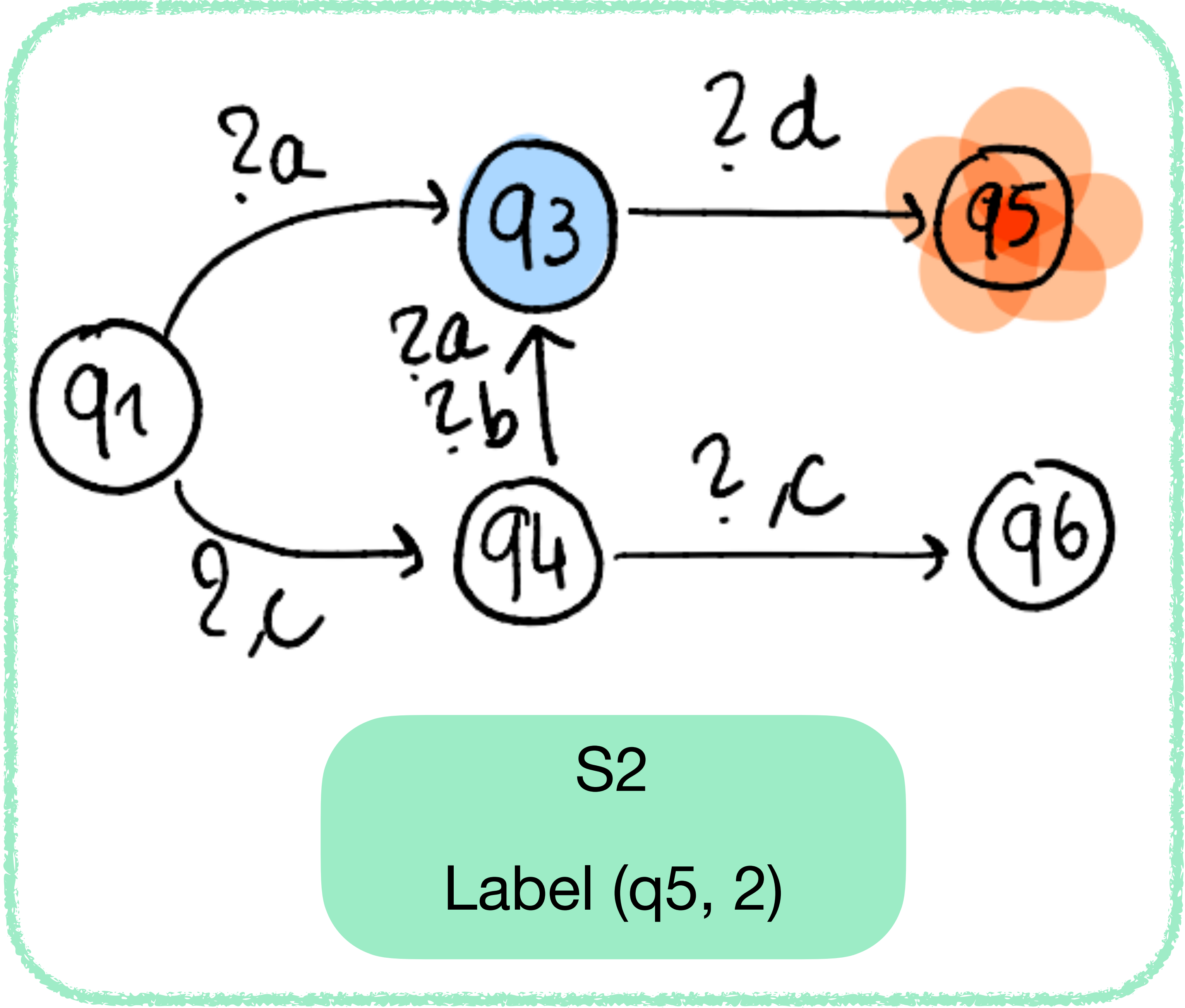




*** two processes on different summaries don’t reach the same state OR reach the same state but not at the same time**

Coherent* sets of Summaries

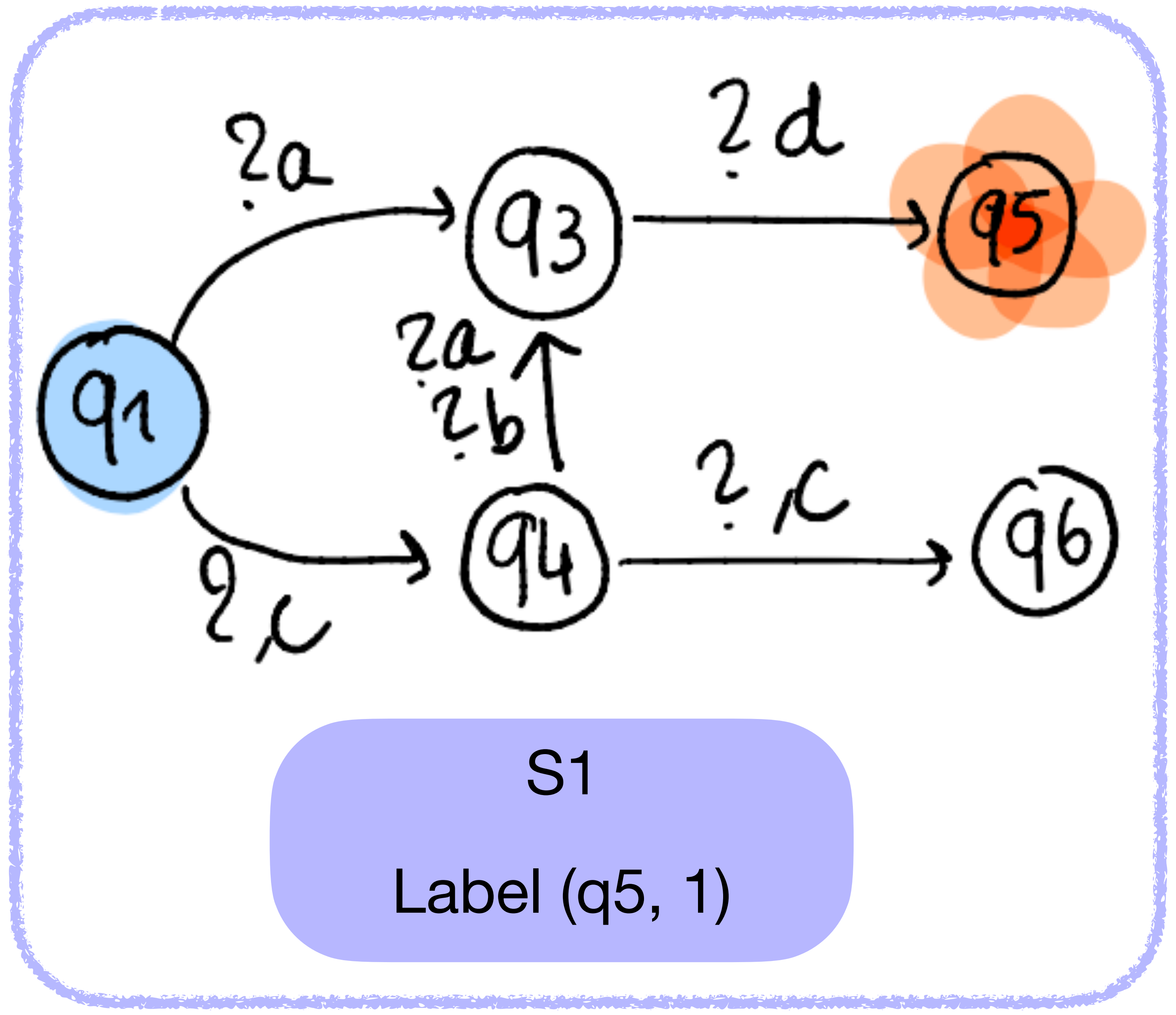


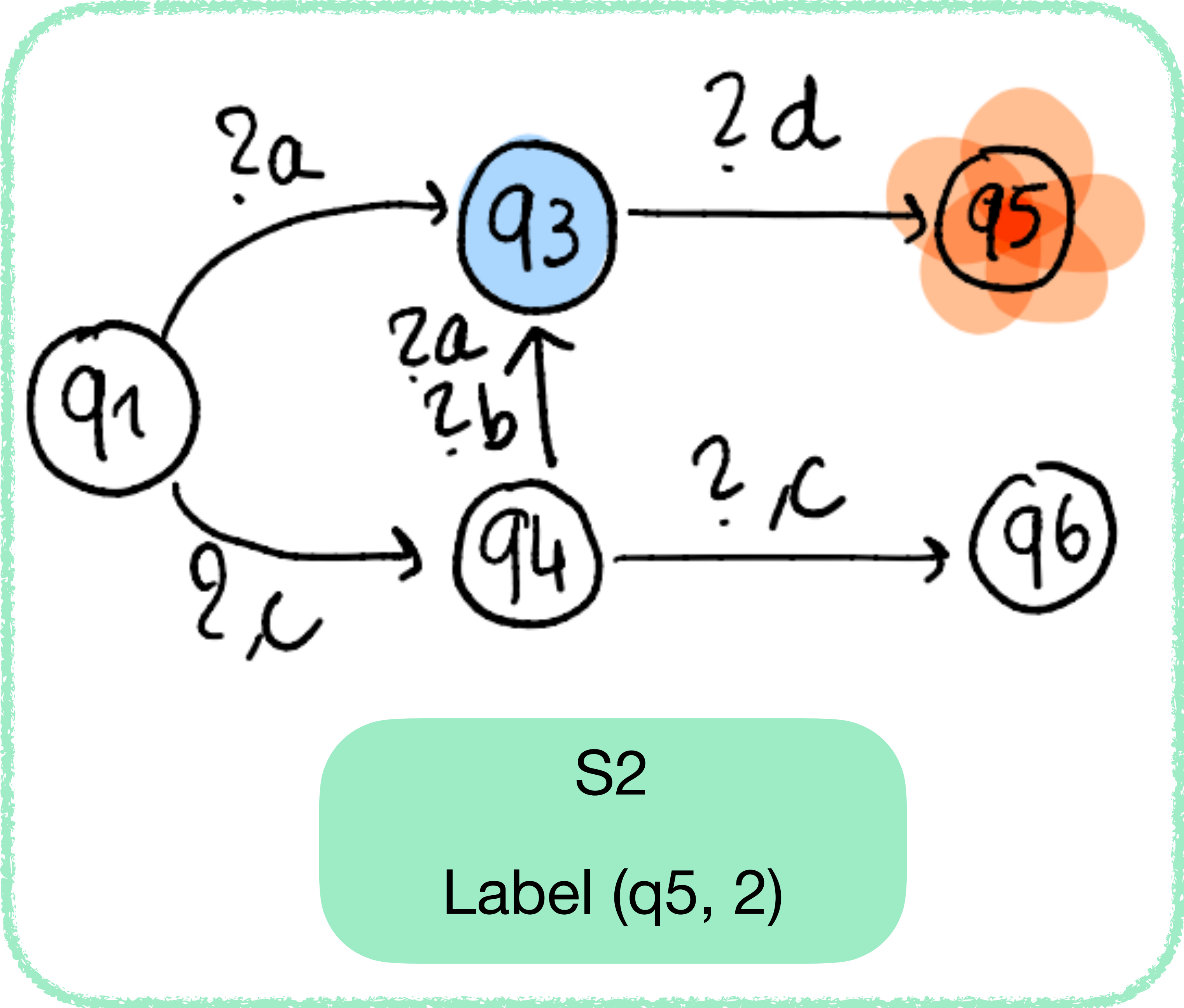


ex: !!d !!a !!d

*** two processes on different summaries don't reach the same state OR reach the same state but not at the same time**

Coherent* sets of Summaries

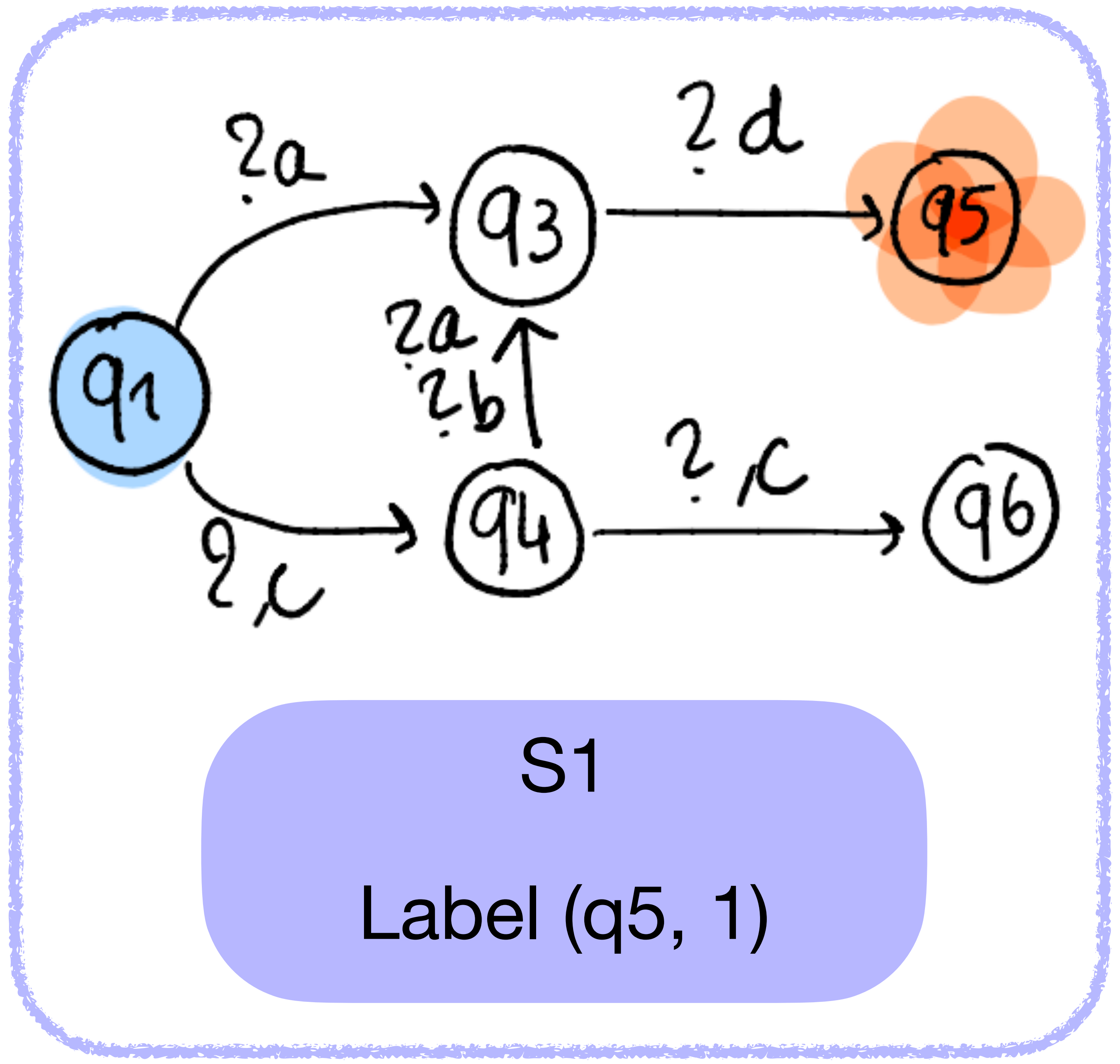


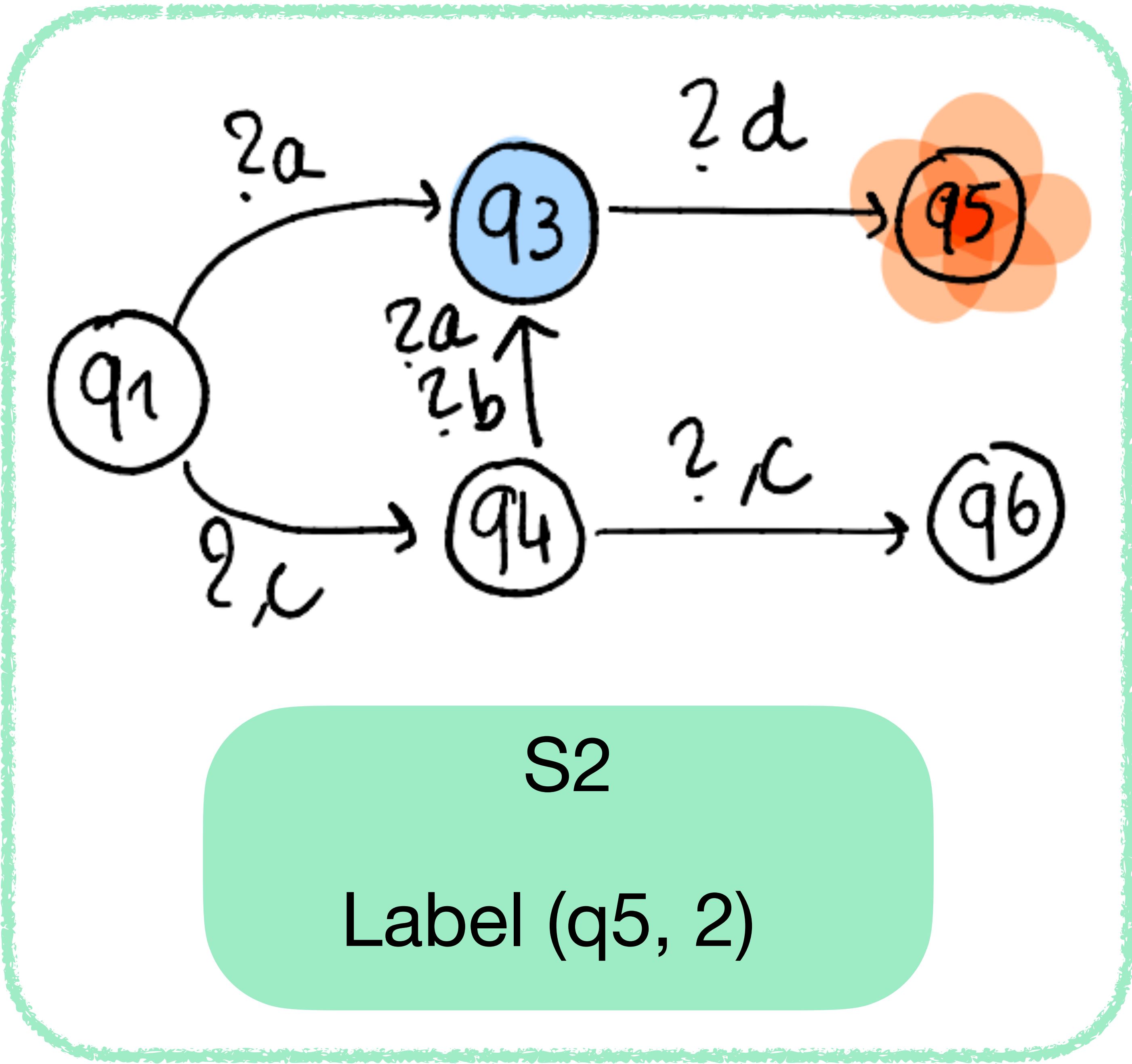


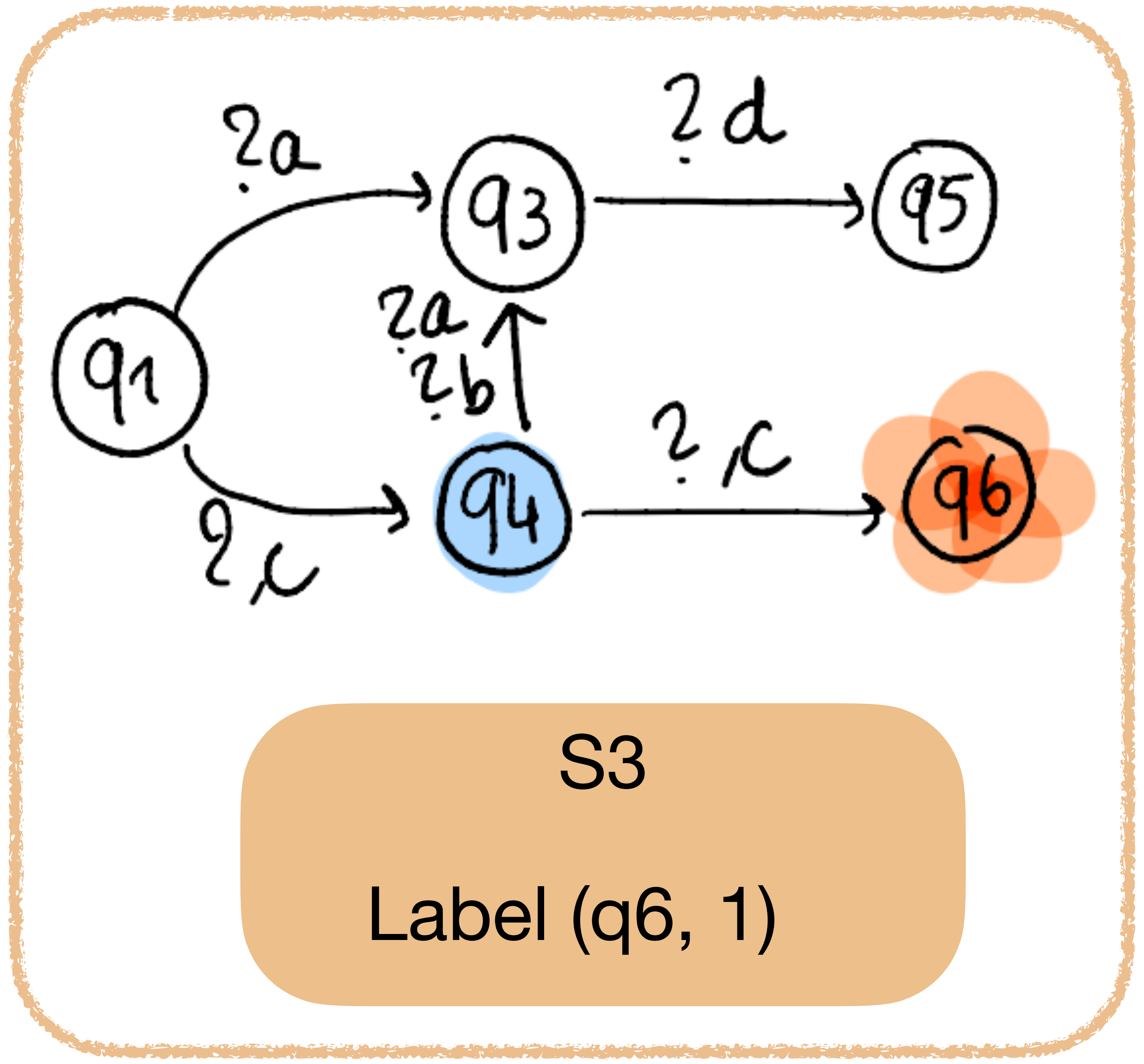
Coherent!

* two processes on dif... ...maries don’t reach the same state
OR reach the same state but not at the same time

Coherent* sets of Summaries

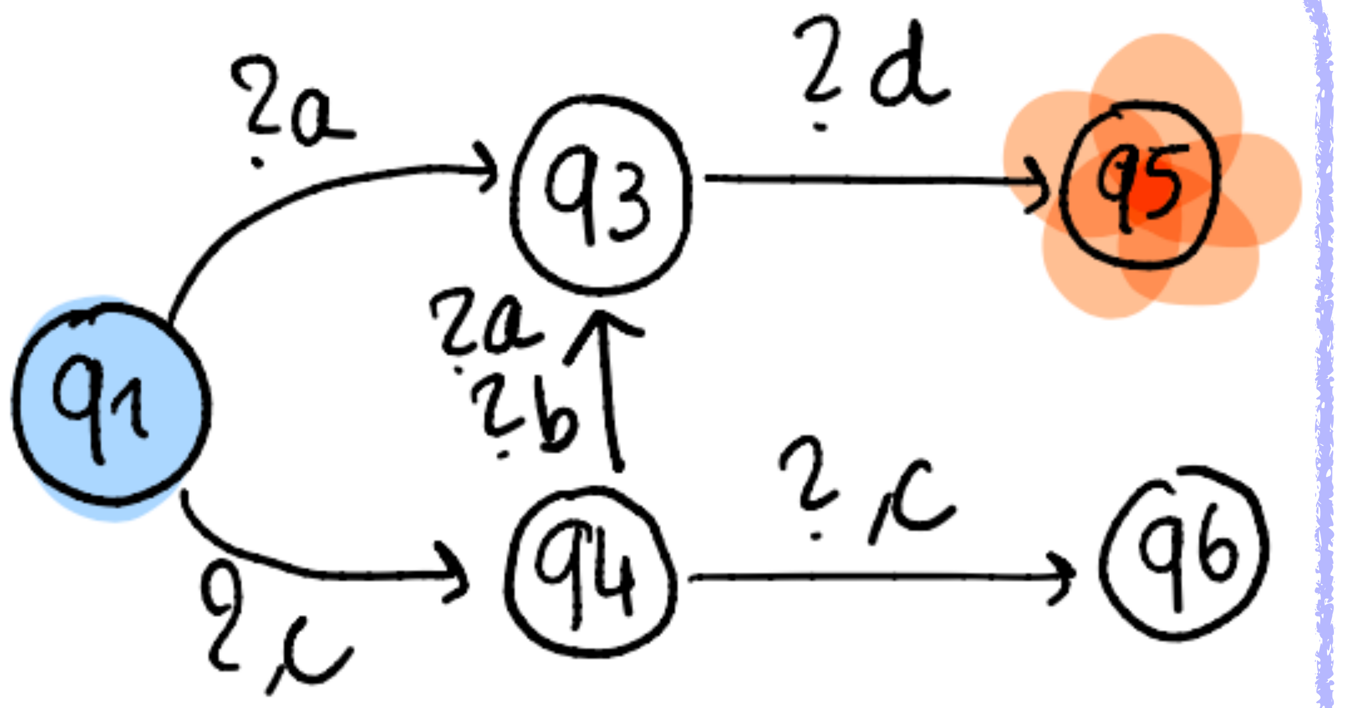






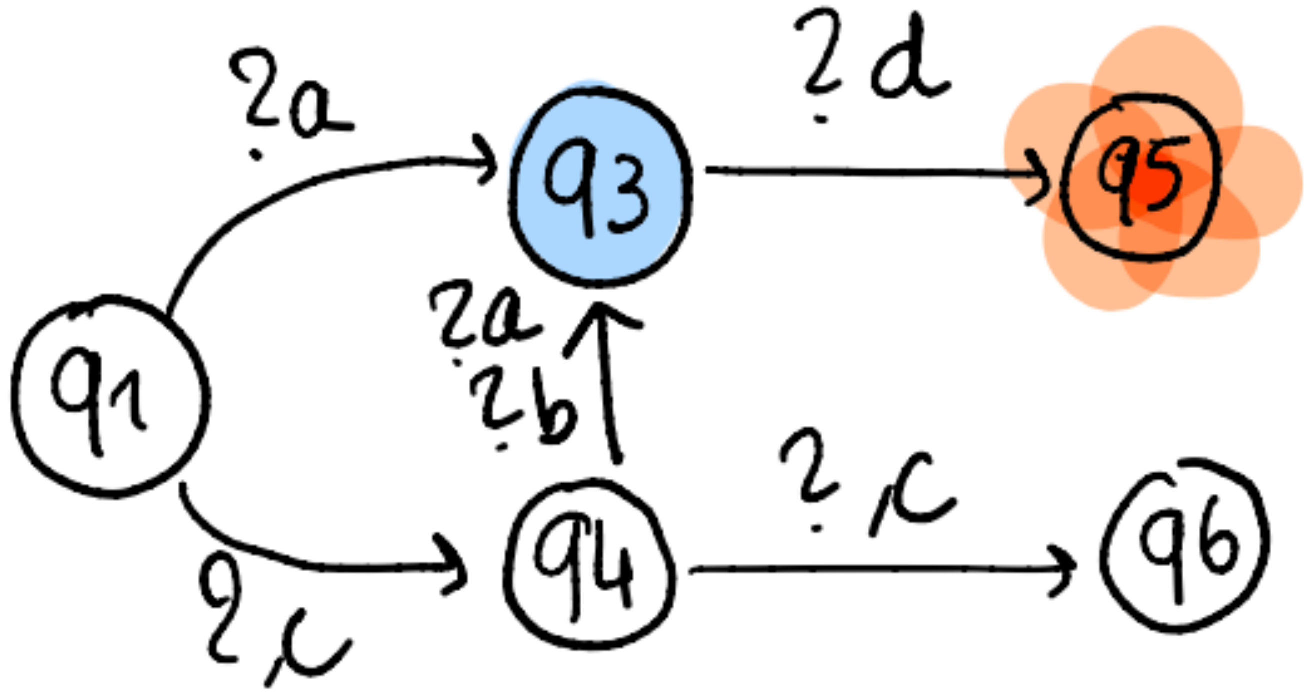
*** two processes on different summaries don’t reach the same state OR reach the same state but not at the same time**

Coherent* sets of Summaries



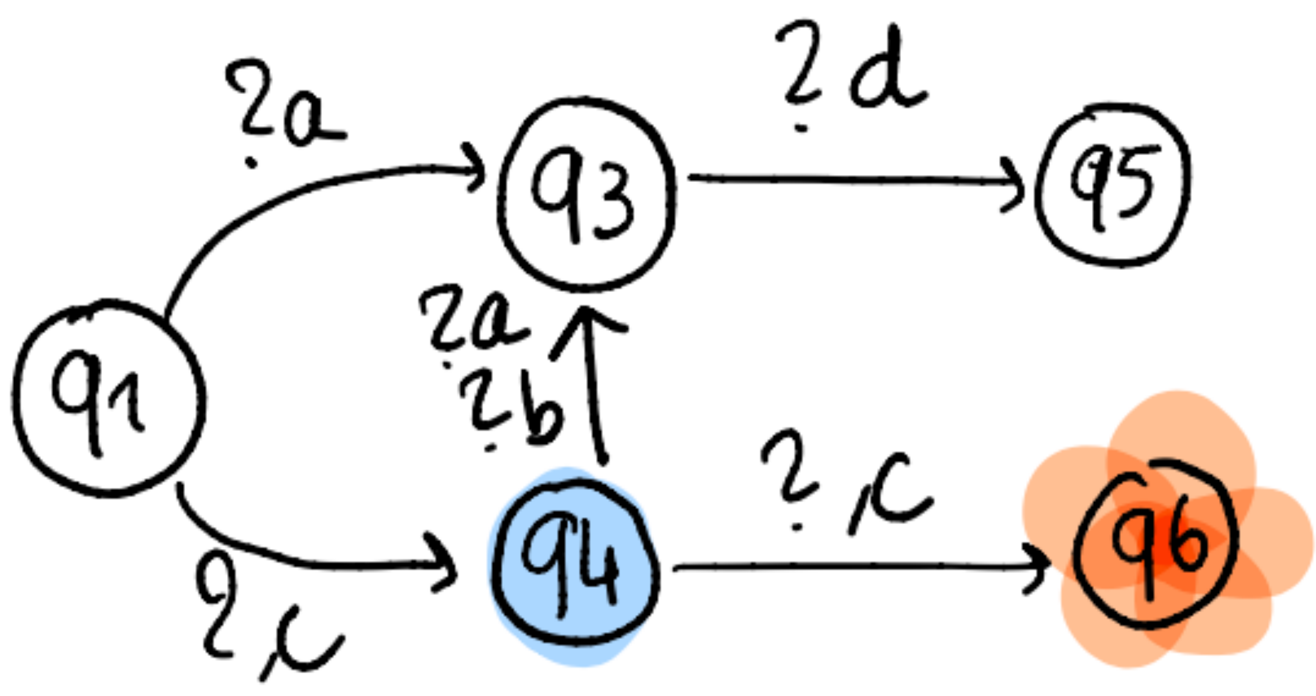
S1

Label (q5, 1)



S2

Label (q5, 2)



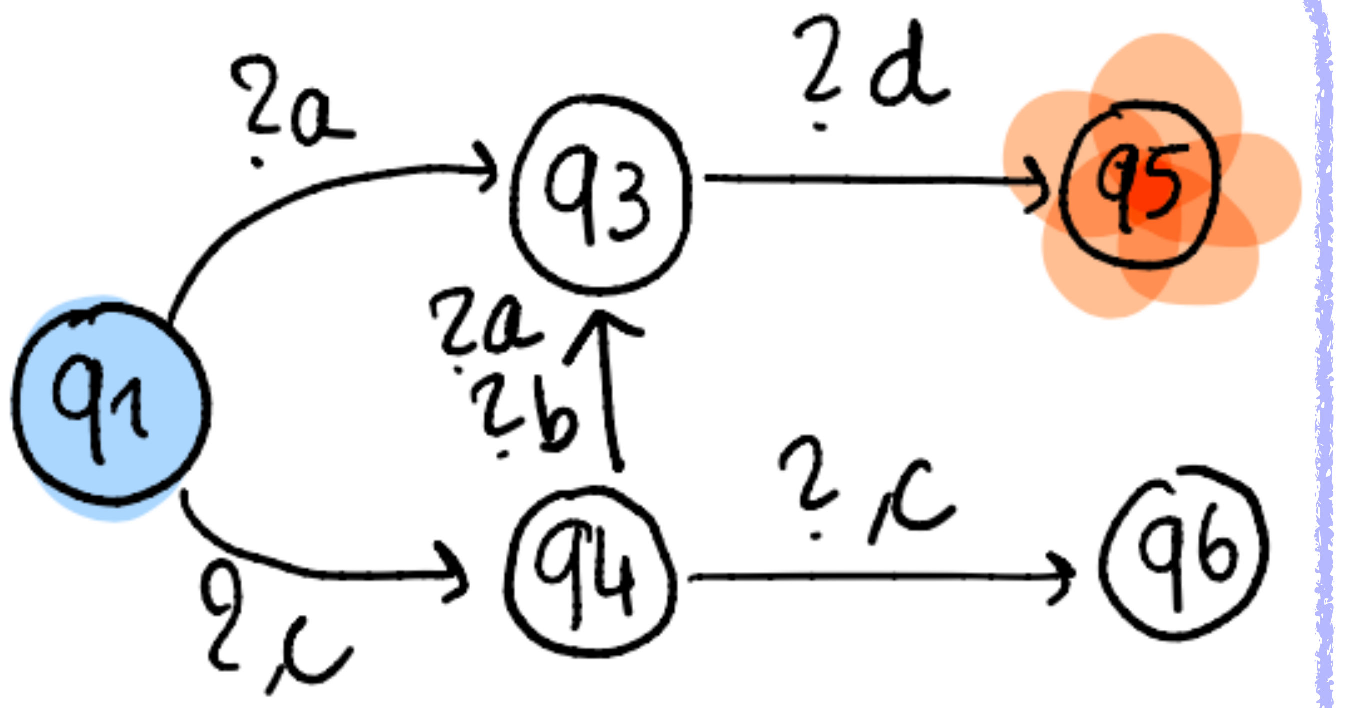
S3

Label (q6, 1)

ex: !!d !!c !!b !!d

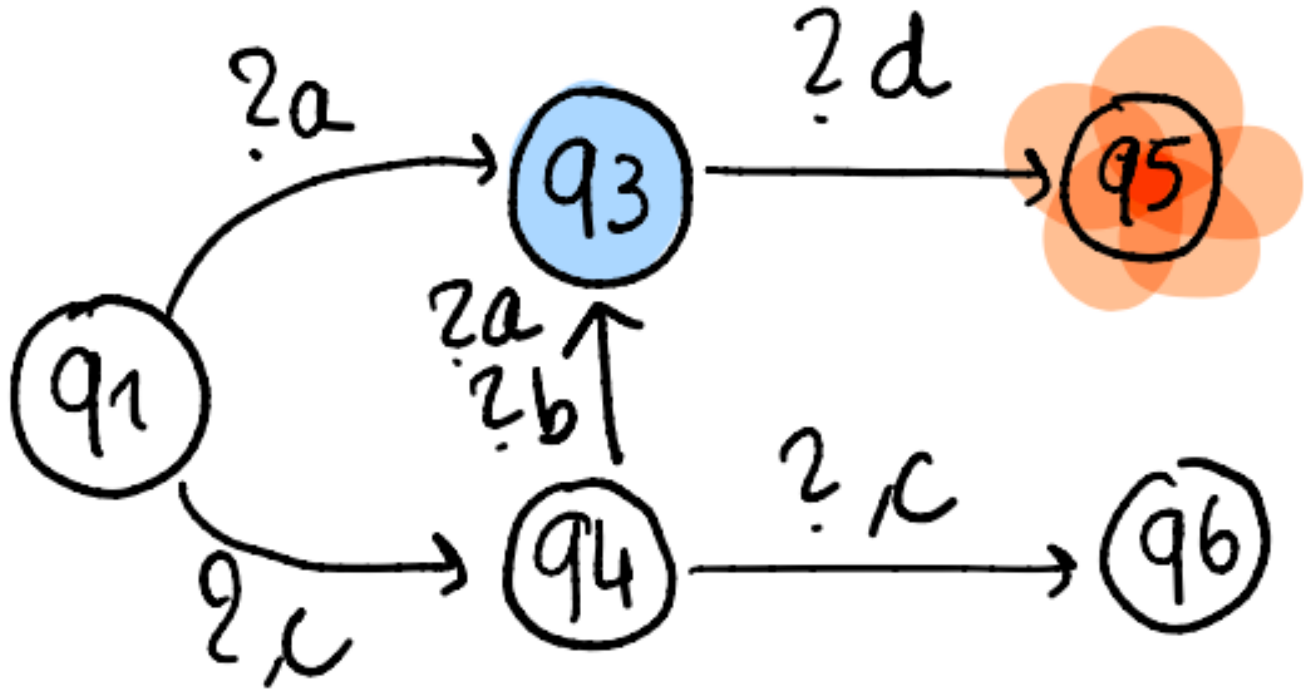
*** two processes on different summaries don't reach the same state OR reach the same state but not at the same time**

Coherent* sets of Summaries



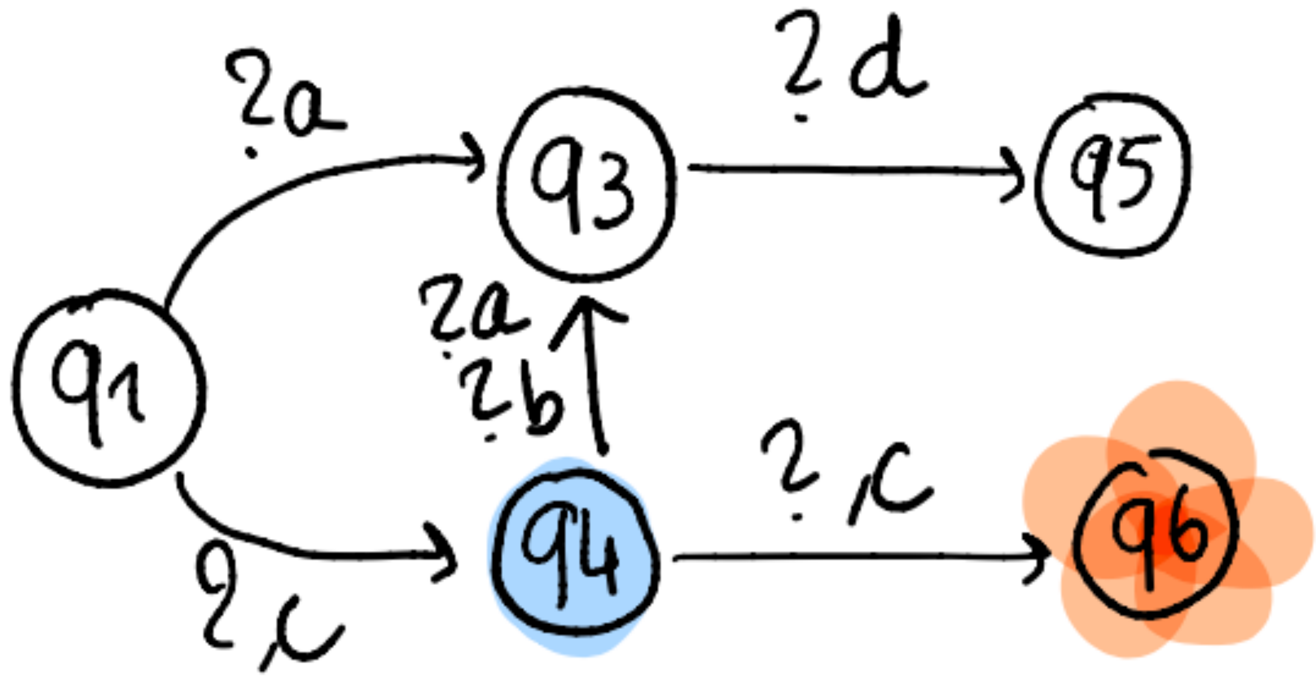
S1

Label (q5, 1)



S2

Label (q5, 2)



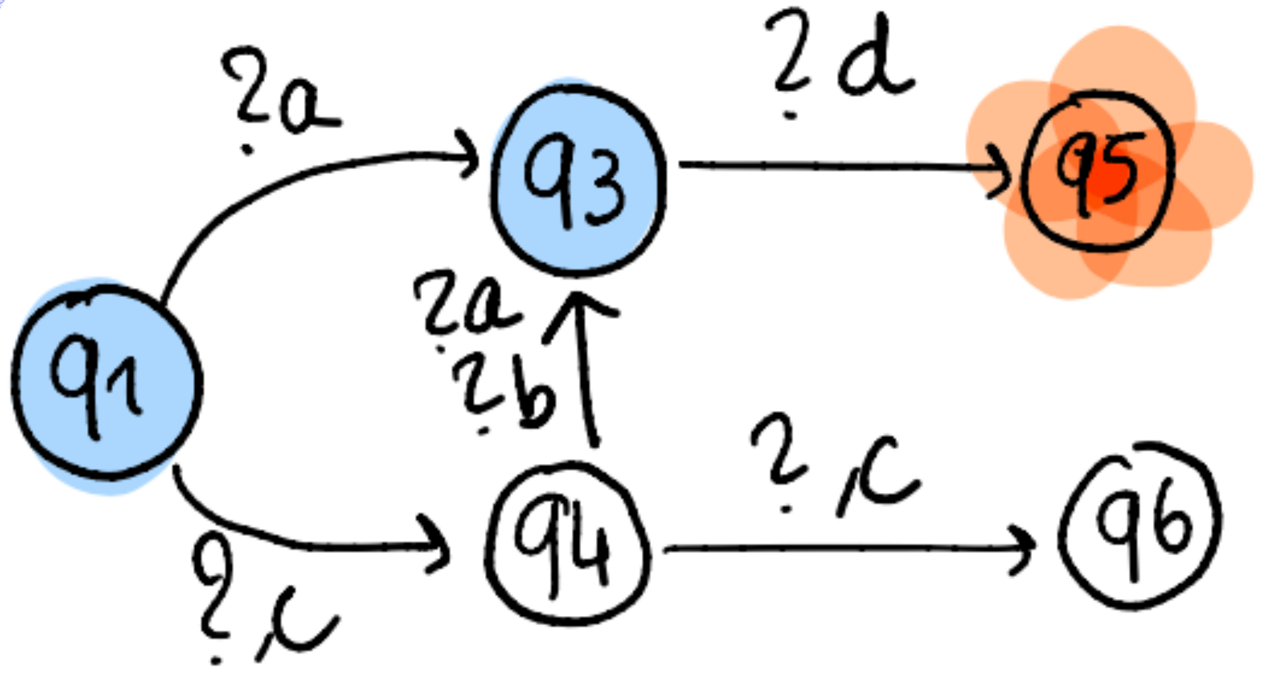
S3

Label (q6, 1)

Coherent again!

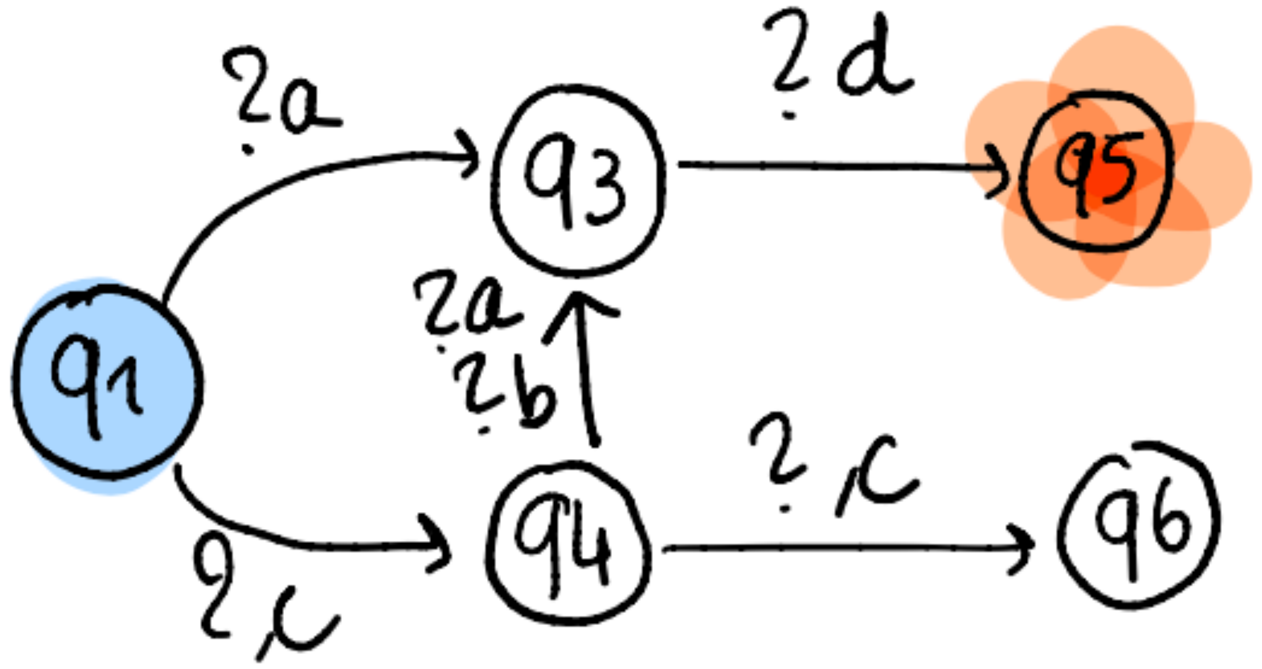
* two processes on different summaries don't reach the same state OR reach the same state but not at the same time

Coherent sets of Summaries



S1

Label (q5, 1)

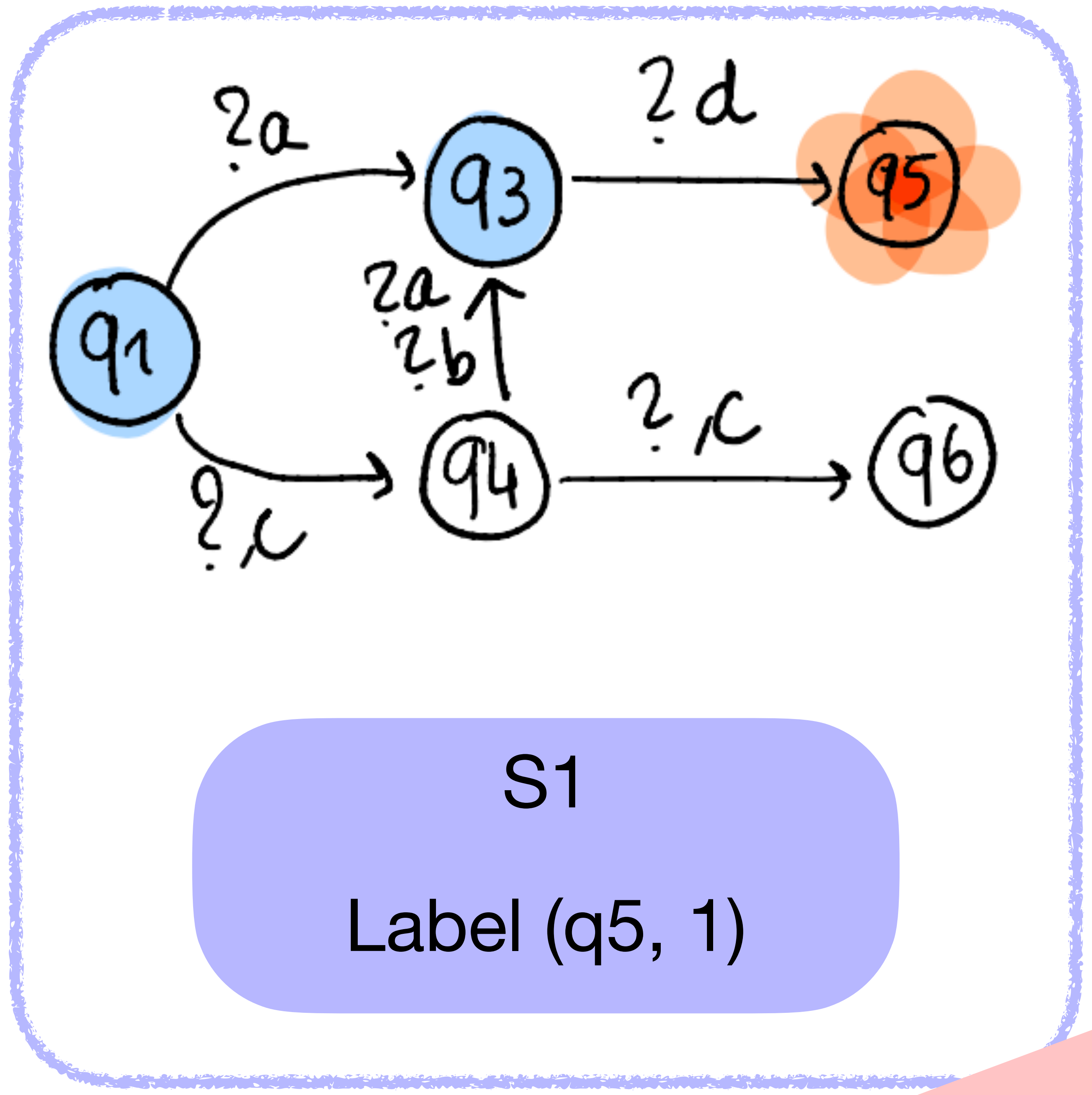


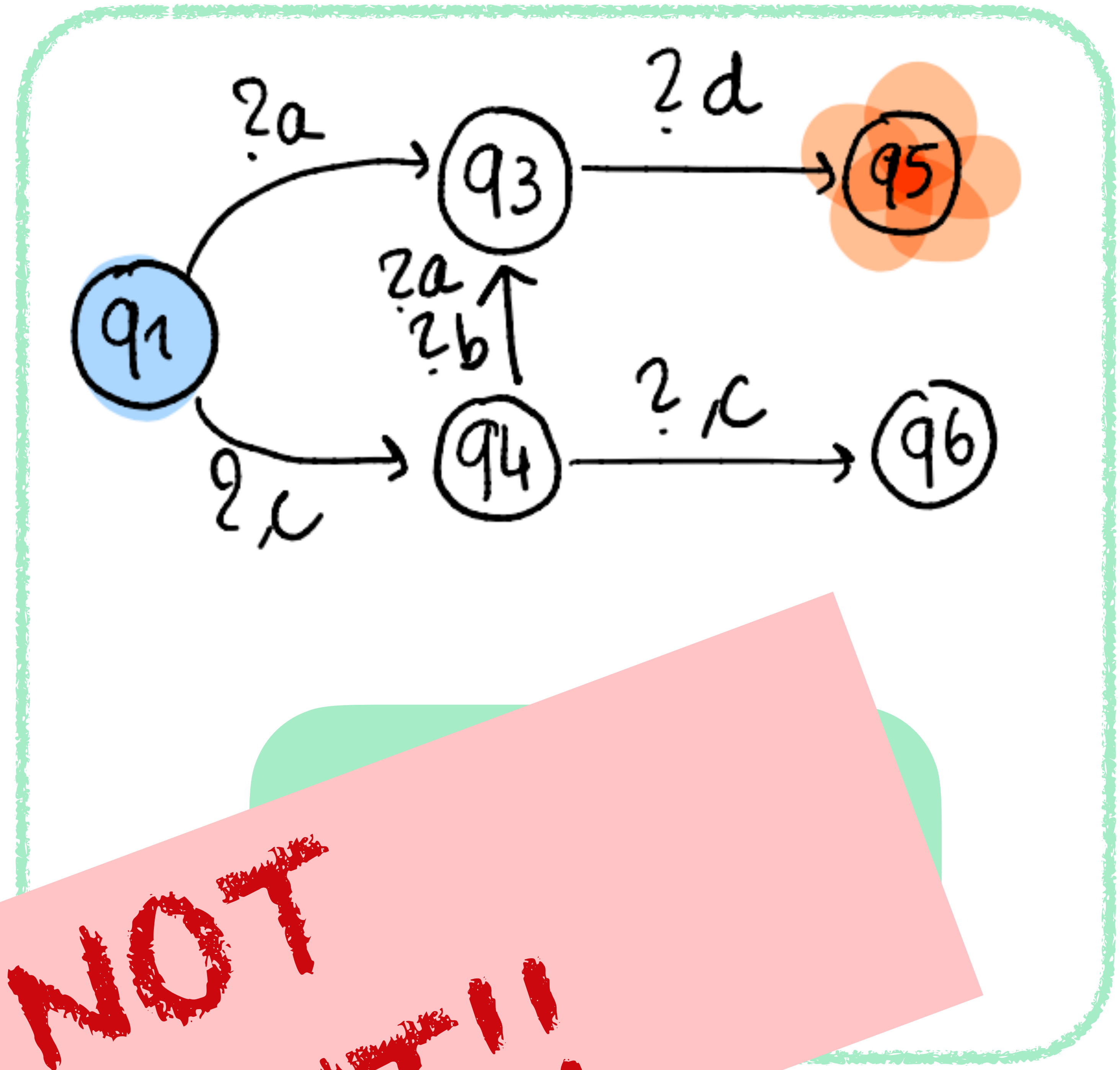
S2

Label (q5, 2)

* two processes on different summaries don’t reach the same state OR reach the same state but not at the same time

Coherent sets of Summaries



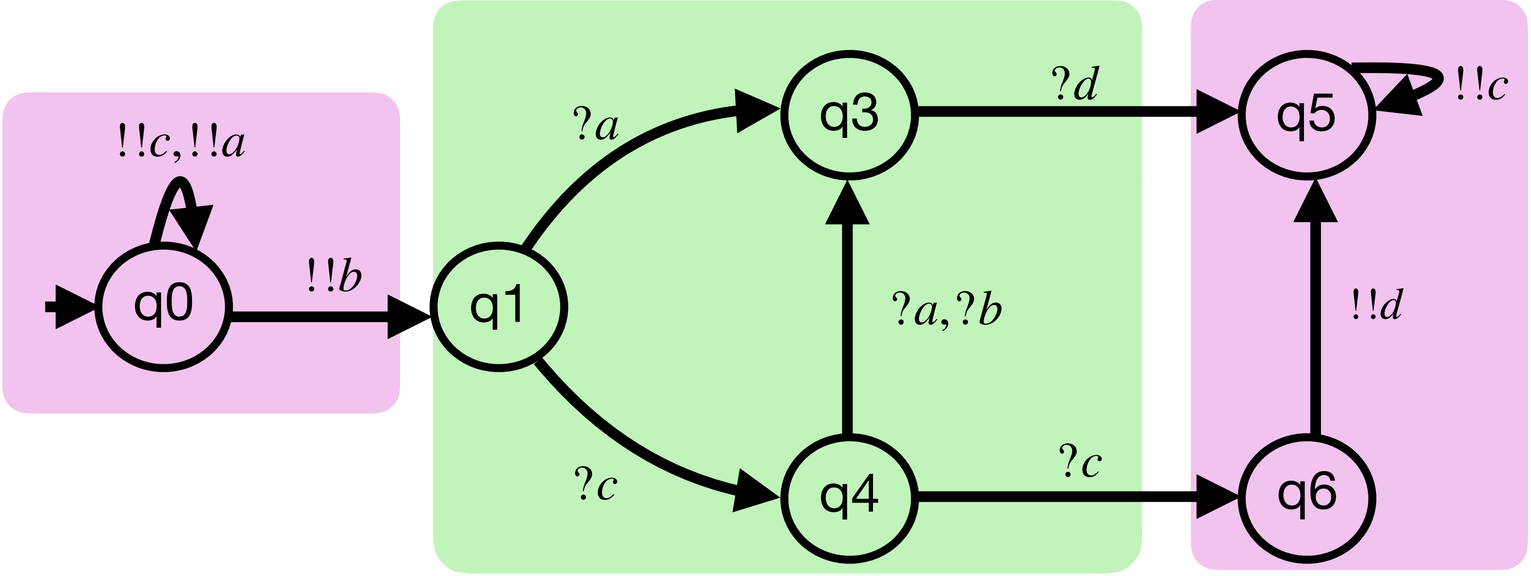


/!\ NOT COHERENT!!

* two processes on different summa… on't reach the same state OR reach the same state but not at the same time

At most #(waiting states) summaries per target states

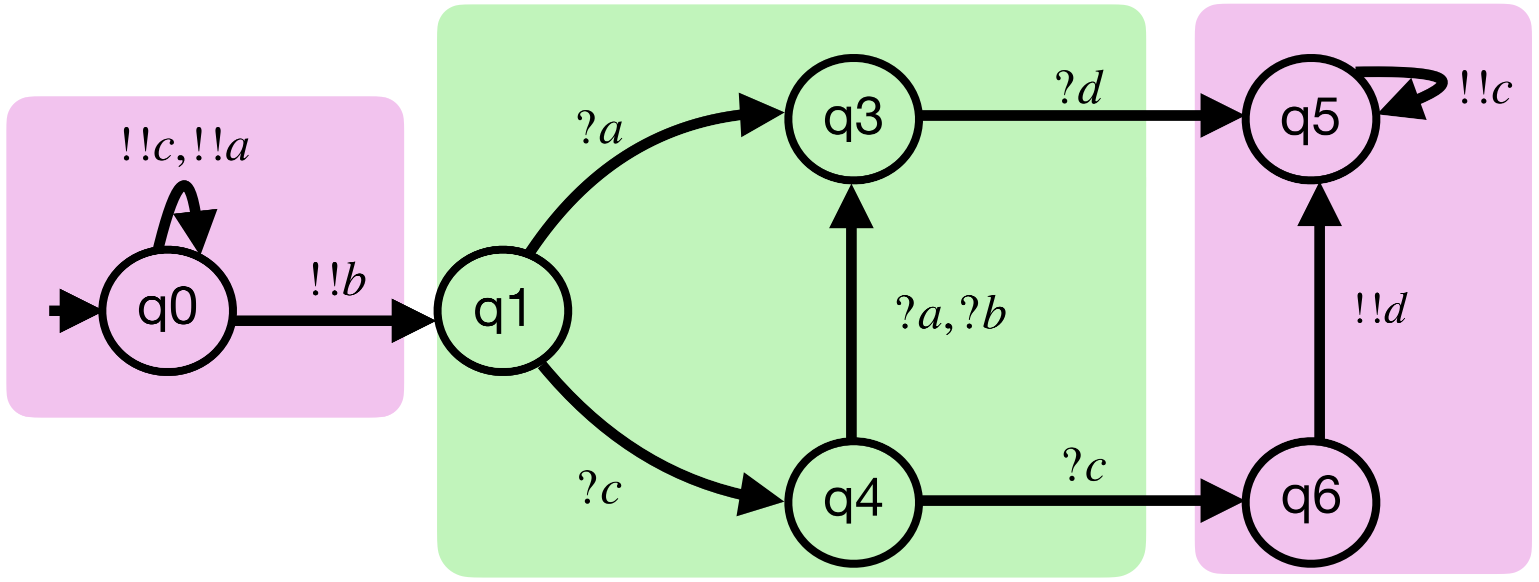
Creation of a Summary

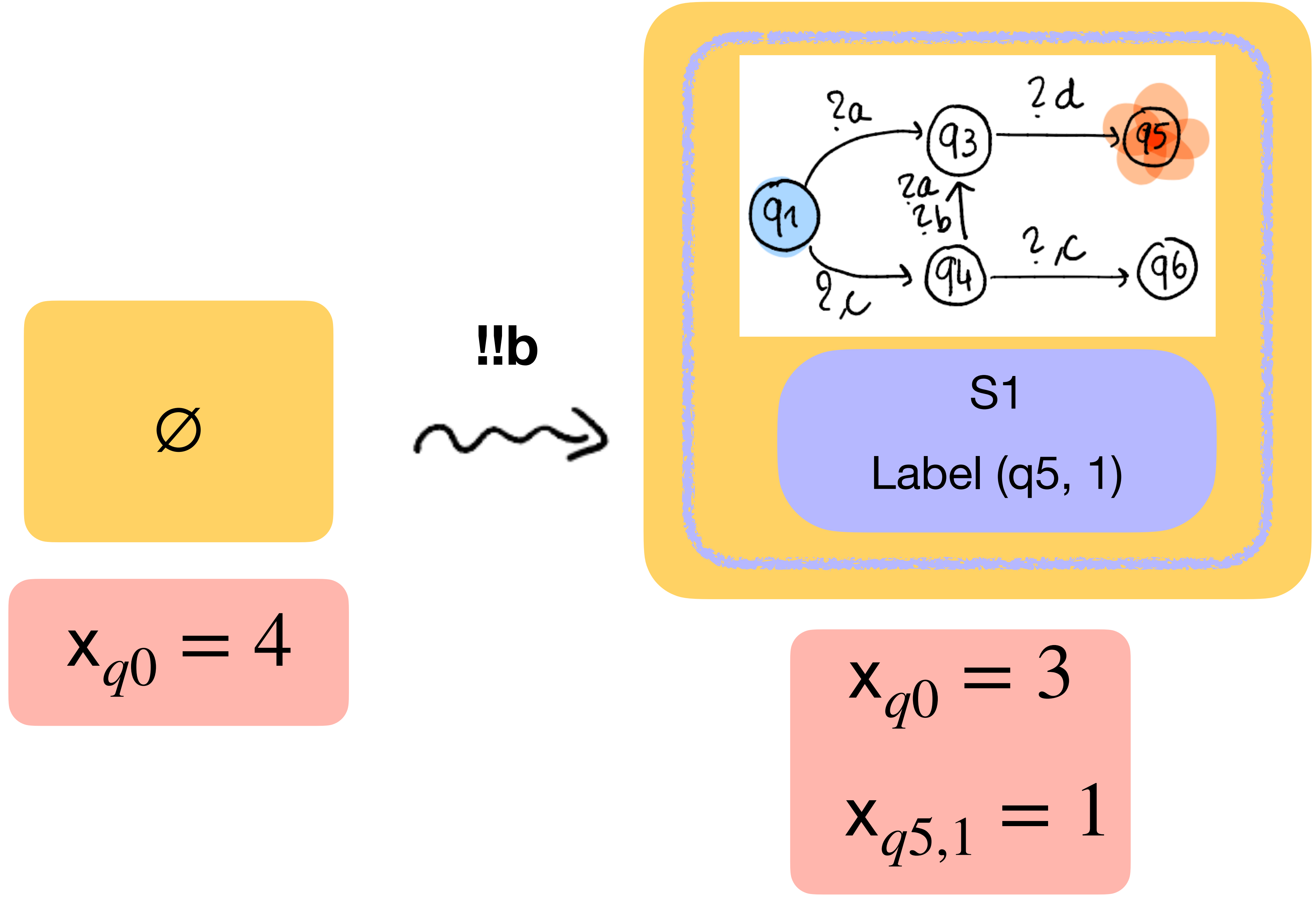


$\varnothing$

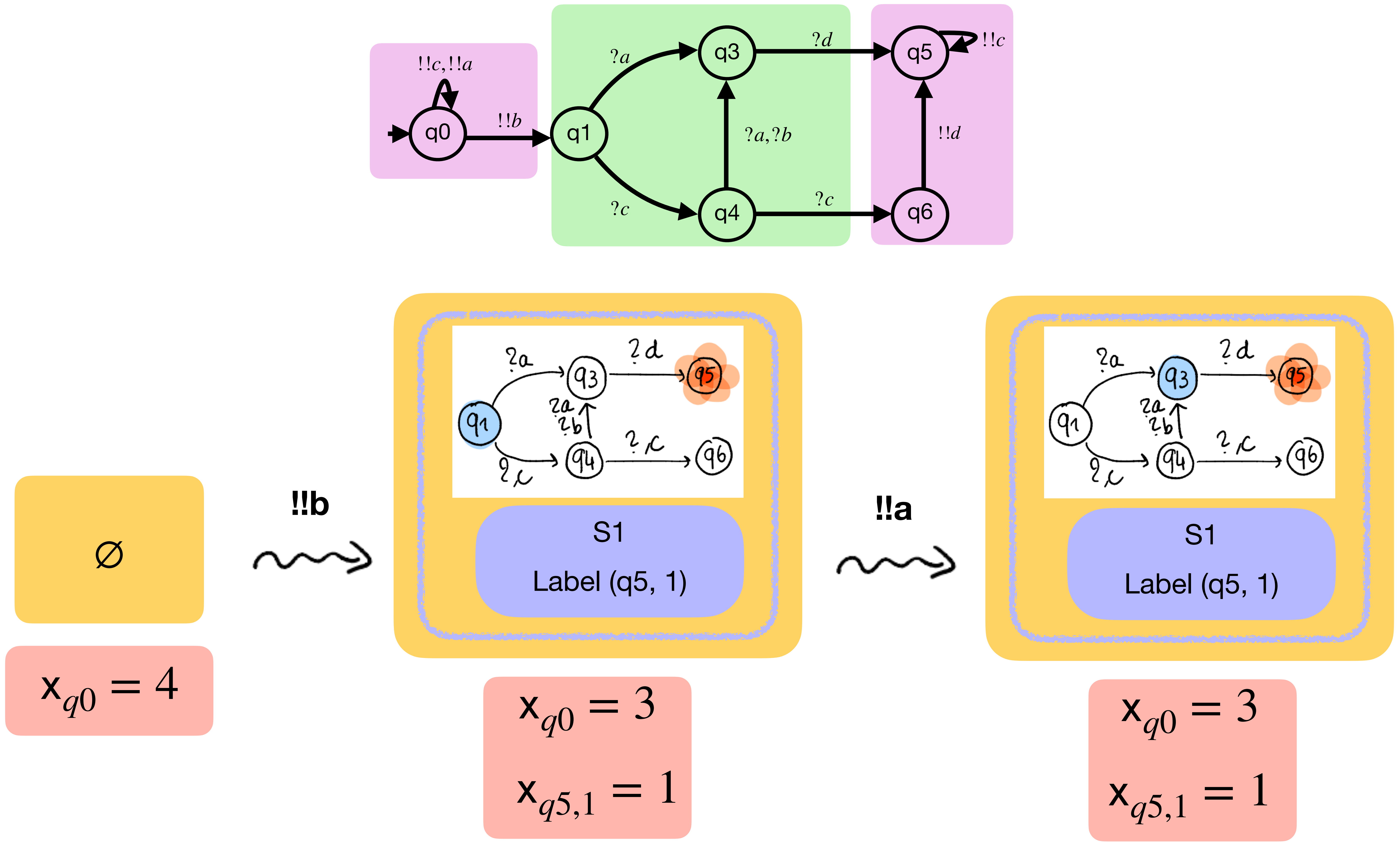
$x_{q0} = 4$

Creation of a Summary

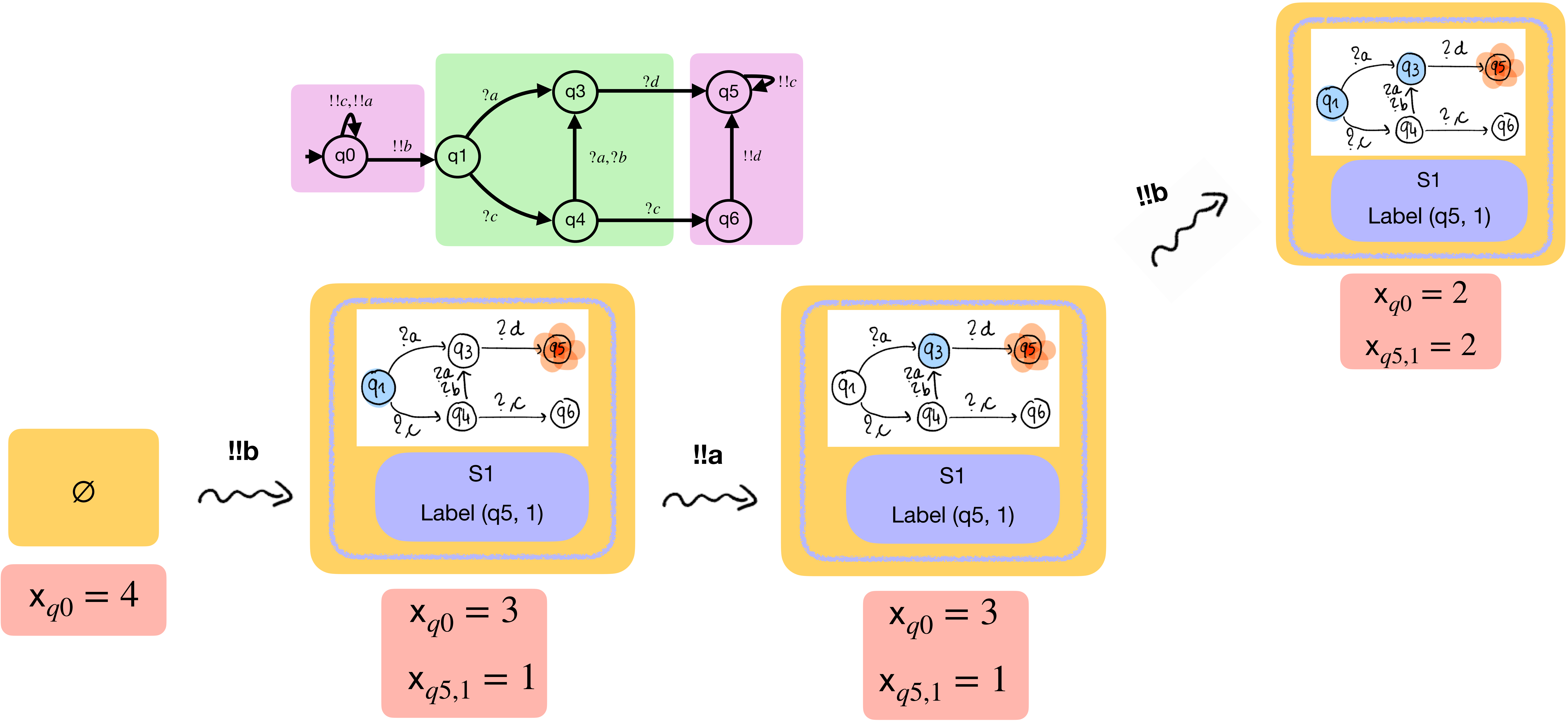




Creation of a Summary

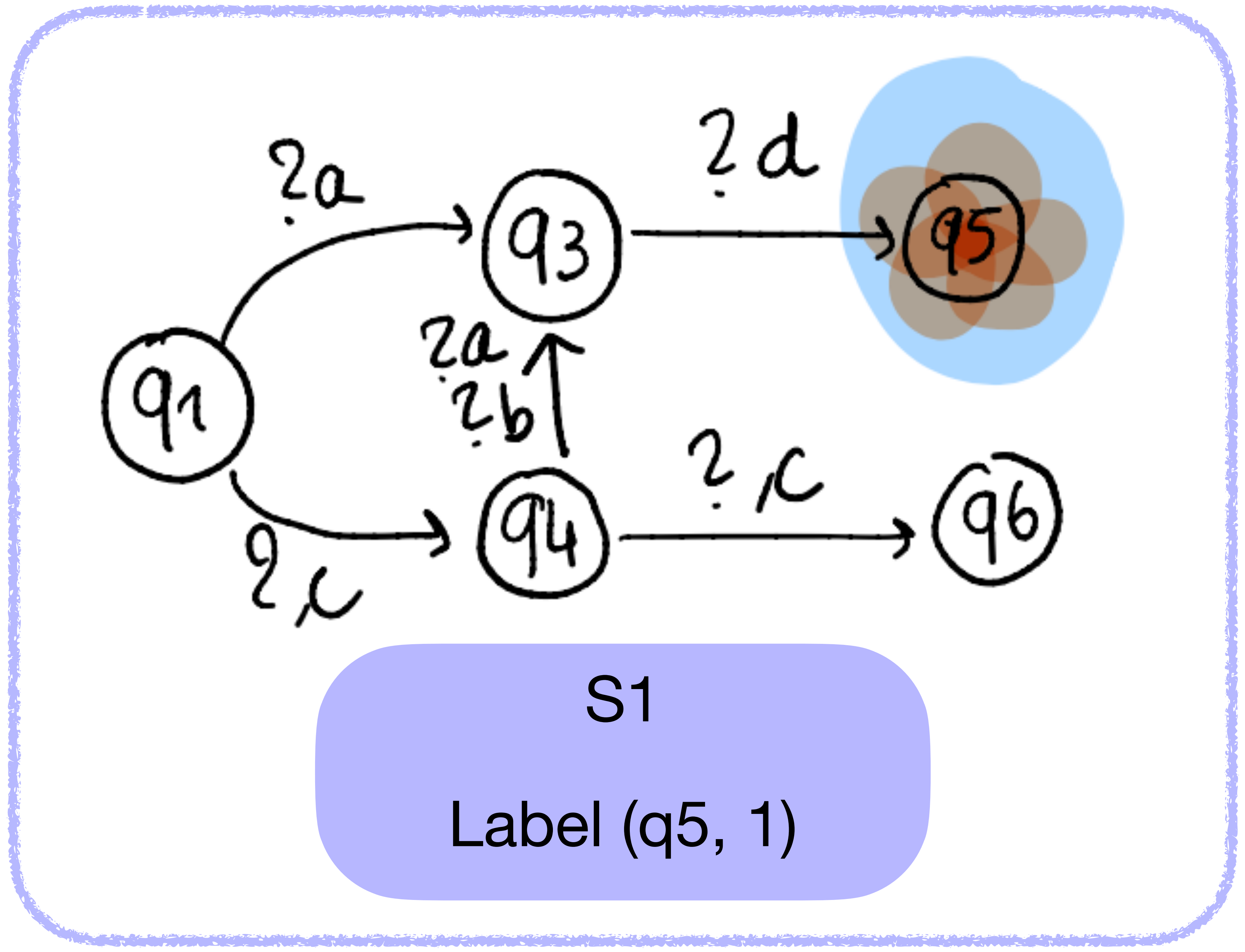


Creation of a Summary



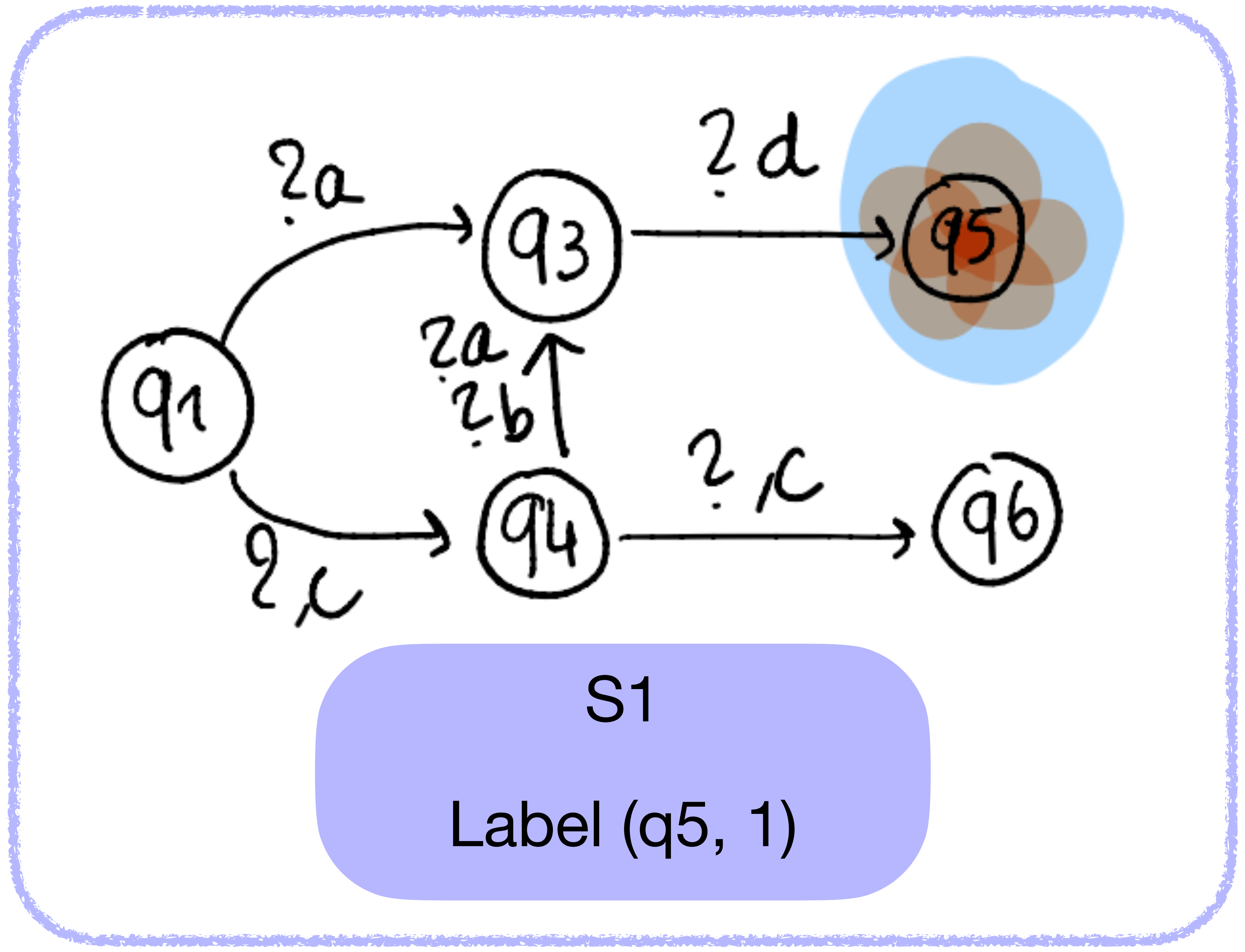
Creation of a Summary

Empty a Summary?



$x_{q5,1} = 4$

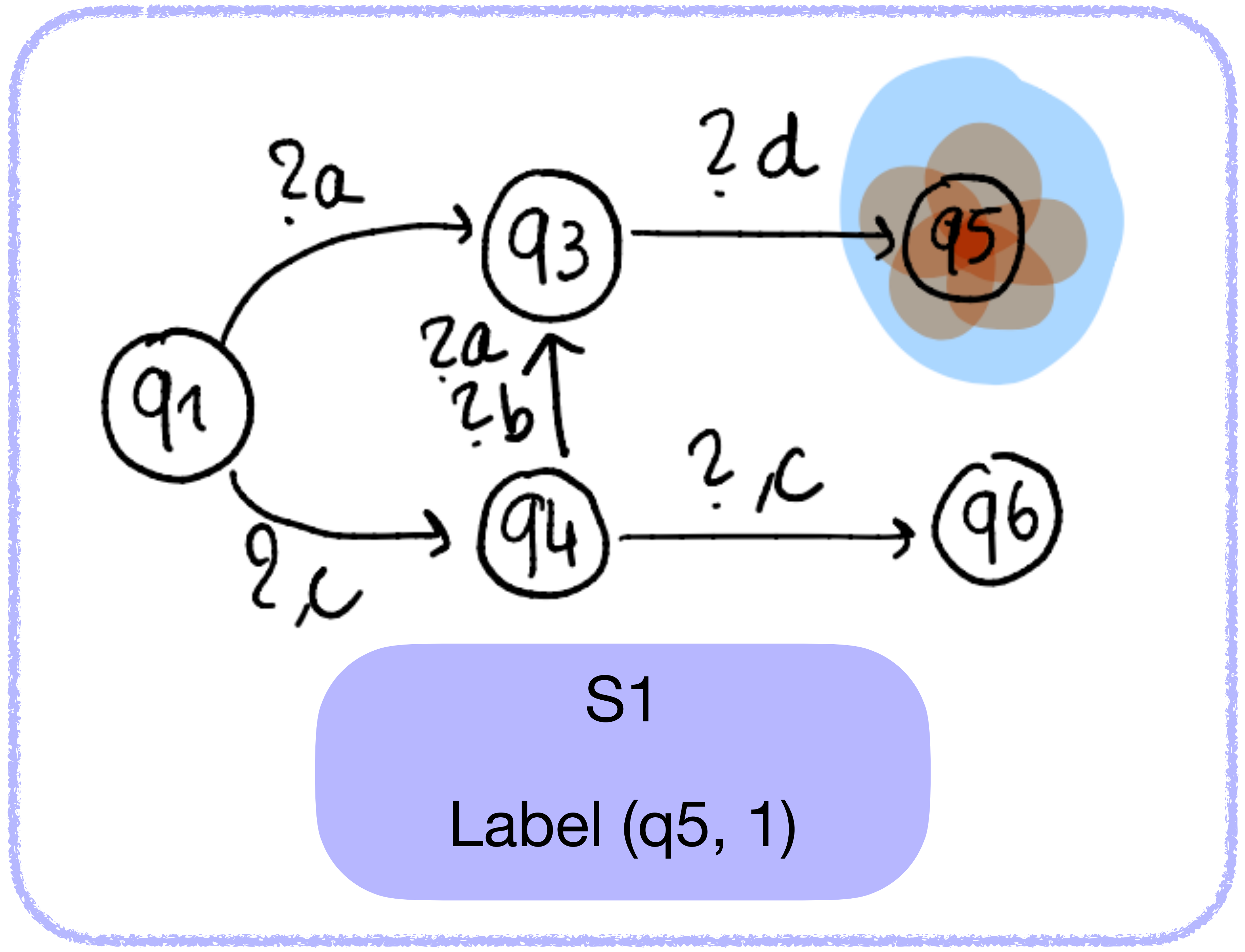
Empty a Summary?



$x_{q5,1} = 4$

Everyone has arrived on q5, what should we do?

Empty a Summary?

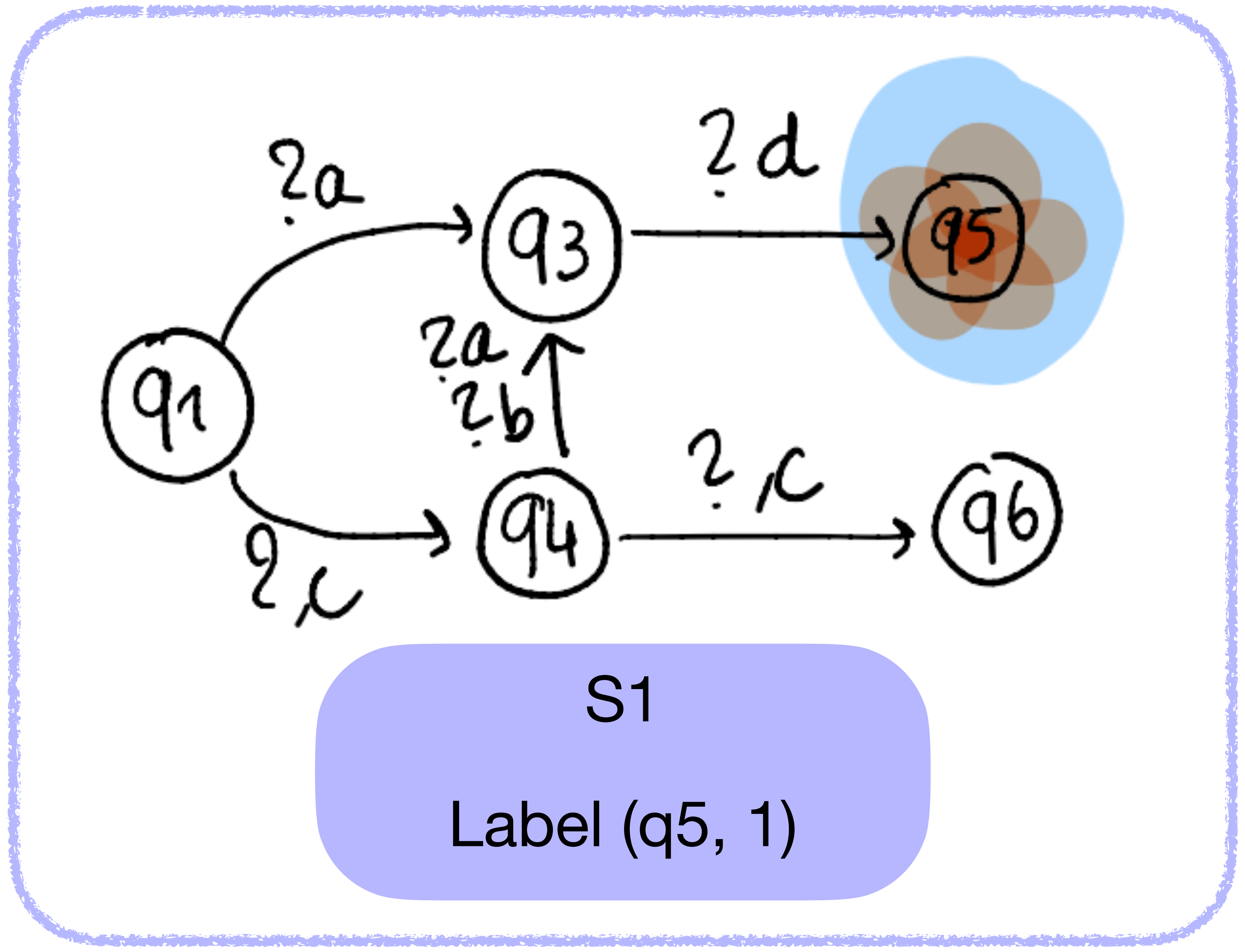


$x_{q5,1} = 4$

Everyone has arrived on q5, what should we do?

Forget about the summary

Empty a Summary?



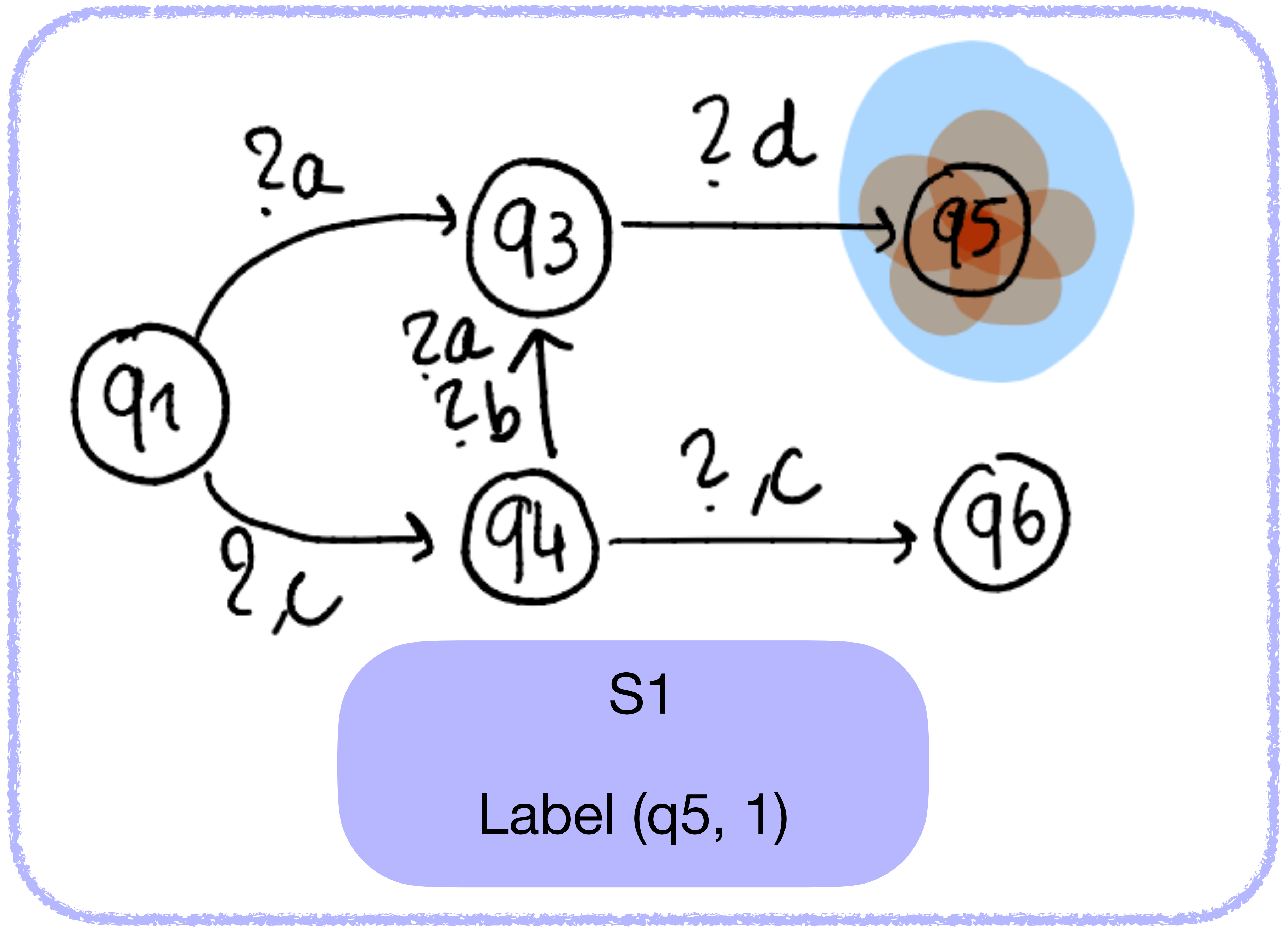
$x_{q5,1} = 4$

Everyone has arrived on q5, what should we do?

Forget about the summary

Transfer the counter $x_{q5,1}$ to x_{q5}

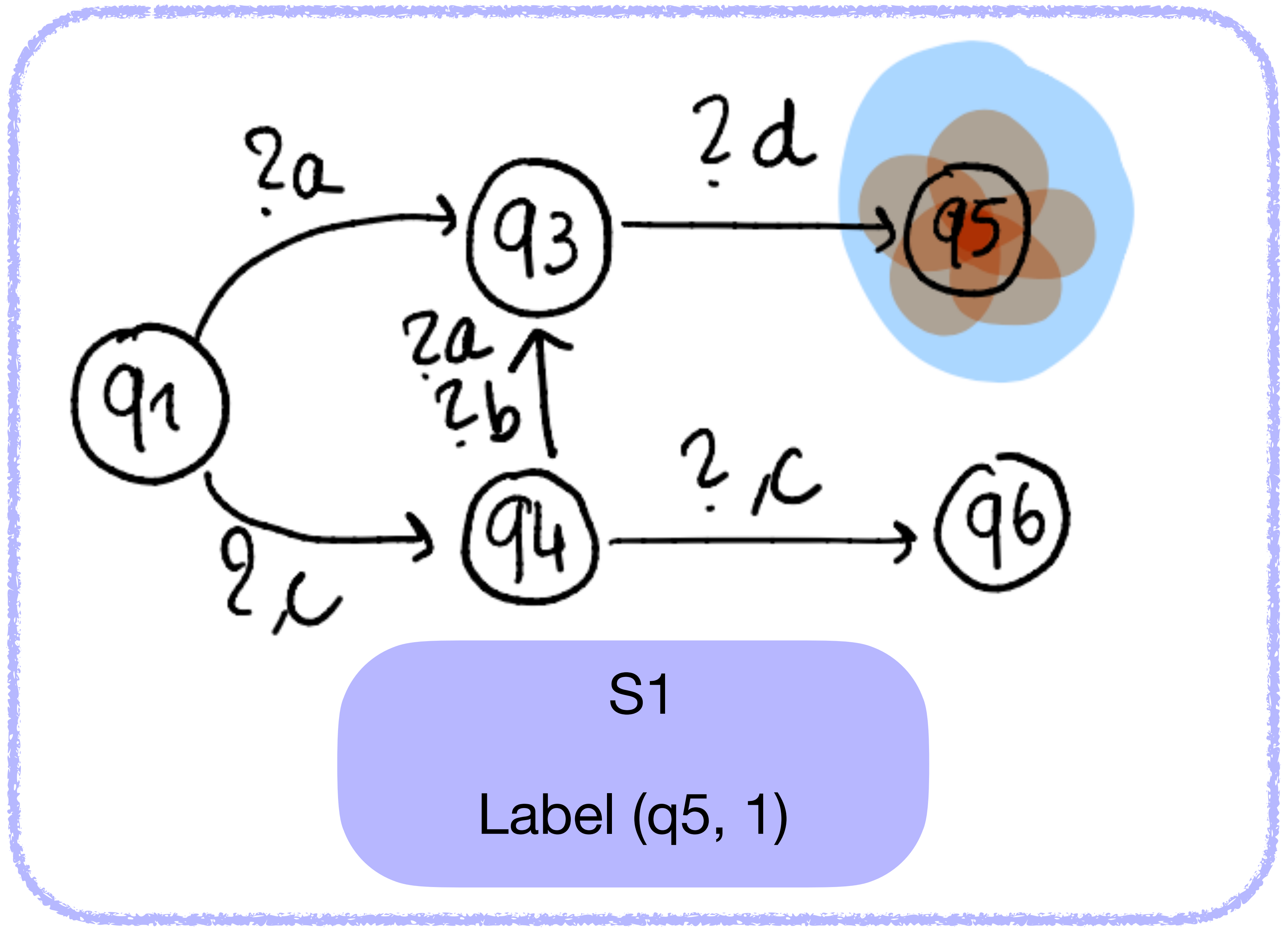
Empty a Summary?



$x_{q5,1} = 4$

$x_{q5} = 0$

Empty a Summary?



$x_{q5,1} = 4$

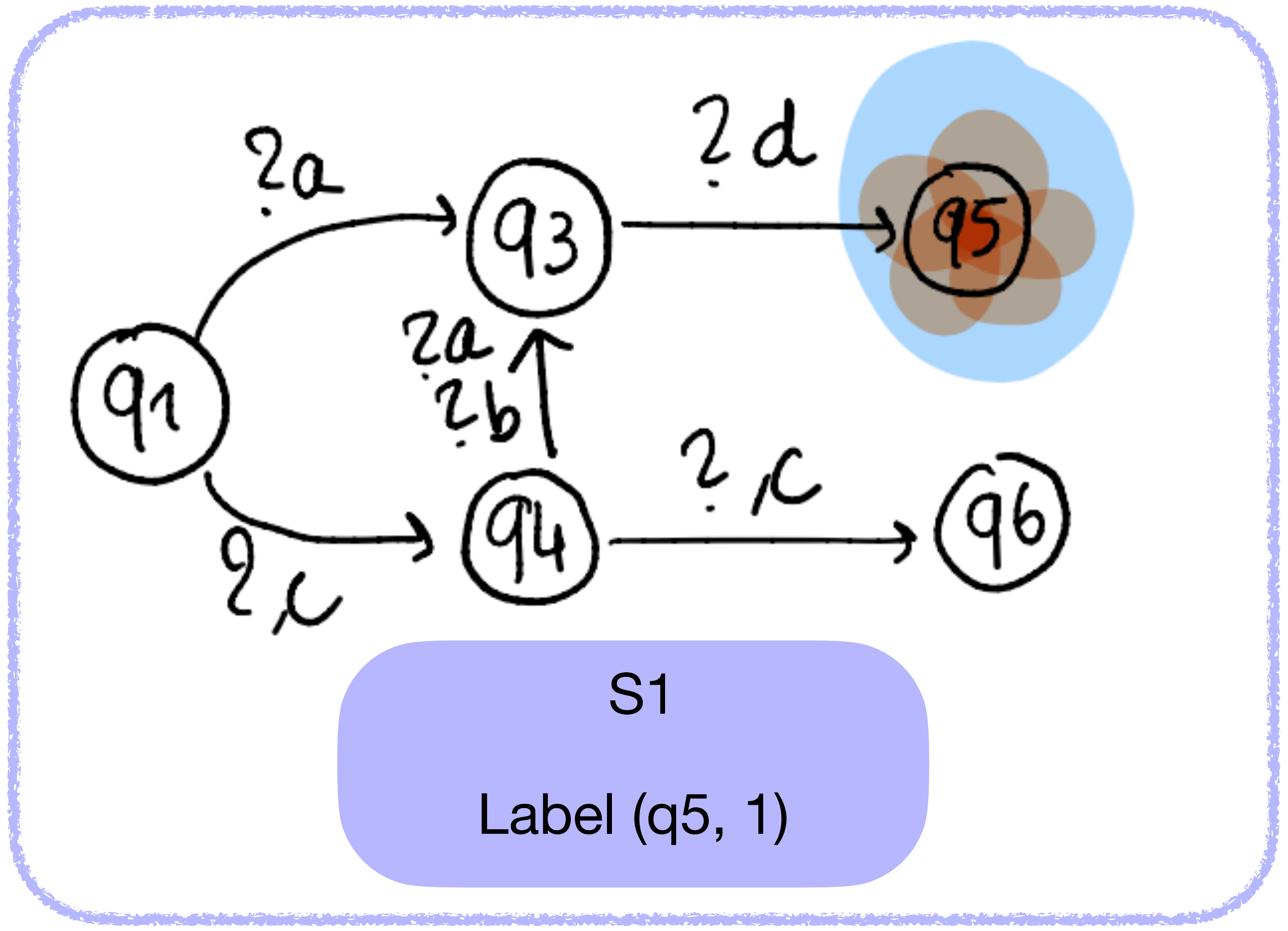
$x_{q5} = 0$

$\rightsquigarrow$

$x_{q5,1} = 0$

$x_{q5} = 4$

Empty a Summary?



$x_{q5,1} = 4$

$x_{q5} = 0$

⇝

$x_{q5,1} = 0$

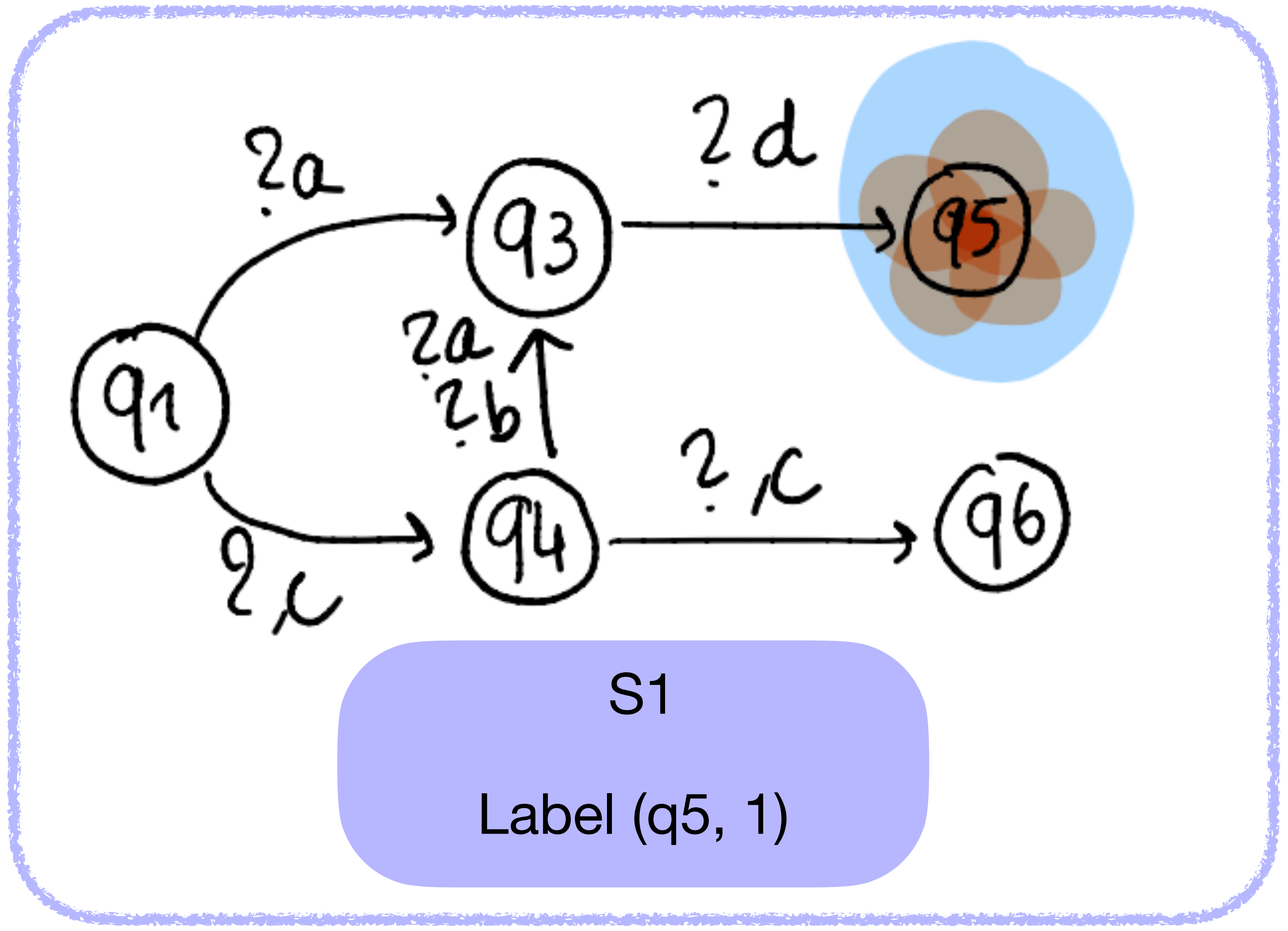
$x_{q5} = 4$

⇝

$x_{q5,1} = 1$

$x_{q5} = 3$

Empty a Summary?



$x_{q5,1} = 4$

$x_{q5} = 0$

⇝

$x_{q5,1} = 0$

$x_{q5} = 4$

⇝

$x_{q5,1} = 1$

$x_{q5} = 3$

Not a problem!

We let a process asleep on q5 until we re use label (q5,1) and re transfer the counter

Conclusion

- Reachability for Wait-Only protocols is decidable but Ackermann-hard
- Model Checking W-O protocols against LTL specification is EXPSPACE-complete (cf. [Habermehl'97])
- Single-Wait-Only protocols

Thank you!