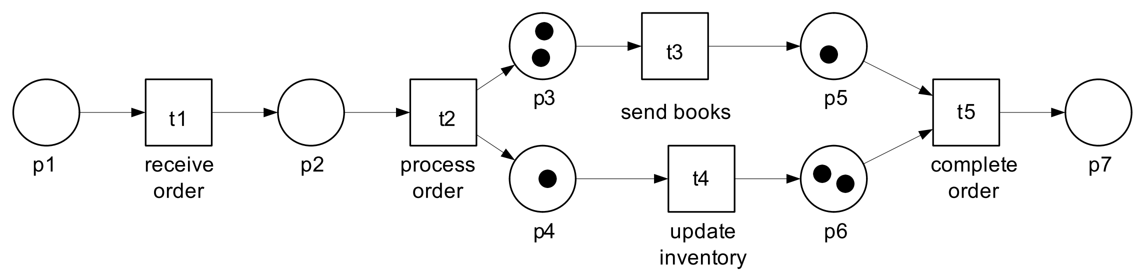
Business Processes Modelling

MPB (6 cfu, 295AA)

Roberto Bruni
http://www.di.unipi.it/~bruni

08 - From automata to nets

Object



Overview of the basic concepts of Petri nets

Free Choice Nets (book, optional reading)
https://www7.in.tum.de/~esparza/bookfc.html

Why Petri nets?

Business process analysis:
validation: testing correctness
verification: proving correctness
performance: planning and optimization

Use of Petri nets (or alike)
visual + formal
tool supported

Petri net theory can be studied at several level of details

We study some basic aspects, relevant to the analysis of business processes

Petri nets have a faithful and convenient graphical representation, that we introduce and motivate next

Approaching Petri nets

Are you familiar with
automata / transition systems?
They are fine for sequential
protocols / systems
but do not capture concurrent
behaviour directly

A Petri net is a mathematical model
of a parallel and concurrent system

in the same way that a finite
automaton is a mathematical model of
a sequential system

Applications
Automata are widely used, e.g., in
protocol analysis,
text parsing,
video game character behaviour,
security analysis,
CPU control units,
natural language processing,
speech recognition,
mechanical devices
(like elevators, vending machines,
traffic lights)
and many more …

Finite automata examples

How to define an automaton

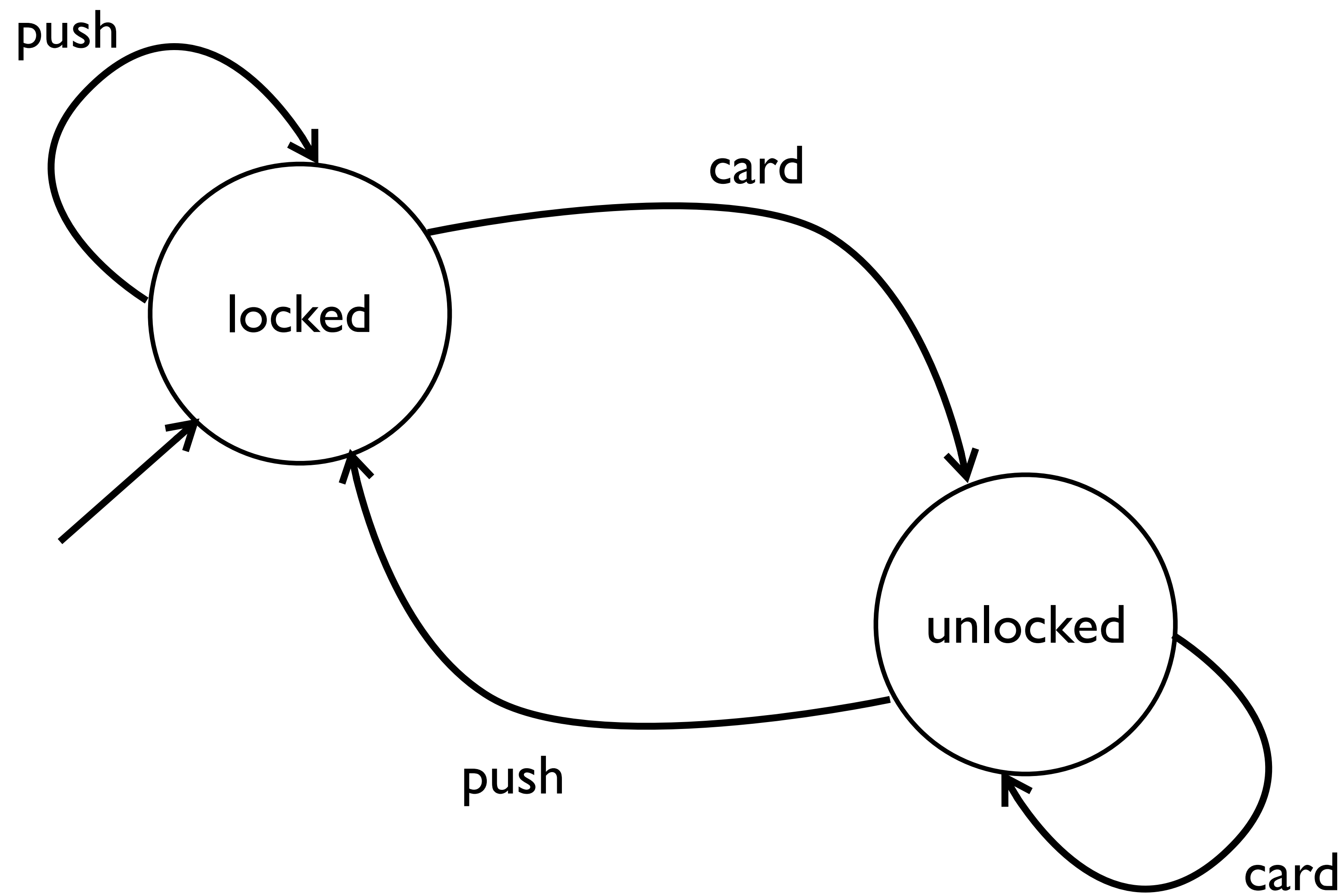
1. Identify the admissible **states** of the system
(Optional: mark some states as error states)

2. **Add transitions**
to move from one state to another
(no transition to recover from error states)

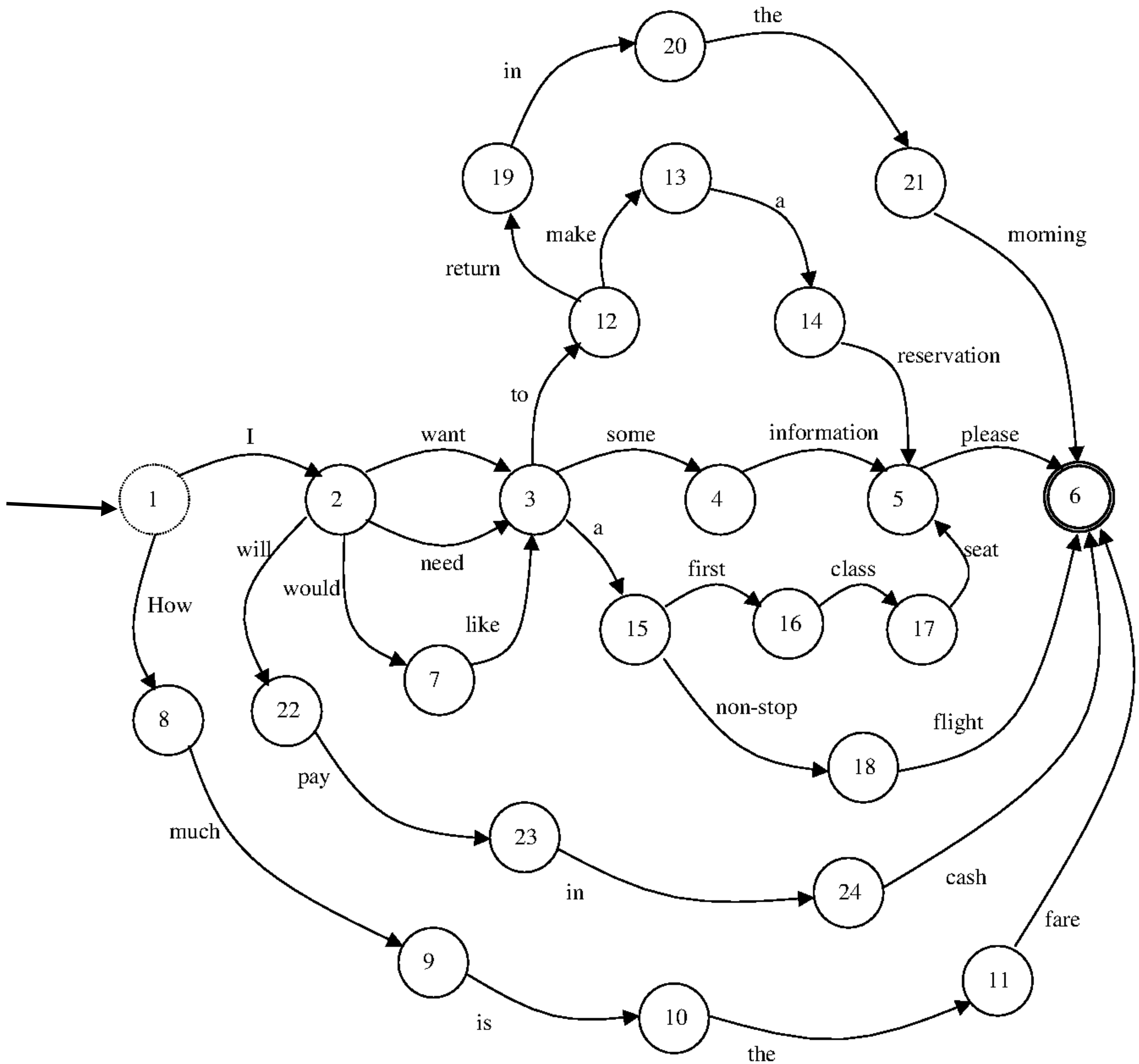
3. Set the **initial state**

4. *(Optional: mark some states as **final states**)*

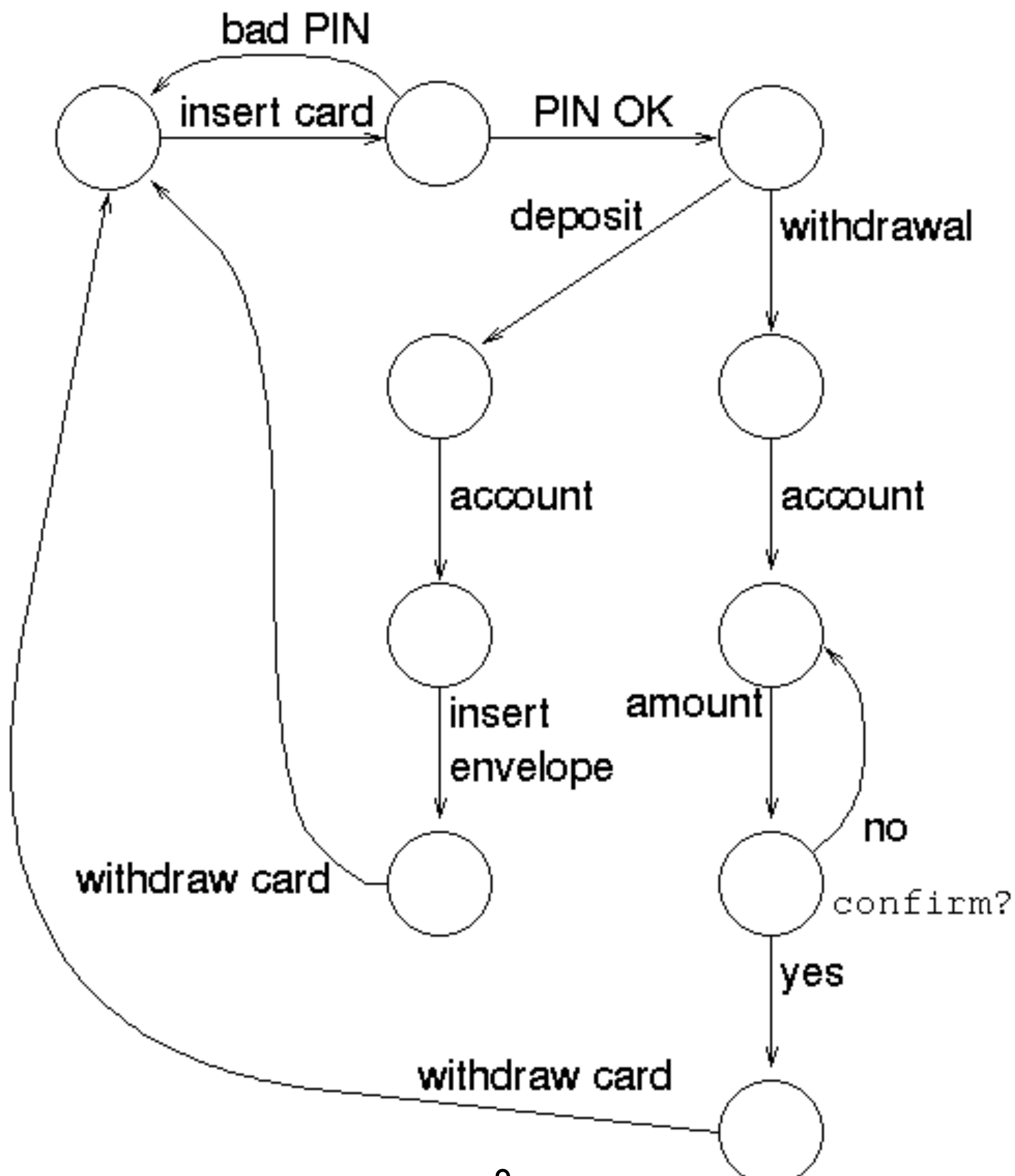
Example: Turnstile



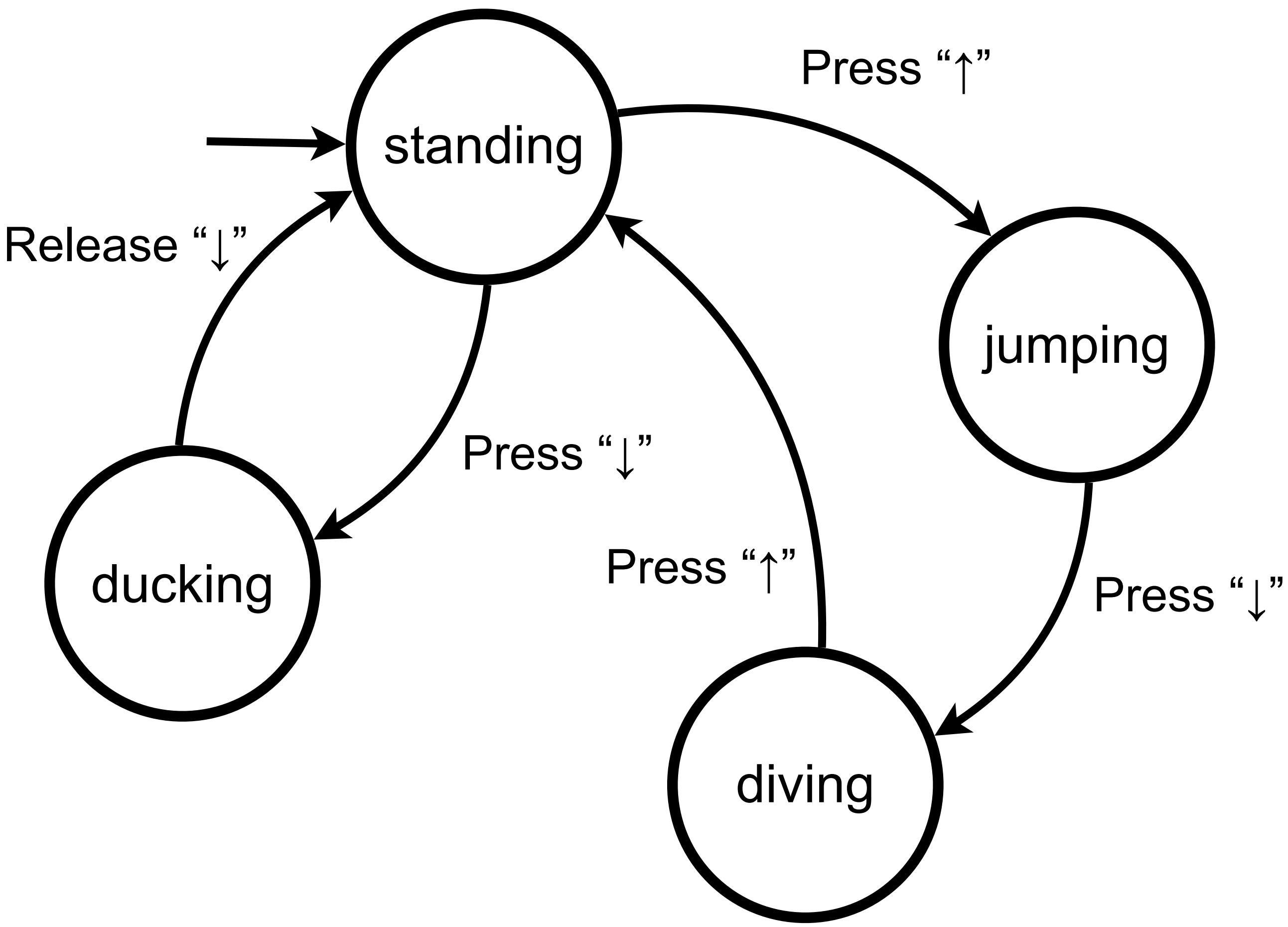
Example: Language Processing

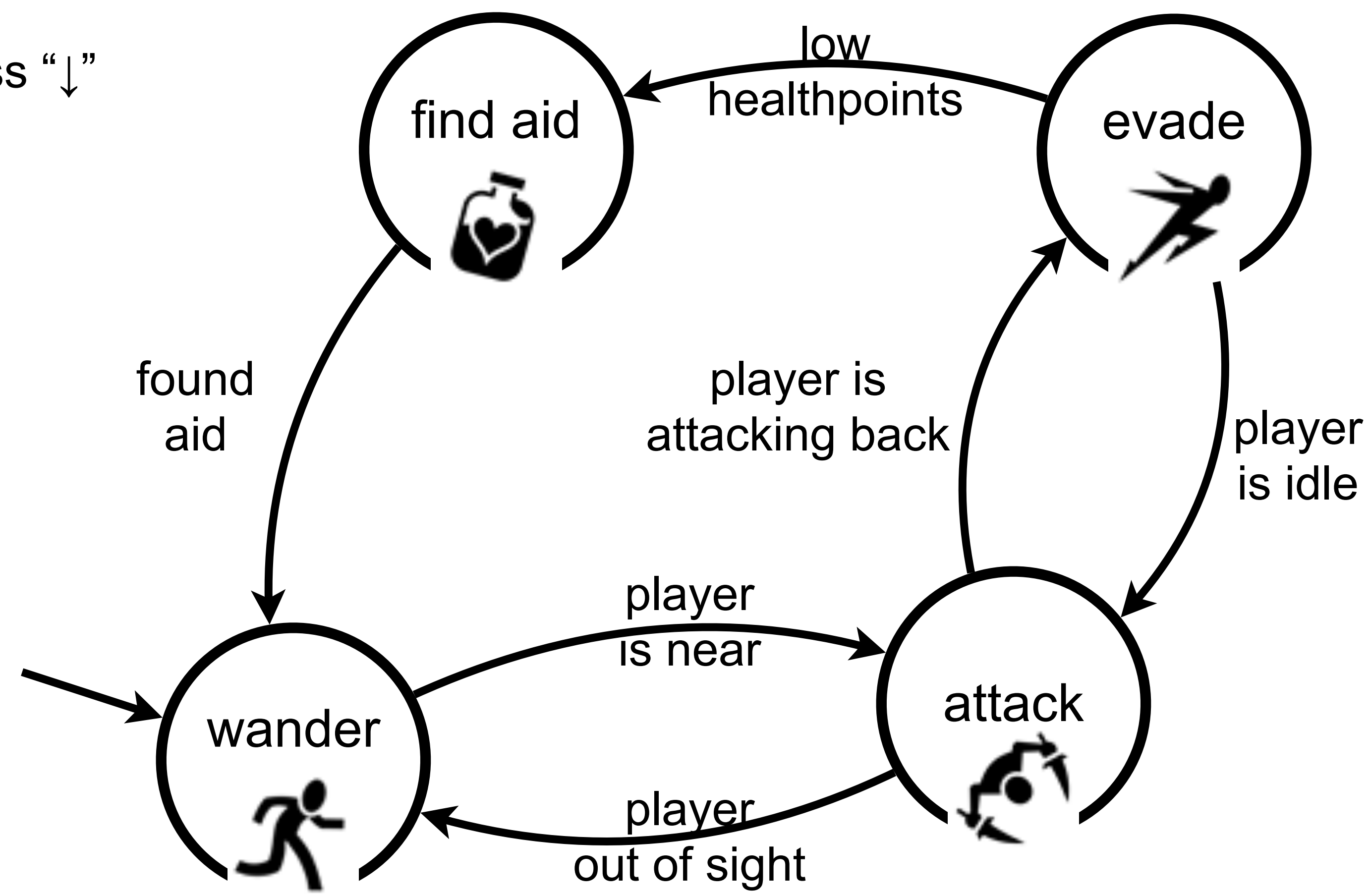


Example: ATM


bad PIN
insert card
PIN OK
deposit
withdrawal
account
account
insert
envelope
amount
no
confirm?
withdraw card
yes
withdraw card

Other examples





Finite state automata, formally

So far, which category for this course?

Subjects are divided in two categories:

1) *too difficult matters, that CANNOT be studied*
2) *easy matters, that DO NOT NEED to be studied*

\- back of a t-shirt

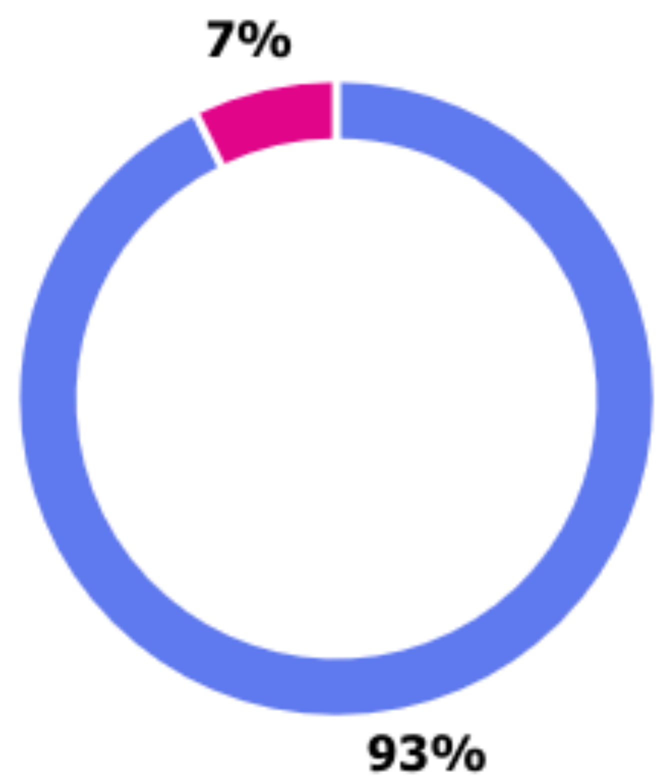
Let us take a U-turn!

Preliminaries

8. Are you familiar with set notation?

Più dettagli

- Yes 51
- No 4



Set notation

$\varnothing$	$A \cap B$	$A \cup B$	$A \setminus B$ $A - B$	$\overline{A}$
$a \in A$	$A = B$	$A \subseteq B$	$A \subset B$	$A \times B$
$a \notin A$	$A \neq B$	$A \nsubseteq B$	$\wp(A)$	$A \cap B = \emptyset$

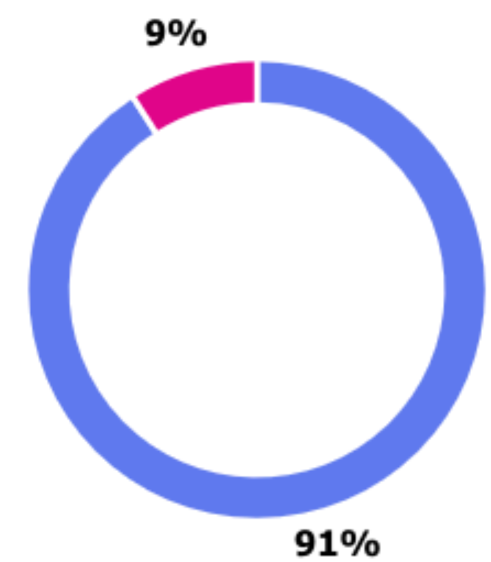
$\mathbb{N}$ $\mathbb{Z}$ $\mathbb{Q}$ $\mathbb{R}$ $\mathbb{B}$

$N \subseteq \mathbb{N}$ $N \in \wp(\mathbb{N})$ $S \subseteq \wp(\mathbb{N})$

9. Are you familiar with functions (f:A->B) and relations?

Più dettagli

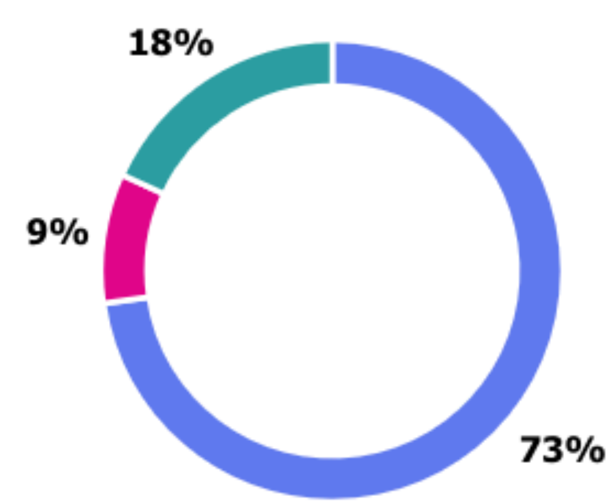
●	Yes	50
●	No	5



10. Do you agree that a subset S of A can be seen as a function from A to the set of Booleans?

Più dettagli

●	Yes	40
●	No	5
●	I do not understand the question	10



Functions, relations

$f : A \to B$

$R \subseteq A \times B$

functions as relations

$$R_f \triangleq \{(a, f(a)) \mid a \in A\}$$

sets as functions
(characteristic function)

$$f_N : \mathbb{N} \to \mathbb{B}$$

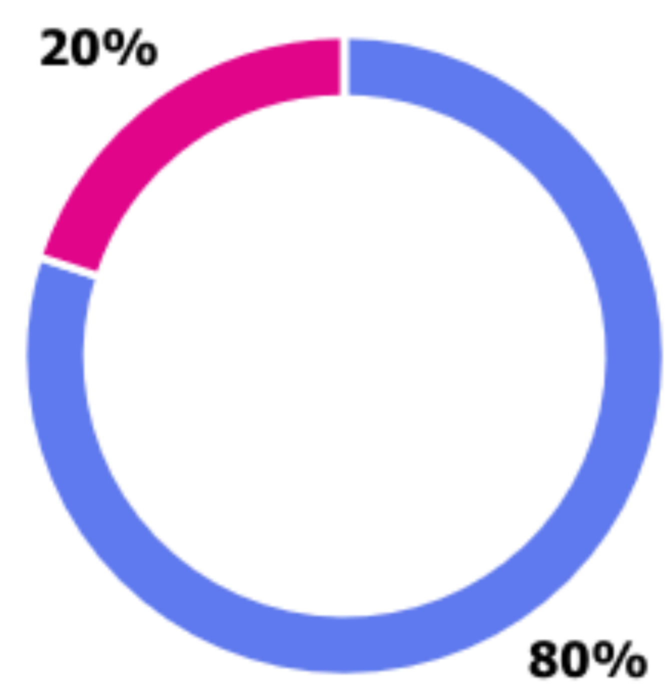
$$f_N(n) \triangleq \begin{cases} 1 & n \in N \\ 0 & \text{otherwise} \end{cases}$$

$$N = \{n \mid f_N(n) = 1\}$$

12. Are you familiar with propositional logic?

Più dettagli

Yes	44
No	11

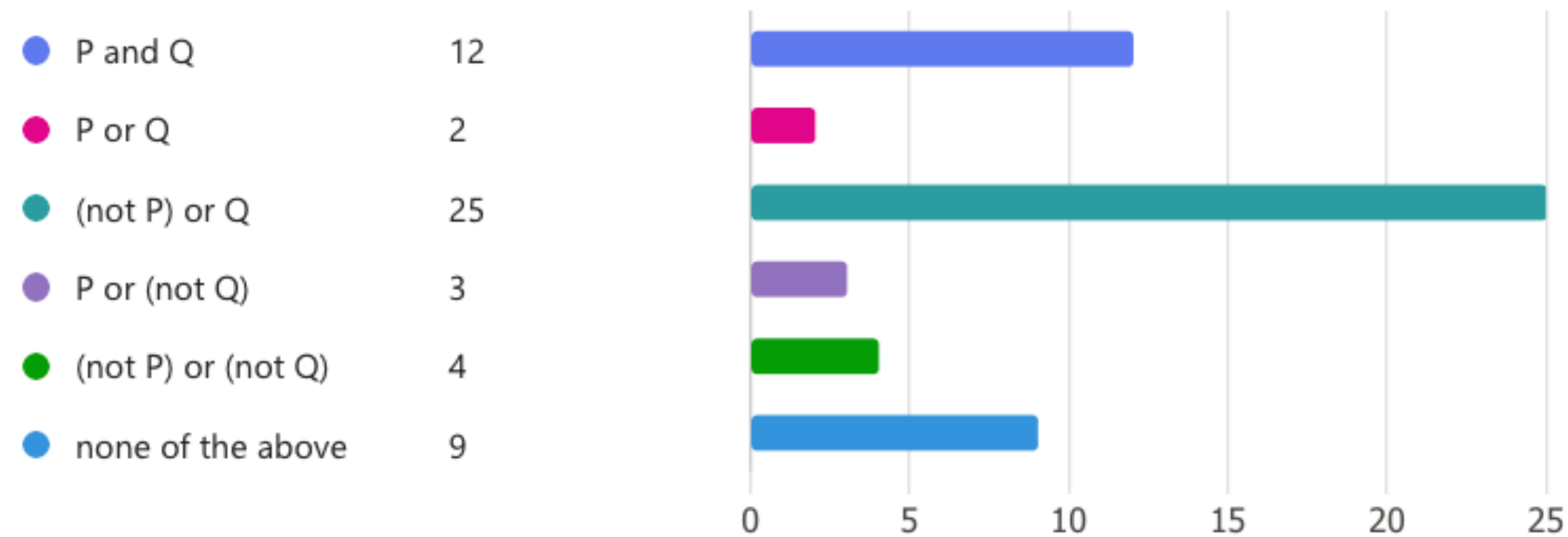


First order logic

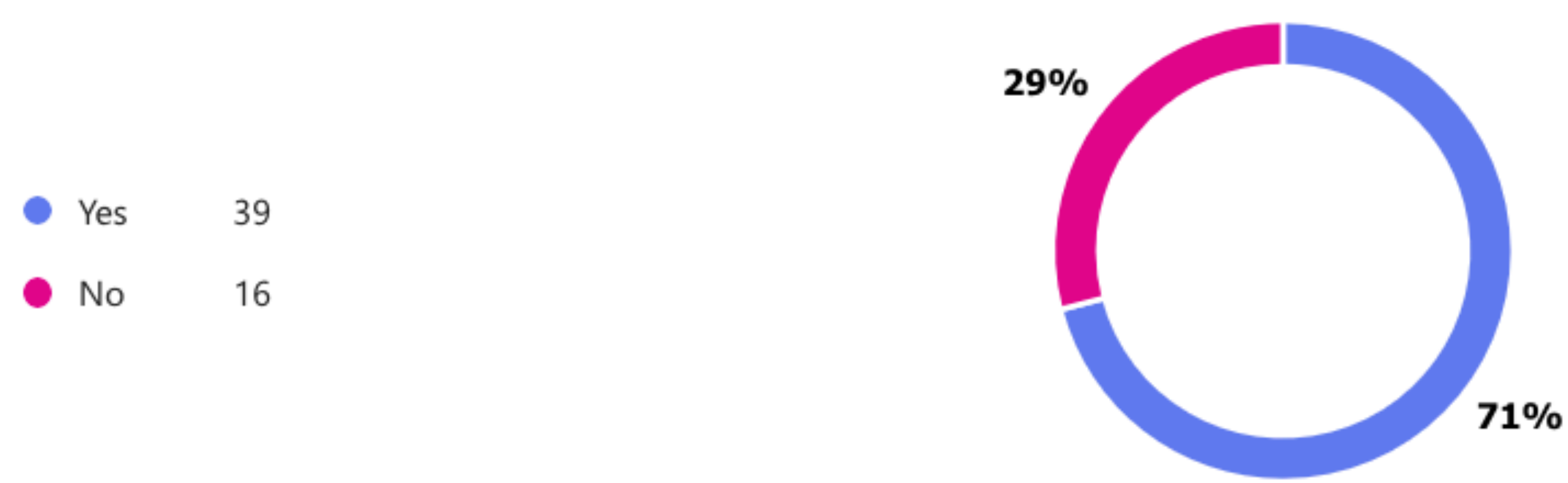
ff false 0 F tt true 1 T $P \wedge Q$ $P \vee Q$ $\neg P$

$\exists x.\ P(x)$ $\forall x.\ P(x)$ $P \Rightarrow Q$ $P \Leftrightarrow Q$

13. Logical implication "P implies Q" (also written "P => Q") is equivalent to: Più dettagli



14. Do you agree that "P implies Q" is logically equivalent to "(not Q) implies (not P)? Più dettagli



First order logic

ff false 0 F

tt true 1 T

$P \wedge Q$ $P \vee Q$ $\neg P$

$\exists x.\ P(x)$ $\forall x.\ P(x)$ $P \Rightarrow Q$ $P \Leftrightarrow Q$

meaning of implication!

$$P \Rightarrow Q$$

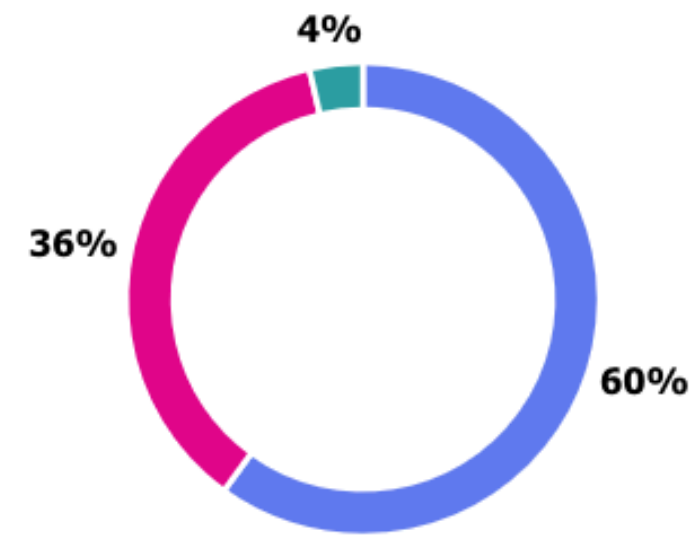
$$Q \vee \neg P$$

$$\neg Q \Rightarrow \neg P$$

15. Do you remember De Morgan's law about negation, conjunction and disjunction?

Più dettagli

Yes	33
No	20
I do not understand the question	2



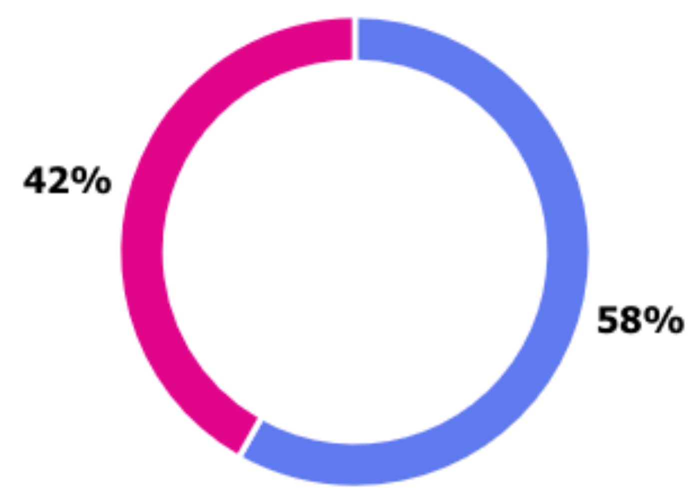
$$\neg(P \land Q) = (\neg P) \lor (\neg Q)$$
$$\neg(P \lor Q) = (\neg P) \land (\neg Q)$$

16. Do you know what are the universal and existential quantifiers in predicate logic?

Più dettagli

$$\exists x\,.\,P$$
$$\forall x\,.\,P$$

Yes	32
No	23



$$\neg(\exists x\,.\,P) = \forall x\,.\,\neg P$$
$$\neg(\forall x\,.\,P) = \exists x\,.\,\neg P$$

First order logic

ff false 0 F tt true 1 T $P \wedge Q$ $P \vee Q$ $\neg P$

$\exists x.\ P(x)$ $\forall x.\ P(x)$ $P \Rightarrow Q$ $P \Leftrightarrow Q$

meaning of implication!

$$P \Rightarrow Q$$

$$Q \vee \neg P$$

$$\neg Q \Rightarrow \neg P$$

order of quantifiers matters!

$$\forall n \in \mathbb{N}.\ \exists m \in \mathbb{N}.\ n < m$$

$$\exists m \in \mathbb{N}.\ \forall n \in \mathbb{N}.\ n < m$$

Sequences

Given a set A of symbols, we denote by A^n
the set of finite sequences of exactly n elements in A, i.e.:

$$A^n = \{a_1\ a_2\ \cdots\ a_n \mid a_1, a_2, \ldots, a_n \in A\}$$

i.e., $A^n = \underbrace{A \times \cdots \times A}_{n}$

We denote the empty sequence by ϵ

For example, when $A = \{0,1\}$

$A^0 = \{\epsilon\}$ $\qquad A^1 = A$ $\qquad A^2 = \{00,\ 01,\ 10,\ 11\}$

Kleene-star notation A*

Given a set A of symbols, we denote by $A^\star$
the set of all finite sequences of elements in A, i.e.:

$$A^\star = \{a_1\ a_2\ \cdots\ a_n \mid n \geq 0\ \wedge\ a_1, a_2, \ldots, a_n \in A\}$$

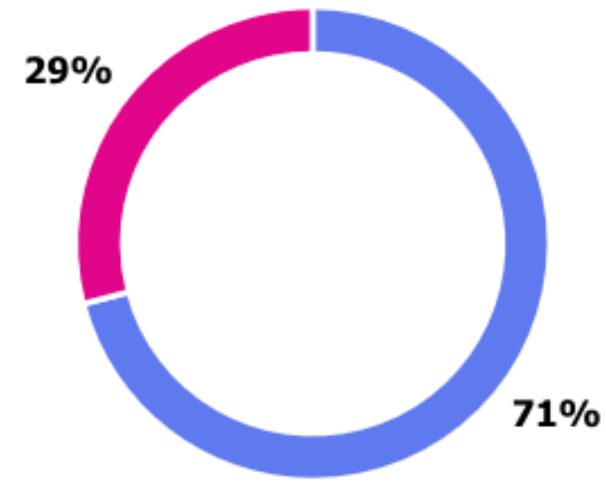
Clearly, we have $A^\star = \bigcup_{n \geq 0} A^n$

For example, when $A = \{0,1\}$
$A^\star = \{\epsilon,\ 0,\ 1,\ 00,\ 01,\ 10,\ 11,\ 000,\ \ldots\}$

17. Do you know what is an inductive definition?

Più dettagli

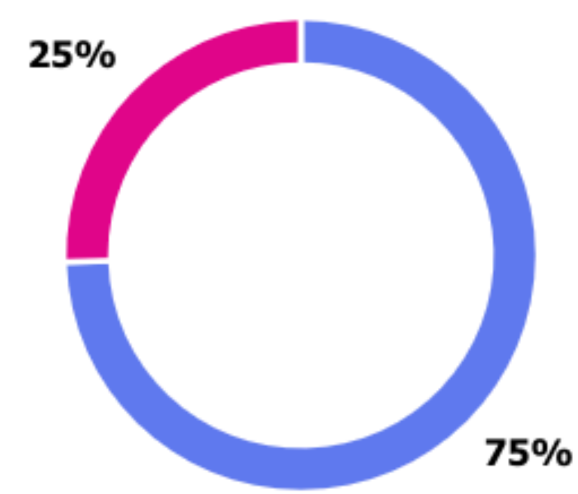
- Yes 39
- No 16



18. Do you know what is a recursive definition?

Più dettagli

- Yes 41
- No 14



Inductive definitions

A natural number is either:

- 0
- or the successor $n + 1$ of a natural number n

A sequence over the alphabet A is either:

- the empty sequence ϵ
- or the juxtaposition wa of a sequence w with an element a of A

Inductively defined functions

Let us define the exponential function k^n

base case: for any $k > 0$ we set
$exp(\ k, 0\ ) = 1$

inductive case: for any $k > 0,\ n \geq 0$ we set
$exp(\ k, n + 1\ ) = exp(\ k, n\ ) \times k$

Inductively defined functions

Let us define the exponential function k^n

base case: for any $k > 0$ we set
$exp(\ k,\ 0\ ) = 1$

inductive case: for any $k > 0,\ n \geq 0$ we set
$exp(\ k,\ n+1\ ) = exp(\ k,\ n\ ) \times k$

Recursive definition

Inductively defined functions

Let us define the exponential function k^n

base case: for any $k > 0$ we set
$exp(\ k, \ 0\ ) = 1$

inductive case: for any $k > 0, \ n \geq 0$ we set
$exp(\ k, \ \ n+1\ ) = exp(\ k, \ \ n\ ) \times k$

More complex case **Simpler case**

Recursive definitions

$$\binom{n}{k} \triangleq \frac{n!}{k!\,(n-k)!}$$

$$\binom{n}{0} \triangleq 1$$

$$\binom{n+1}{k+1} \triangleq \frac{(n+1)\binom{n}{k}}{k+1}$$

$$f(n) \triangleq \begin{cases} 1 & \text{if } n \leq 1 \\ f(n/2) & \text{if } n > 1 \wedge n\%2 = 0 \\ f(3n+1) & \text{otherwise} \end{cases}$$

$$f(12) = f(6) = f(3) = f(10) = f(5) = f(16) = f(8) = f(4) = f(2) = f(1) = 1$$

Inductive definitions

$$
\begin{aligned}
0! &\triangleq 1 \\
(n+1)! &\triangleq n! \cdot (n+1)
\end{aligned}
$$

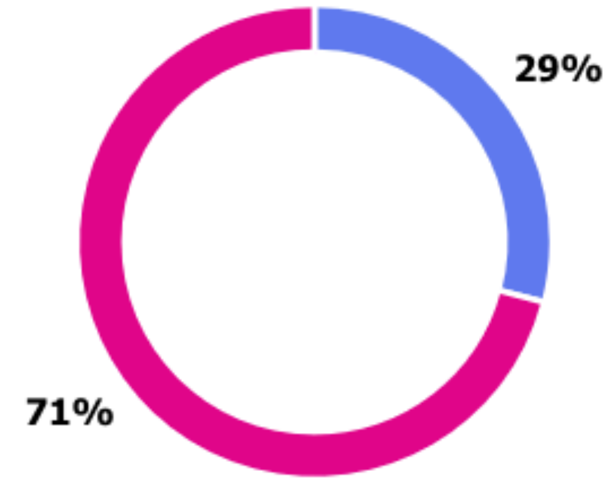
$$
\begin{aligned}
A^0 &\triangleq \{\epsilon\} \\
A^{(n+1)} &\triangleq A \times A^n
\end{aligned}
$$

$$
\begin{aligned}
|\epsilon| &\triangleq 0 \\
|w\ a| &\triangleq |w| + 1
\end{aligned}
$$

21. Do you know what is a Finite State Automata?

Più dettagli

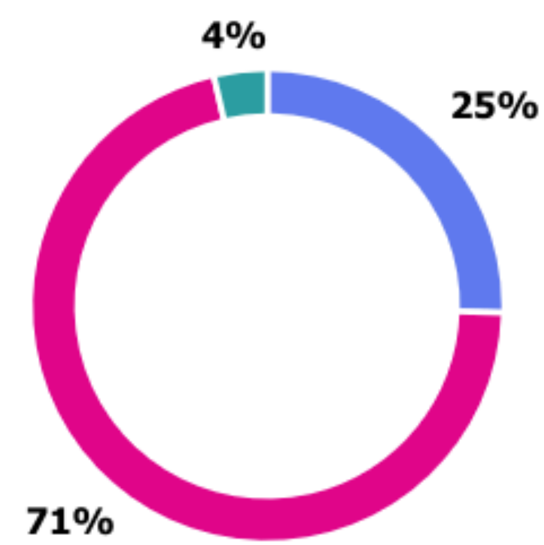
●	Yes	16
●	No	39



22. Do you know what is the language recognized by an automata?

Più dettagli

●	Yes	14
●	No	39
●	I do not understand the question	2



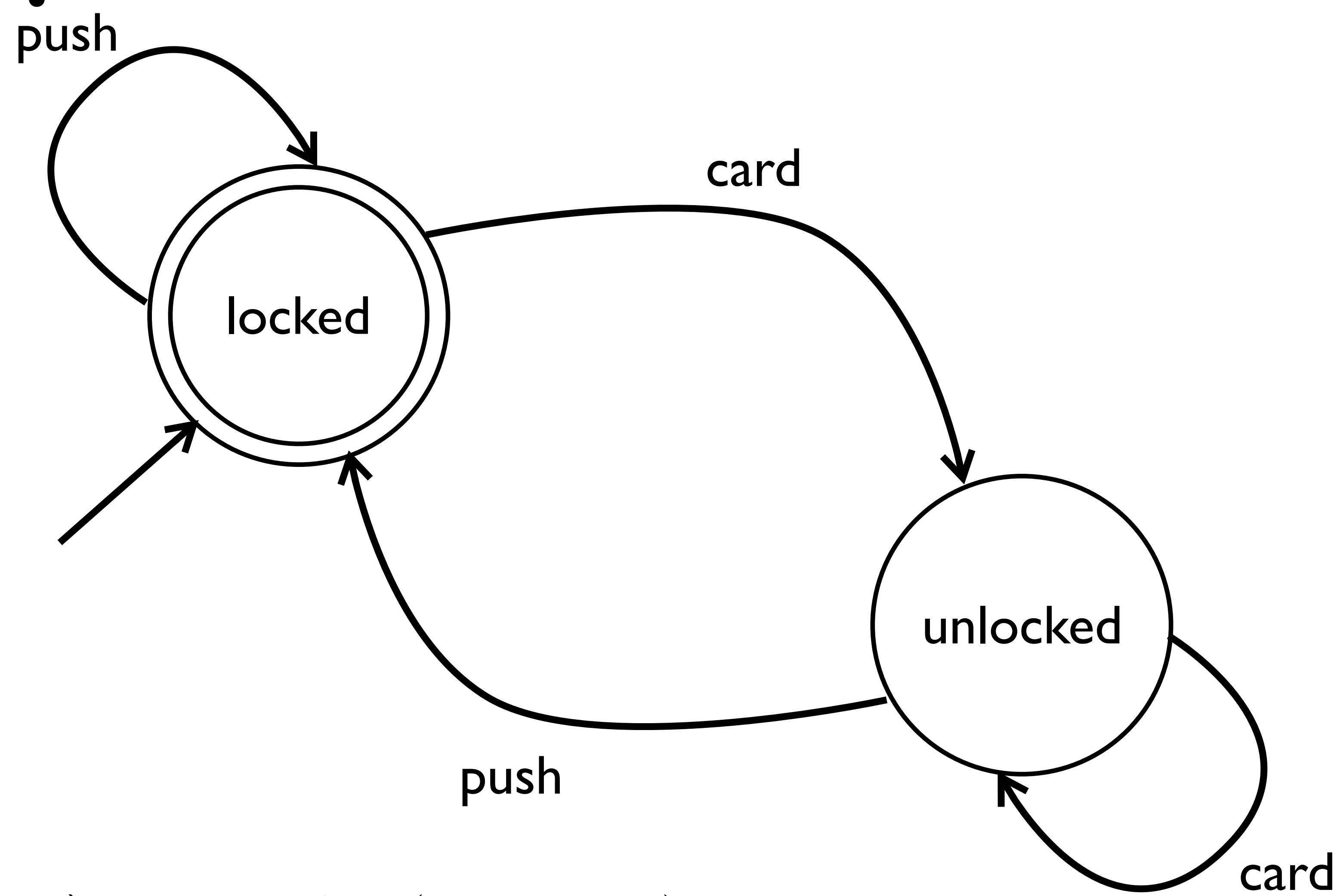
Don't be scared!

DFA

A **Deterministic Finite Automaton** (**DFA**) is a tuple $A = (Q, \Sigma, \delta, q_0, F)$, where

- Q is a finite set of states;
- Σ is a finite set of input symbols;
- $\delta : Q \times \Sigma \to Q$ is the transition function;
- $q_0 \in Q$ is the initial state (also called start state);
- $F \subseteq Q$ is the set of final states (also accepting states)

Example: Turnstile



A **Deterministic Finite Automaton** (**DFA**) is a tuple $A = (Q, \Sigma, \delta, q_0, F)$, where

- Q is a finite set of states;
- Σ is a finite set of input symbols;
- $\delta : Q \times \Sigma \to Q$ is the transition function;
- $q_0 \in Q$ is the initial state (also called start state);
- $F \subseteq Q$ is the set of final states (also accepting states)

$$Q = \{\text{locked}, \text{unlocked}\}$$

$$\Sigma = \{\text{push}, \text{card}\}$$

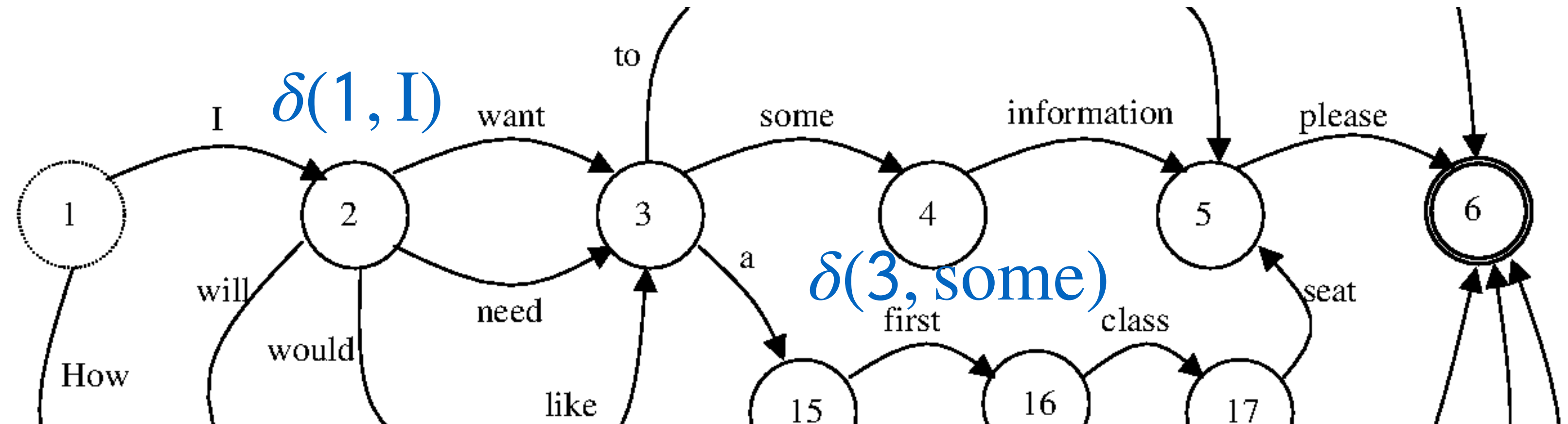
$$\delta(\text{locked}, \text{card}) = \text{unlocked}$$

$$q_0 = \text{locked}$$

$$F = \{\text{locked}\}$$

Transition function

Given the current state q and the input a
the next state is $\delta(q, a)$

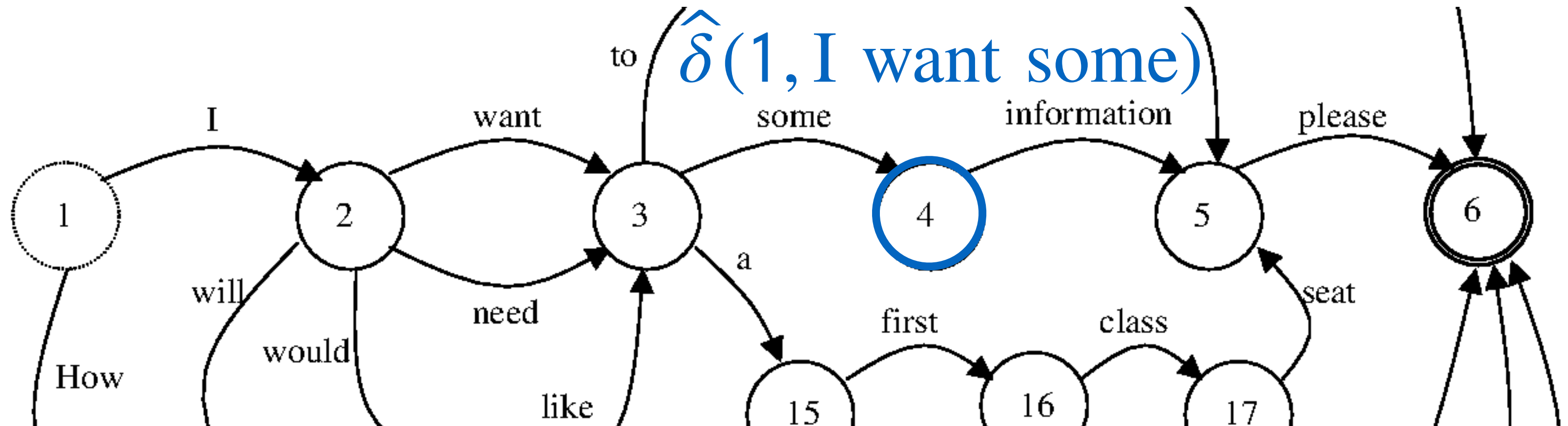


Extended transition function (destination function)

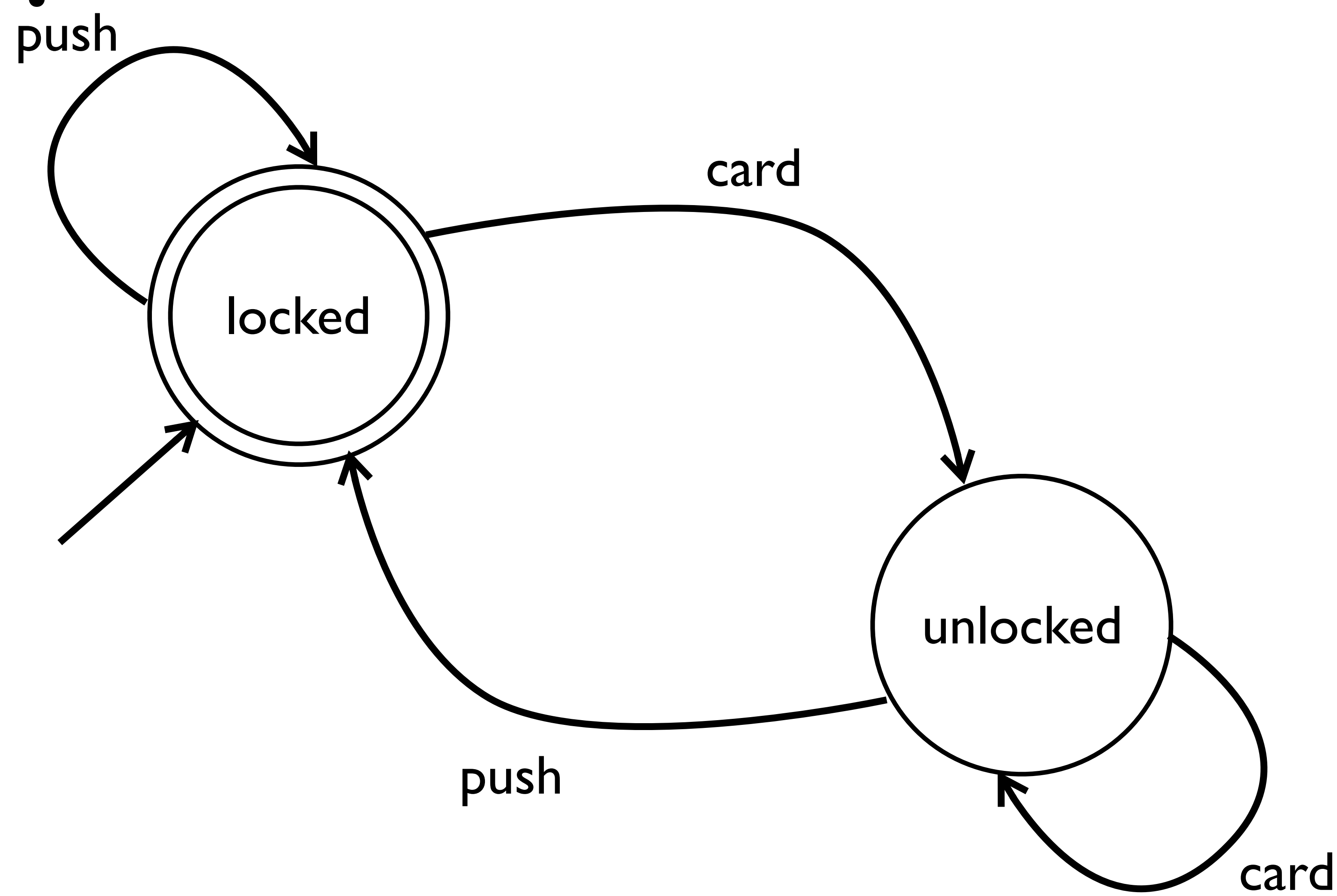
Given the current state q and the input sequence w the state we reach is $\hat{\delta}(q, w)$

$$w = a_1\ a_2\ \cdots\ a_n$$

$$\hat{\delta}(q, w) = \delta(\ \delta(\ \cdots\ \delta(\ \delta(q, a_1)\ , a_2\ )\ , \cdots\ )\ , a_n\ )$$



Example: Turnstile



$$\delta(\text{locked}, \text{card}) = \text{unlocked}$$
$$\delta(\text{locked}, \text{push}) = \text{locked}$$
$$\delta(\text{unlocked}, \text{card}) = \text{unlocked}$$
$$\delta(\text{unlocked}, \text{push}) = \text{locked}$$

$$\widehat{\delta}(\text{locked}, \epsilon) = \text{locked}$$
$$\widehat{\delta}(\text{locked}, \text{card}) = \text{unlocked}$$
$$\widehat{\delta}(\text{locked}, \text{card card}) = \text{unlocked}$$
$$\widehat{\delta}(\text{locked}, \text{card card push}) = \text{locked}$$

Extended transition function (destination function)

Given $A = (Q, \Sigma, \delta, q_0, F)$, we define $\widehat{\delta} : Q \times \Sigma^* \to Q$ by induction:

base case: For any $q \in Q$ we let

$$\widehat{\delta}(q, \epsilon) \triangleq q$$

inductive case: For any $q \in Q, a \in \Sigma, w \in \Sigma^*$ we let

$$\widehat{\delta}(q, wa) \triangleq \delta(\ \widehat{\delta}(q, w)\ ,\ a\ )$$

($\widehat{\delta}(q, w)$ returns the state reached from q by observing w)

Extended transition function (destination function)

Given $A = (Q, \Sigma, \delta, q_0, F)$, we define $\widehat{\delta} : Q \times \Sigma^* \to Q$ by induction:

base case: For any $q \in Q$ we let

$$\widehat{\delta}(q, \epsilon) \triangleq q$$

inductive case: For any $q \in Q, a \in \Sigma, w \in \Sigma^*$ we let

$$\widehat{\delta}(q, wa) \triangleq \delta(\boxed{\widehat{\delta}(q, w)}\,,\ a\ )$$

Recursive definition

($\widehat{\delta}(q, w)$ returns the state reached from q by observing w)

Extended transition function (destination function)

Given $A = (Q, \Sigma, \delta, q_0, F)$, we define $\widehat{\delta} : Q \times \Sigma^* \to Q$ by induction:

base case: For any $q \in Q$ we let

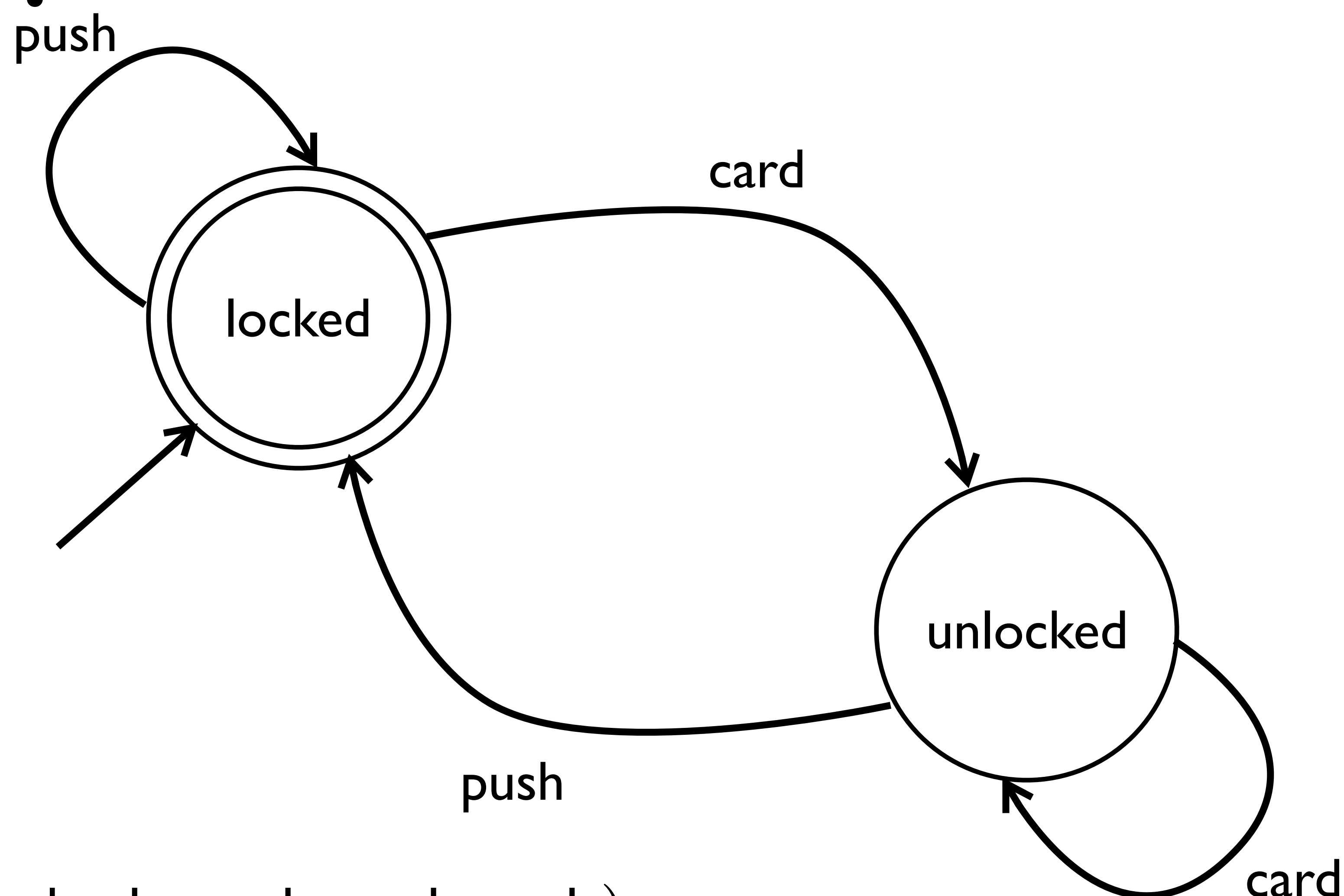
$$\widehat{\delta}(q, \epsilon) \triangleq q$$

inductive case: For any $q \in Q, a \in \Sigma, w \in \Sigma^*$ we let

$$\widehat{\delta}(q, \underbrace{\boxed{wa}}_{\text{More complex case}}) \triangleq \delta(\ \widehat{\delta}(q, \underbrace{\boxed{w}}_{\text{Simpler case}})\ ,\ a\ )$$

($\widehat{\delta}(q, w)$ returns the state reached from q by observing w)

Example: Turnstile



$$
\begin{aligned}
&\widehat{\delta}(\text{locked}, \text{card card push})\\
&= \delta(\widehat{\delta}(\text{locked}, \text{card card}),\ \text{push})\\
&= \delta(\delta(\widehat{\delta}(\text{locked}, \text{card}),\ \text{card}),\ \text{push})\\
&= \delta(\delta(\delta(\widehat{\delta}(\text{locked}, \epsilon),\ \text{card}),\ \text{card}),\ \text{push})\\
&= \delta(\delta(\delta(\text{locked},\ \text{card}),\ \text{card}),\ \text{push})\\
&= \delta(\delta(\text{unlocked},\ \text{card}),\ \text{push})\\
&= \delta(\text{unlocked},\ \text{push}) = \text{locked}
\end{aligned}
$$

$\delta(\text{locked}, \text{card}) = \text{unlocked}$

$\delta(\text{locked}, \text{push}) = \text{locked}$

$\delta(\text{unlocked}, \text{card}) = \text{unlocked}$

$\delta(\text{unlocked}, \text{push}) = \text{locked}$

String processing

Given $A = (Q, \Sigma, \delta, q_0, F)$ and $w \in \Sigma^*$ we say that A **accept** w iff

$$\widehat{\delta}(q_0, w) \in F$$

The **language** of $A = (Q, \Sigma, \delta, q_0, F)$ is

$$L(A) = \{ \ w \ \mid \ \widehat{\delta}(q_0, w) \in F \ \}$$

Transition diagram

We represent $A = (Q, \Sigma, \delta, q_0, F)$ as a graph s.t.

- Q is the set of nodes;
- $\{\, q \xrightarrow{a} q' \mid q' = \delta(q, a) \,\}$ is the set of arcs.

Plus some graphical conventions:

- there is one special arrow $Start$ with $\xrightarrow{Start} q_0$
- nodes in F are marked by double circles;
- nodes in $Q \setminus F$ are marked by single circles.

String processing as paths

A DFA accepts a string w, if there is a path in its transition diagram such that:

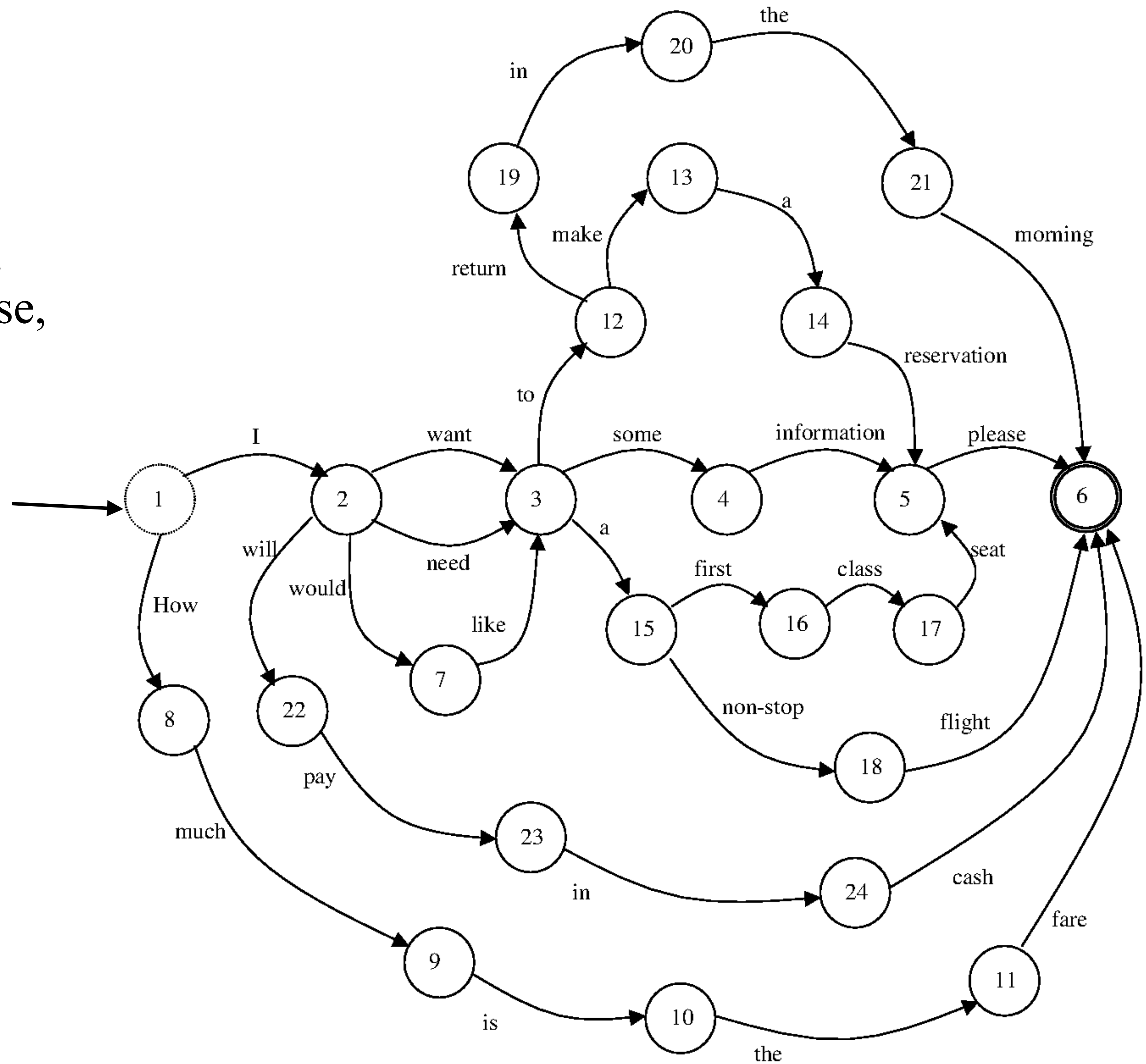
it starts from the initial state

it ends in one final state

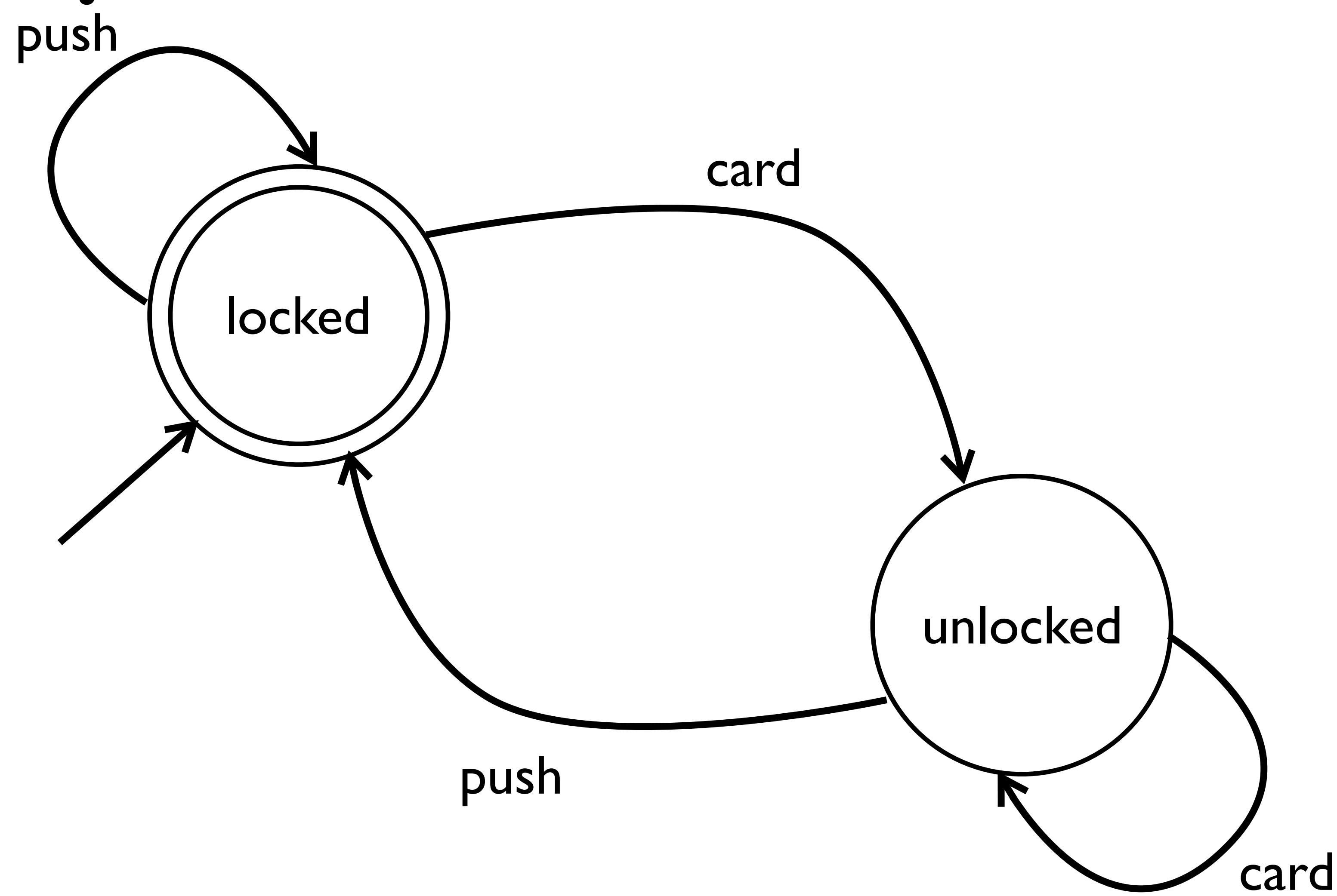
the sequence of labels in the path is exactly w

Example: Language Processing

L(A) = {
How much is the fare,
I will pay in cash,
I would like a non-stop flight,
I want some information please,
…
}



DFA: question time



What is L(A) ?

Transition table

Conventional tabular representation

its rows are in correspondence with states

its columns are in correspondence with input symbols

its entries are the states reached after the transition

Plus some decoration

start state decorated with an arrow

all final states decorated with *

Transition table

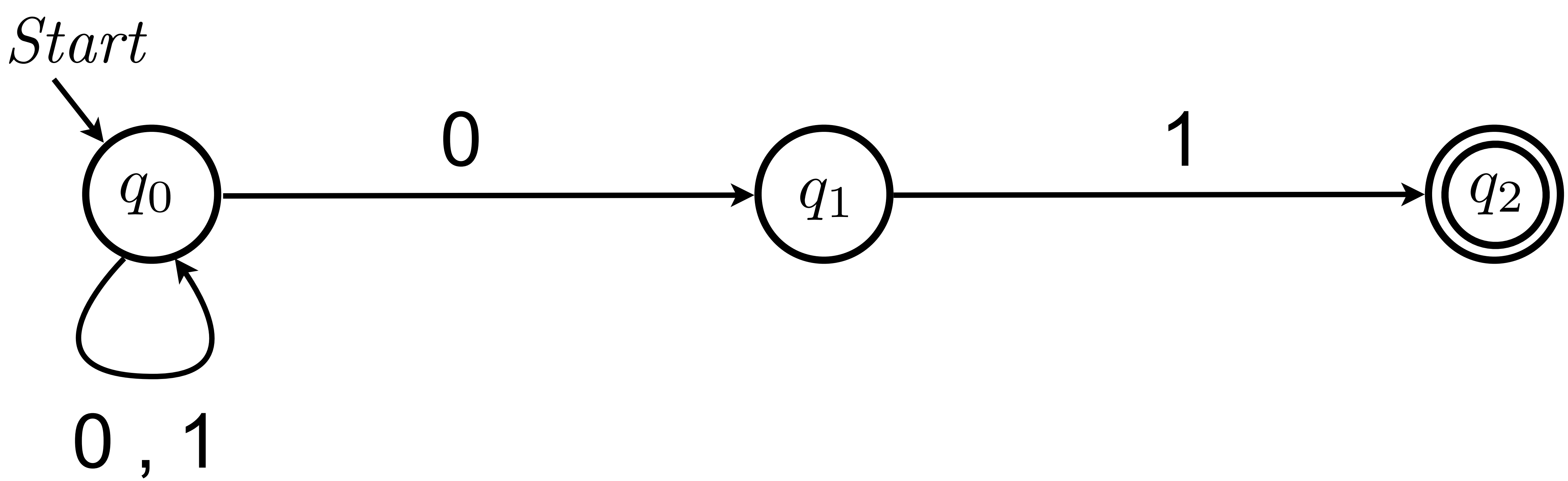
					a			
$\rightarrow$								
	q				$\delta(q, a)$			
$*$								
$*$								

NFA

A **Non-deterministic Finite Automaton** (**NFA**) is a tuple $A = (Q, \Sigma, \delta, q_0, F)$, where

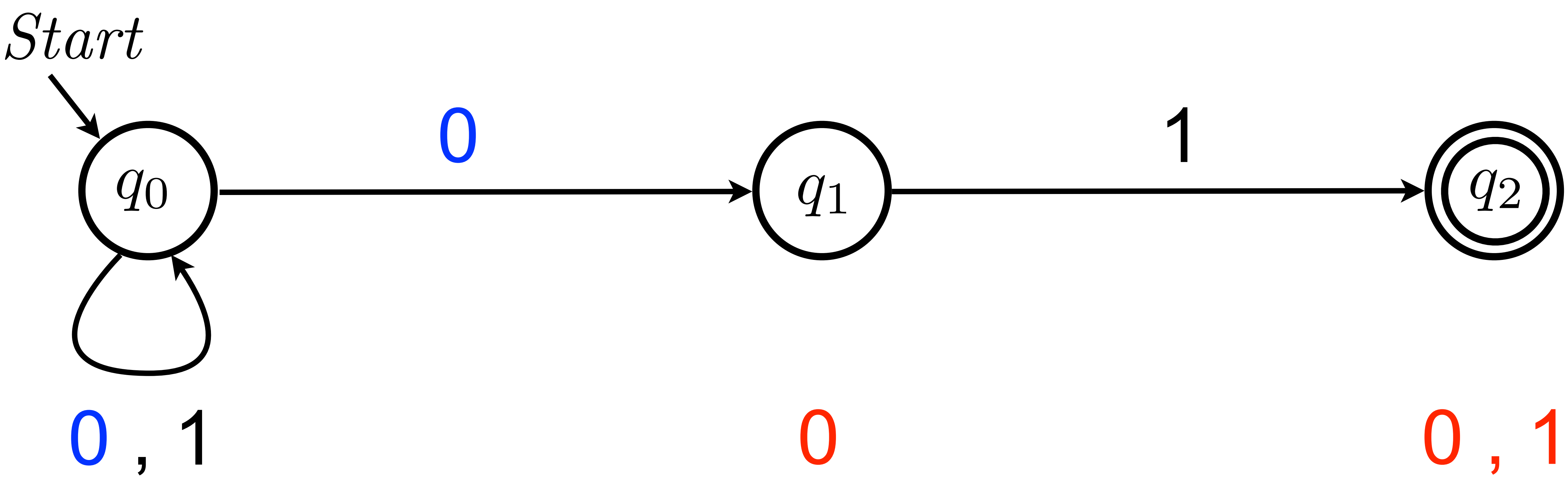
- Q is a finite set of states;
- Σ is a finite set of input symbols;
- $\delta : Q \times \Sigma \to \wp(Q)$ is the transition function; powerset of Q = set of sets over Q
- $q_0 \in Q$ is the initial state (also called start state);
- $F \subseteq Q$ is the set of final states (also accepting states)

NFA: example



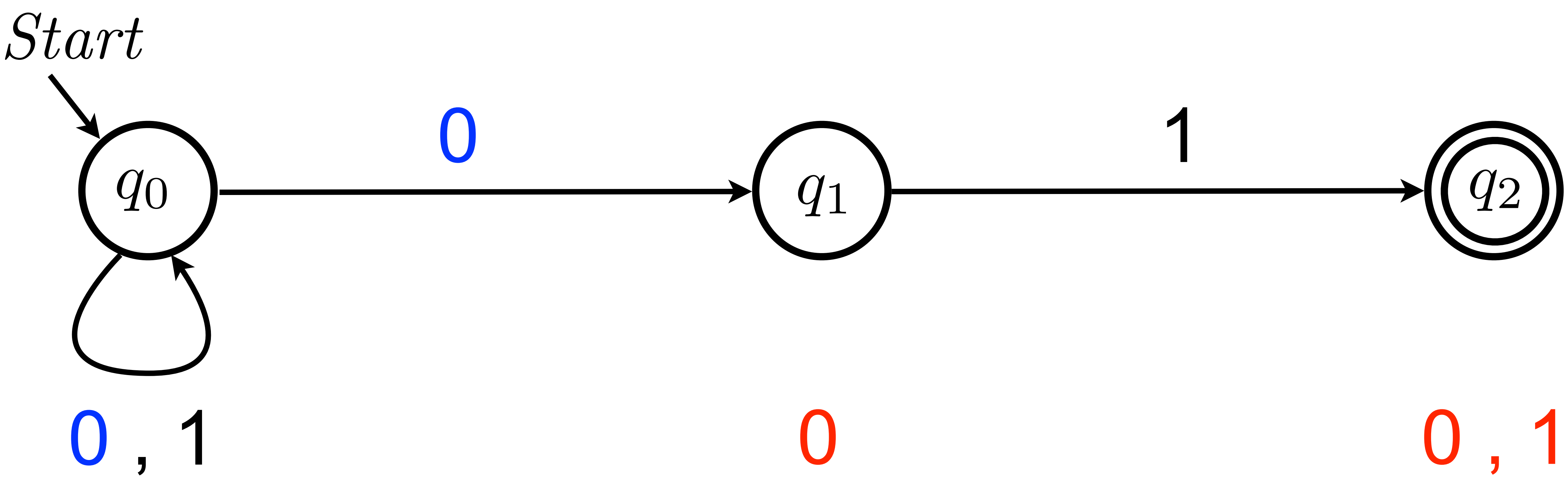
Can you explain why it is not a DFA?

NFA: example



Can you explain why it is not a DFA?

NFA: example



$$\delta(q_0, 0) = \{q_0, q_1\}$$
$$\delta(q_0, 1) = \{q_0\}$$

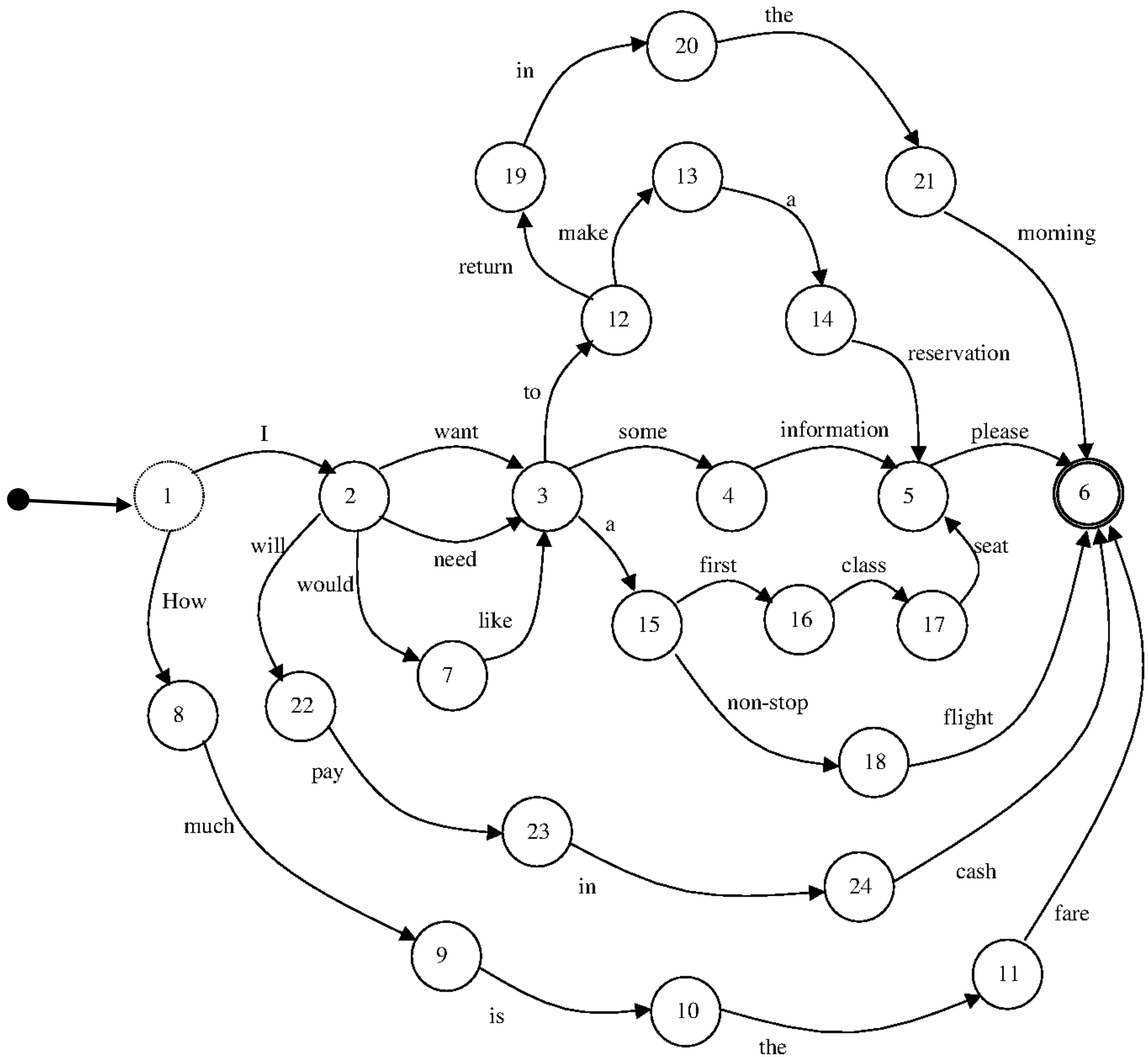
$$\delta(q_1, 0) = \emptyset$$
$$\delta(q_1, 1) = \{q_2\}$$

$$\delta(q_2, 0) = \delta(q_2, 1) = \emptyset$$

Can you explain why it is not a DFA?

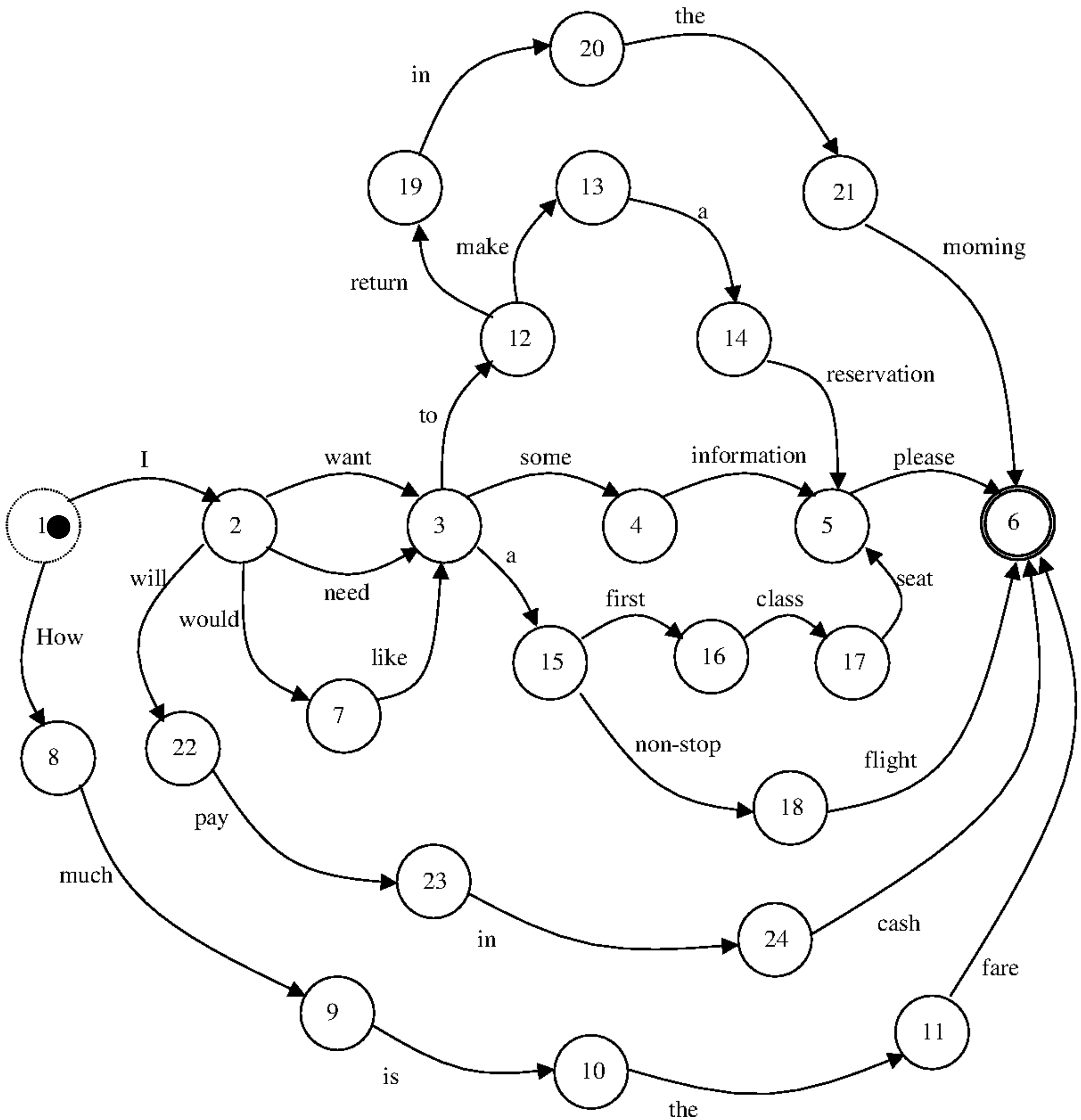
Reshaping

#1: get a token

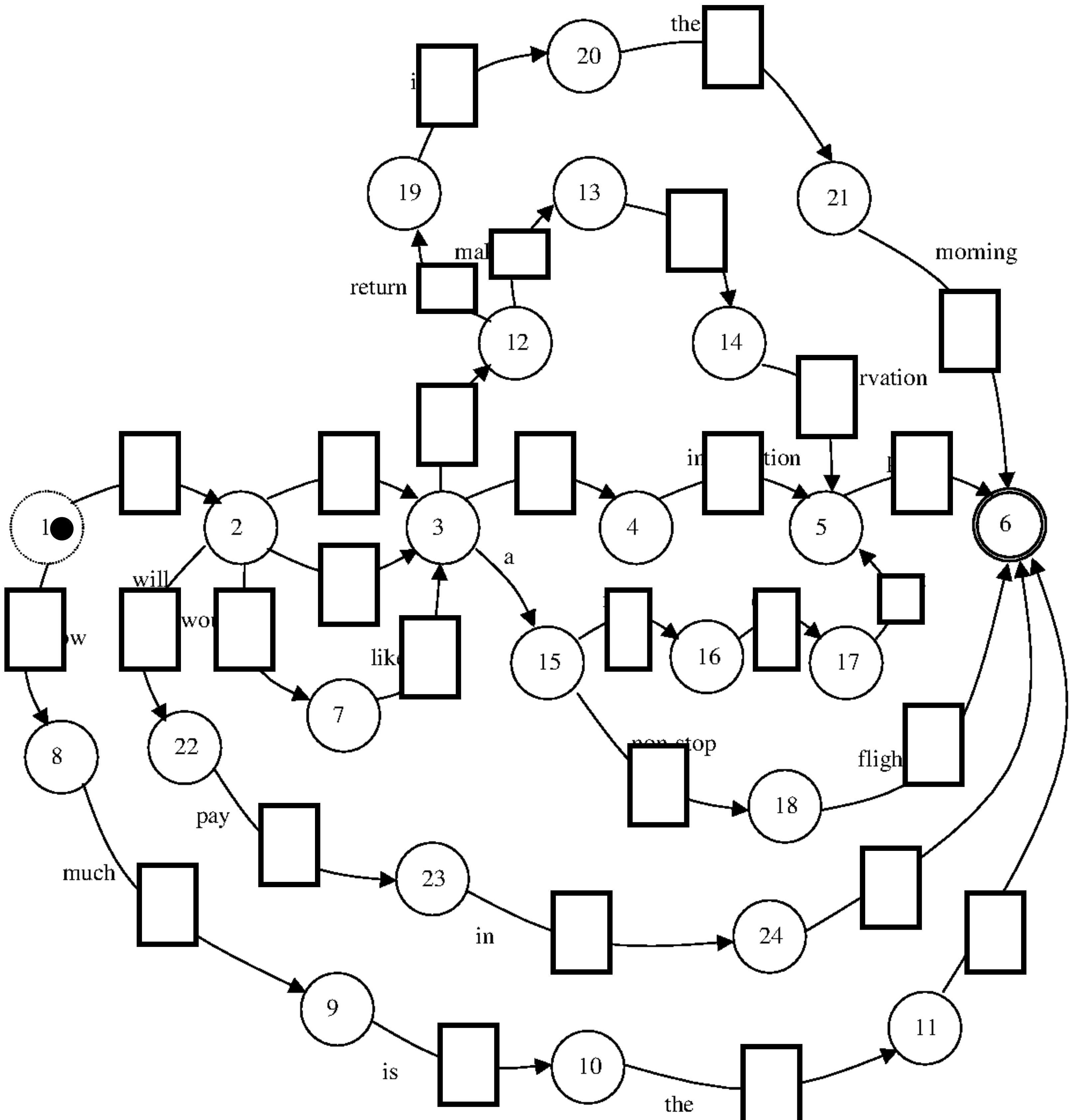


I
need
to
make
a
reservation
please

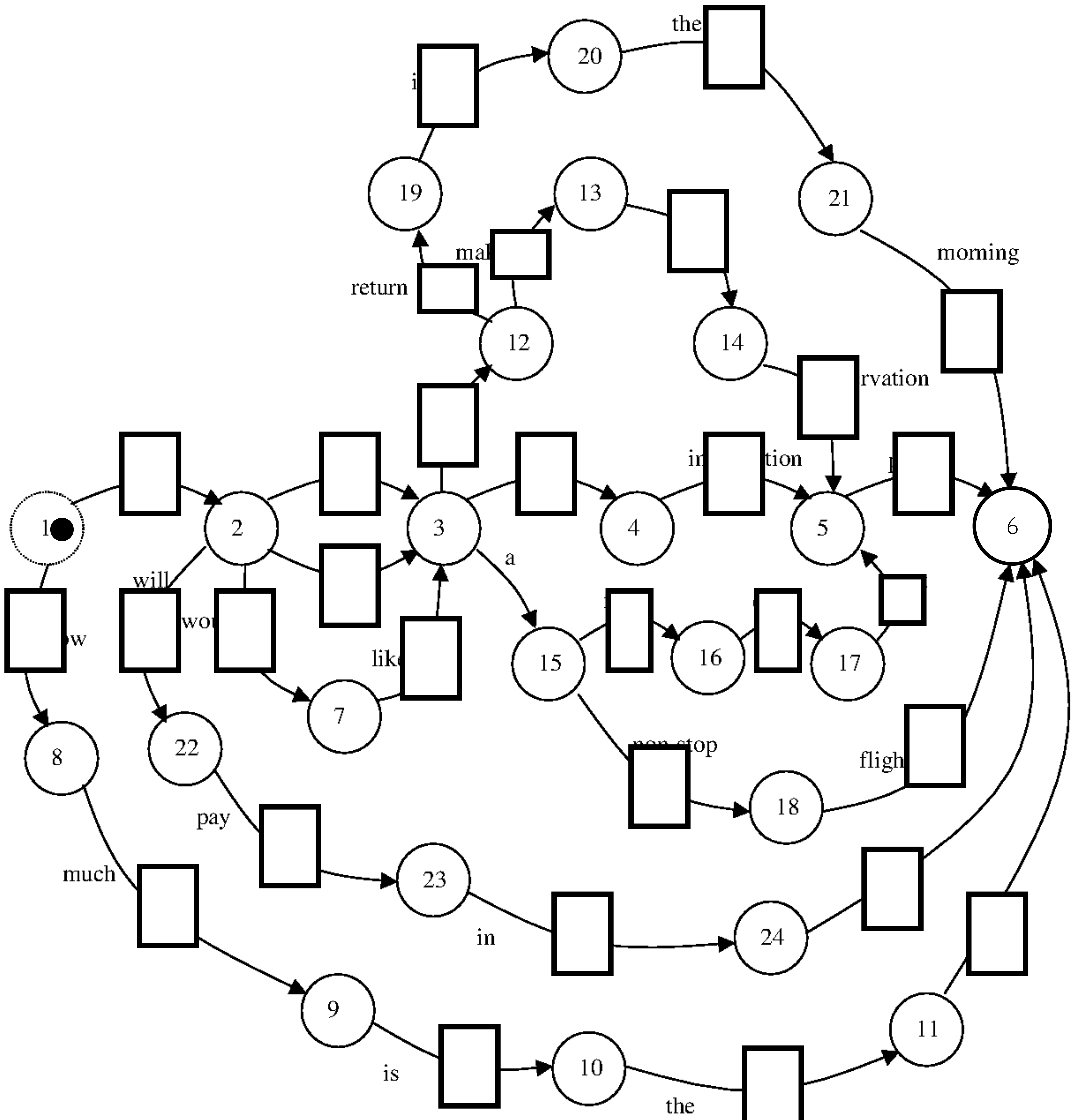
#2: drop initial state arrow



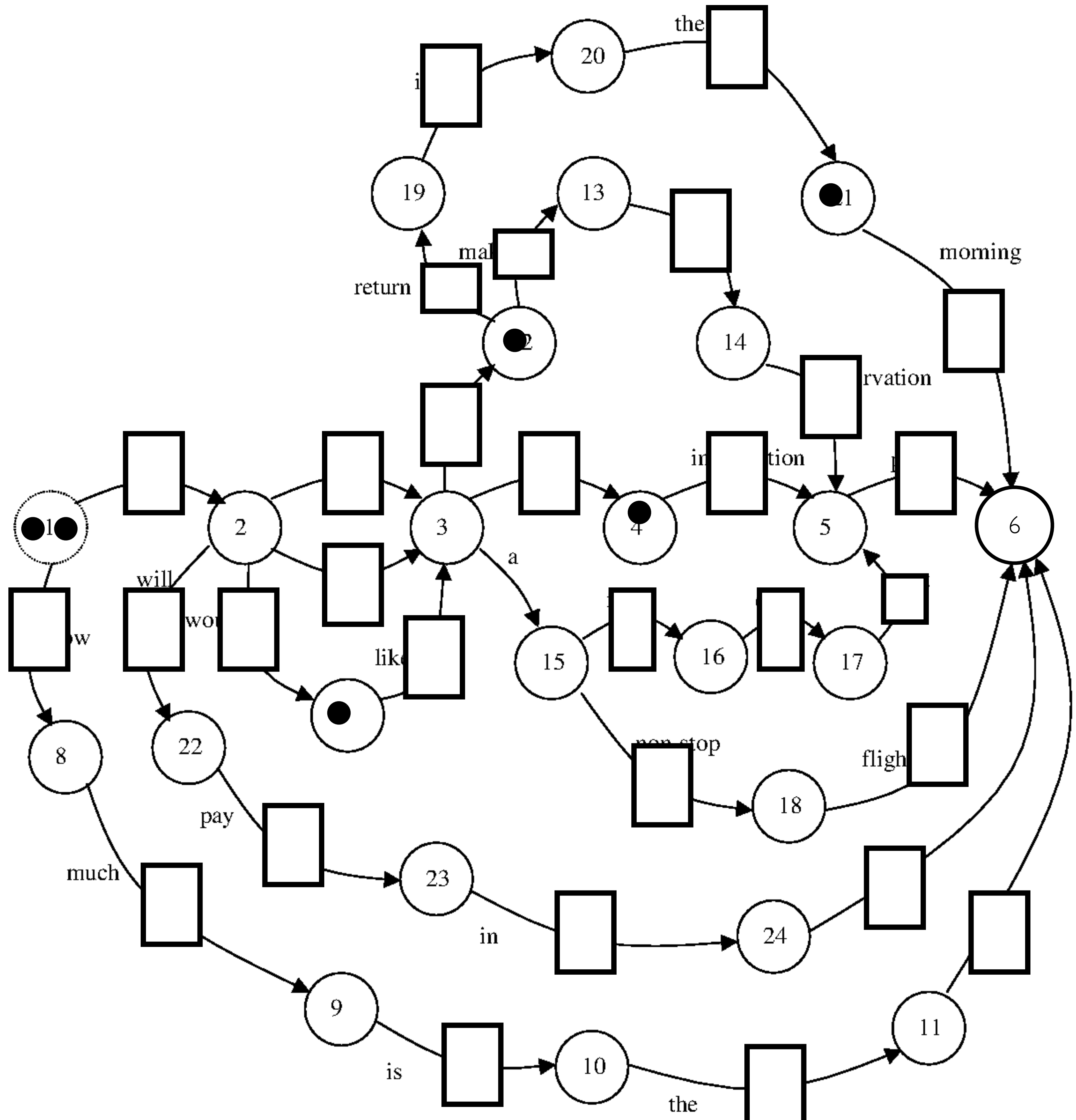
#3: transitions as boxes



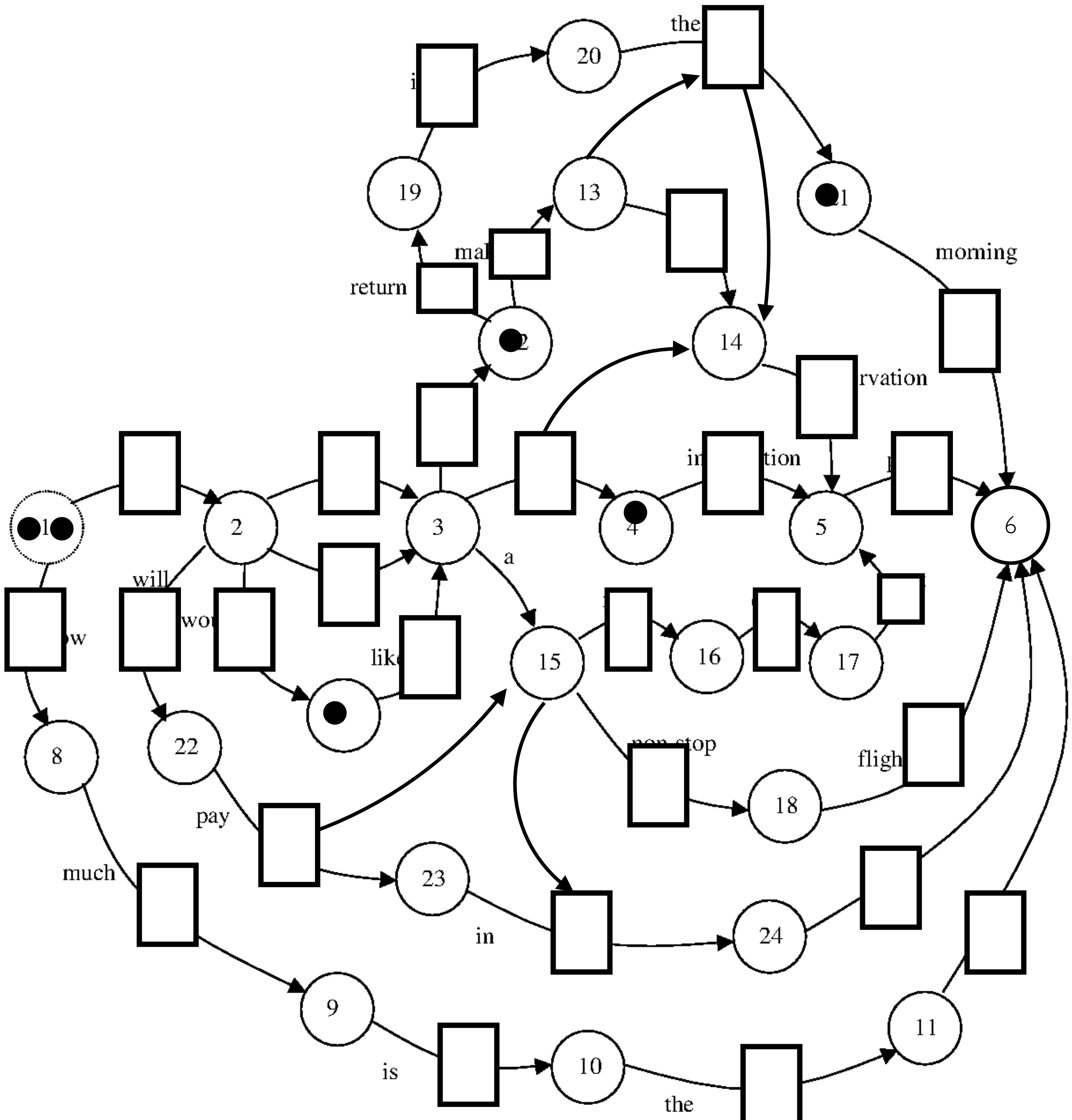
#4: forget final states



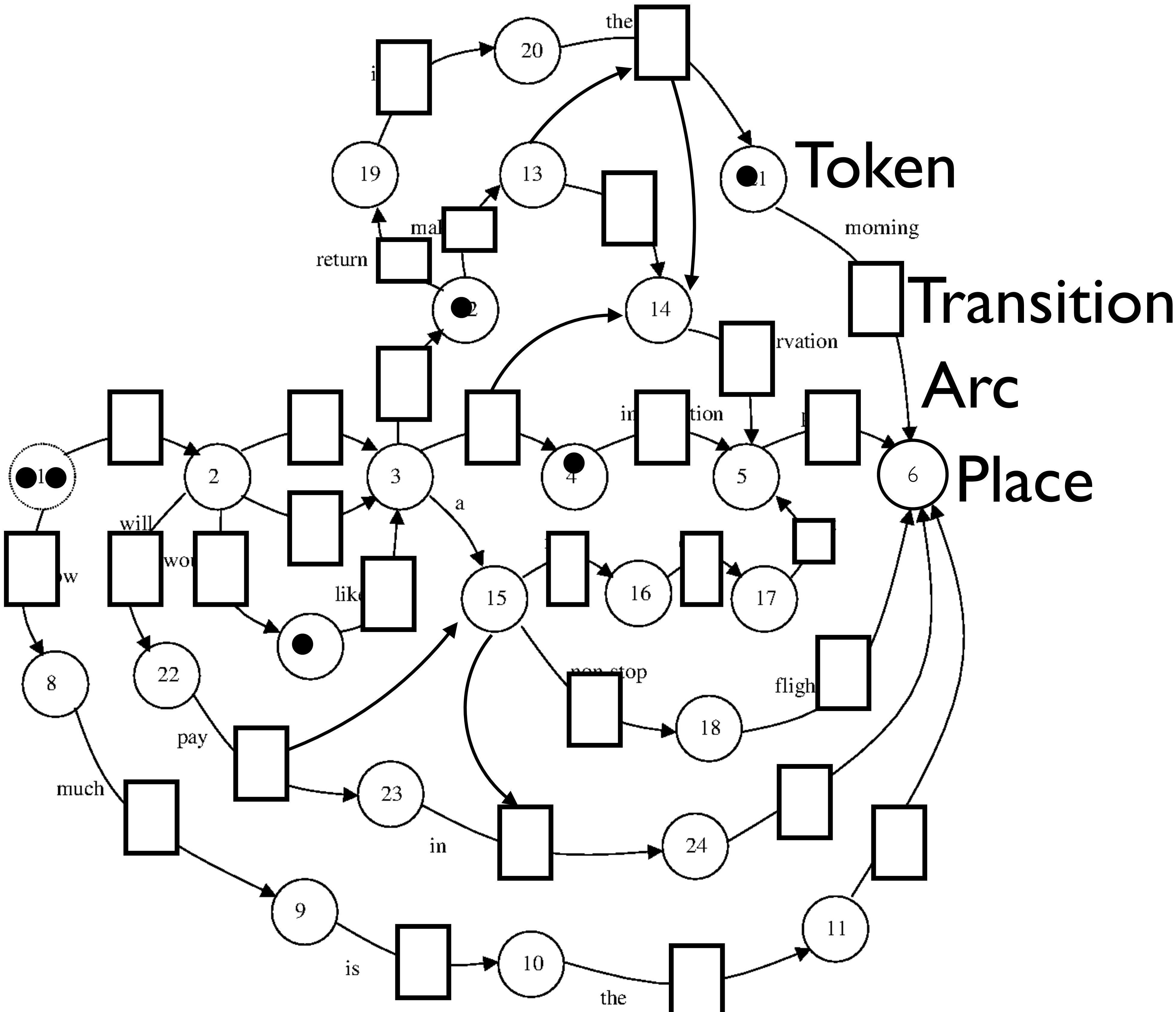
#5: add more tokens



#6: allow for more arcs



Terminology



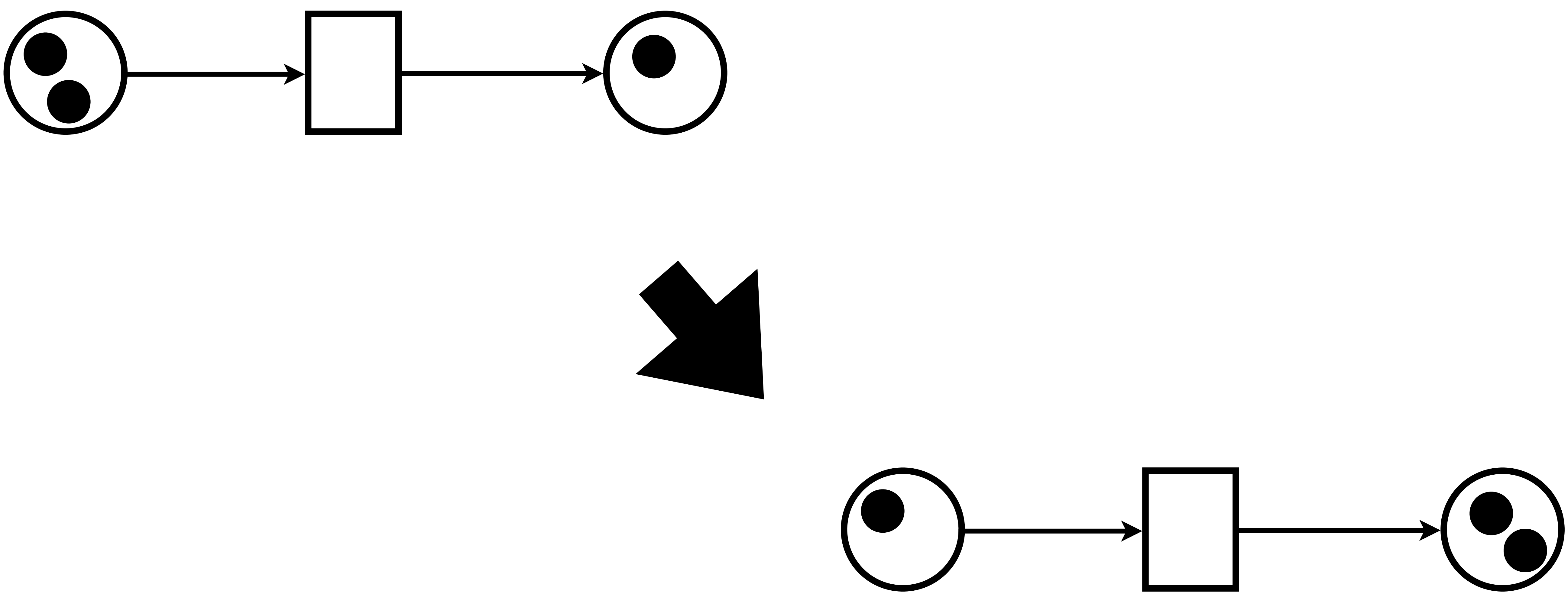
Some facts

Nets are **bipartite graphs**:
arcs never connect two places
arcs never connect two transitions

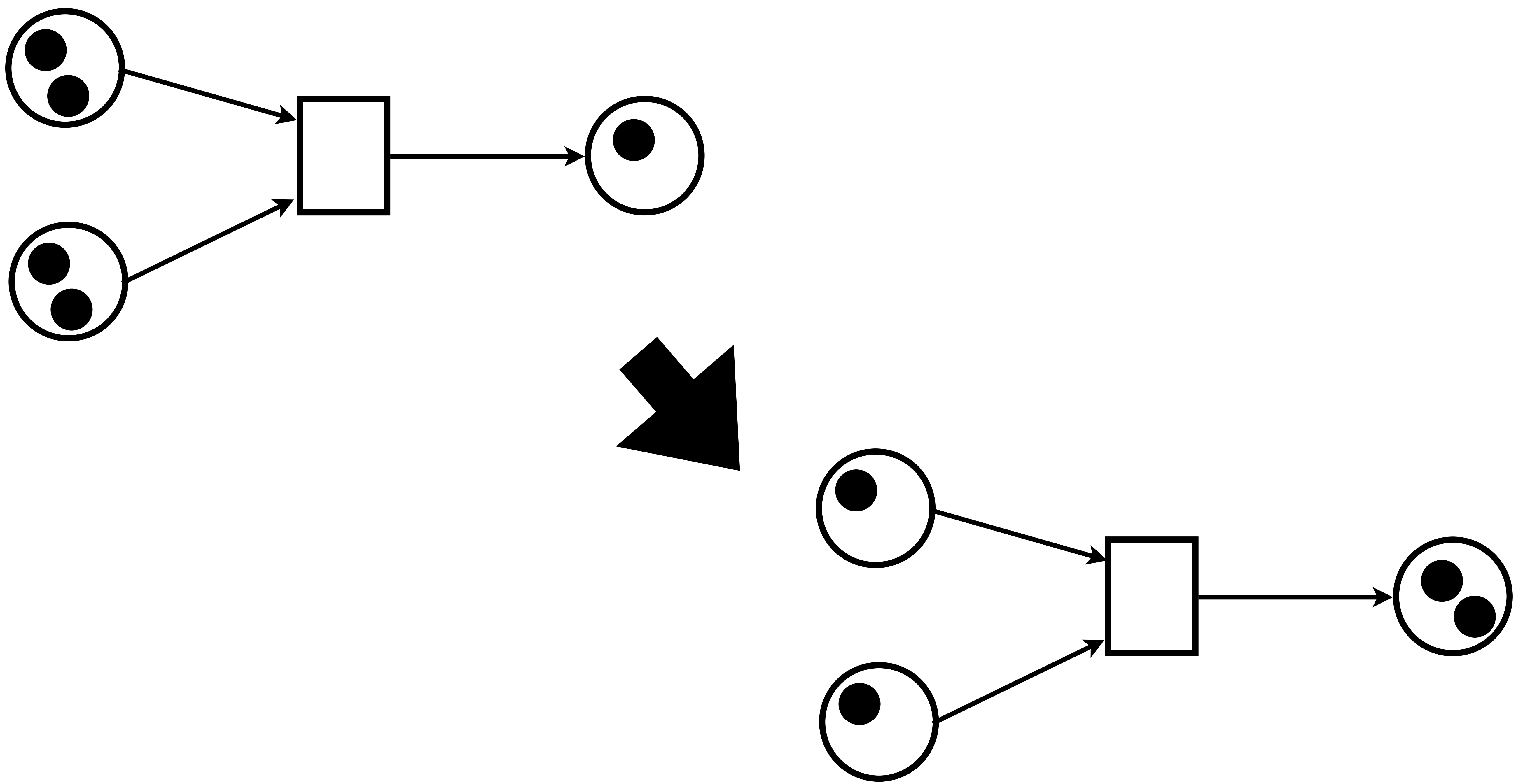
Static structure for dynamic systems:
places, transitions, arcs do not change
tokens move around places

Places are passive components
Transitions are active components:
tokens do not flow!
(they are removed or freshly created)

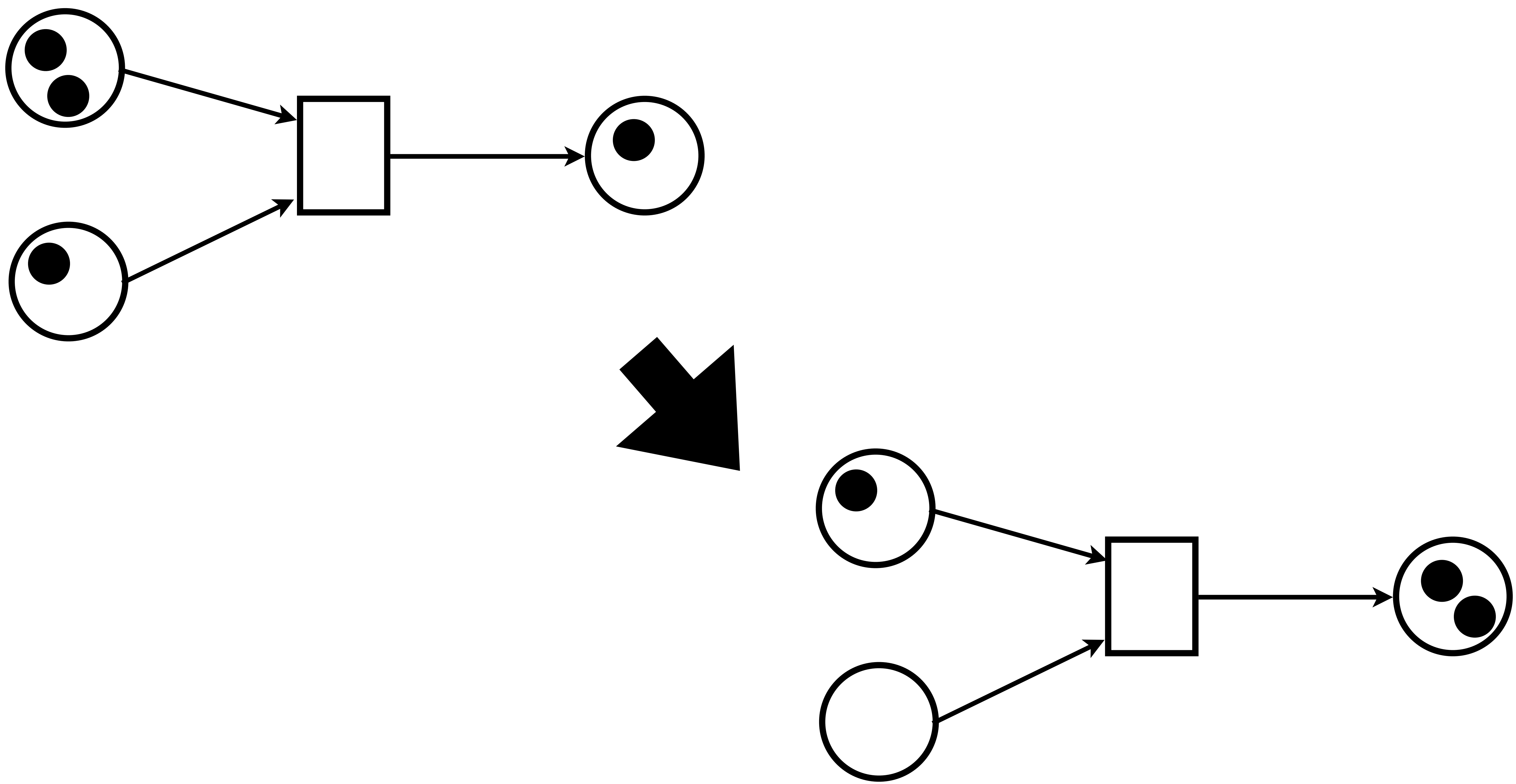
Token game: example



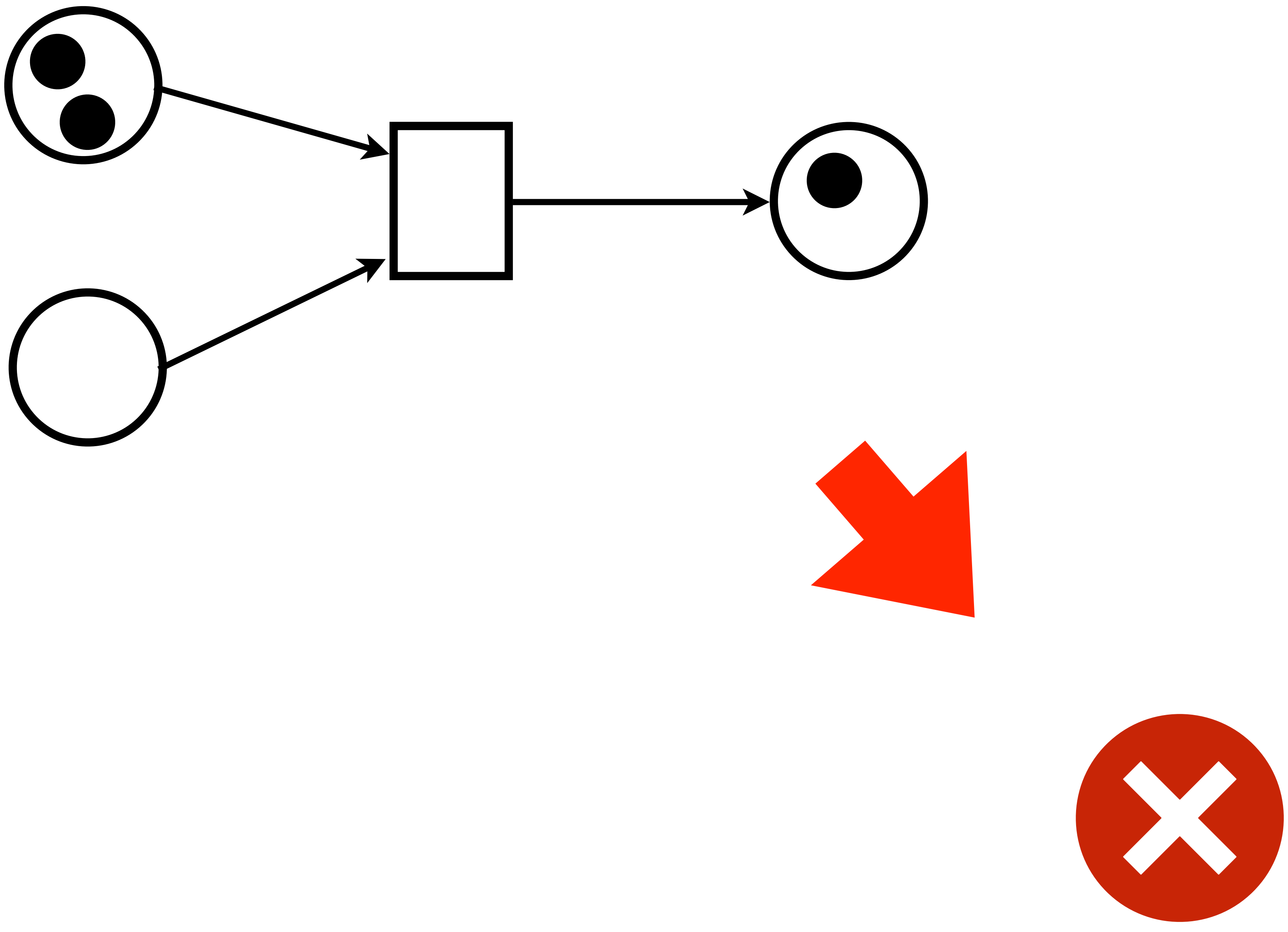
Token game: example



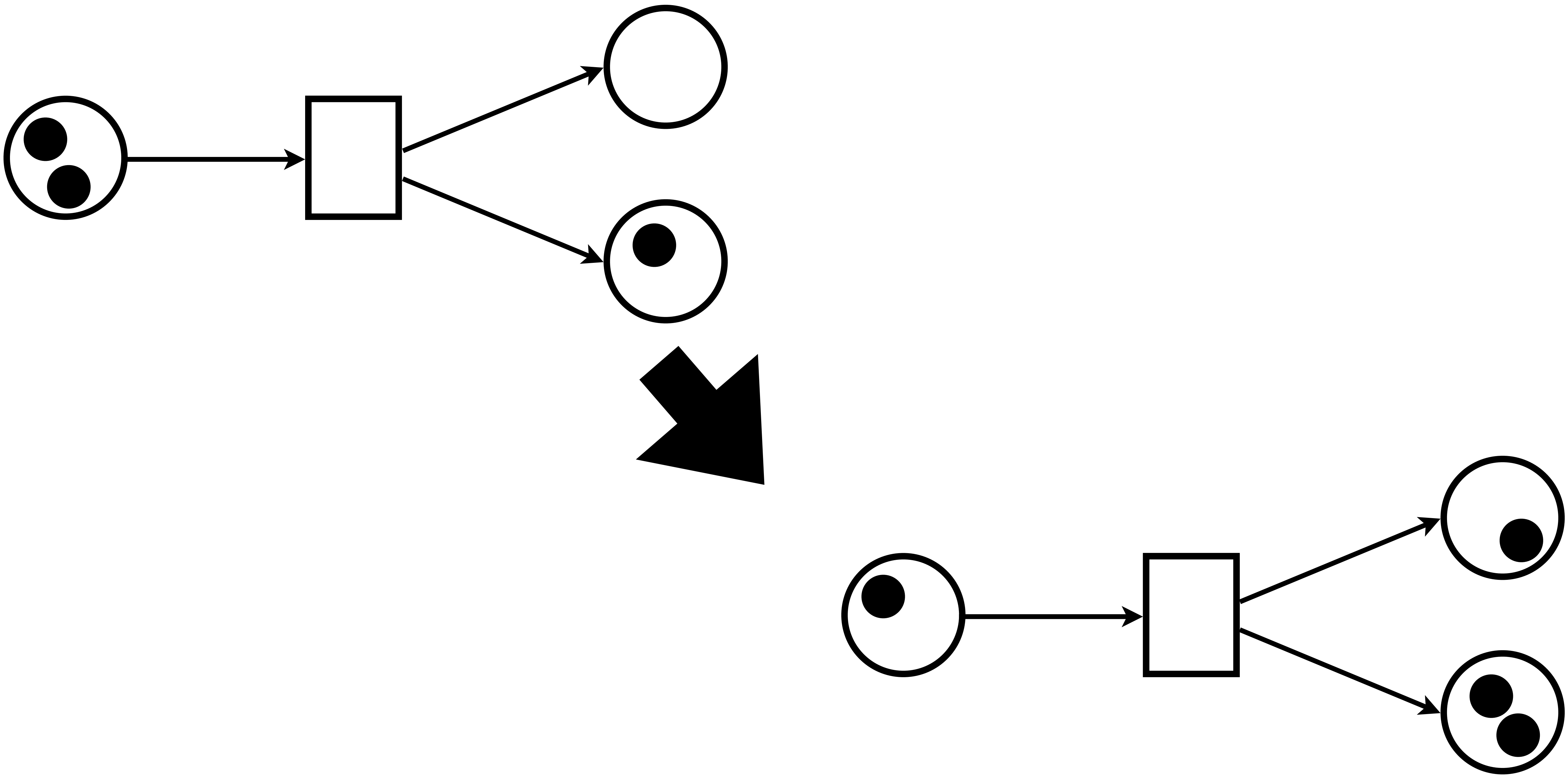
Token game: example



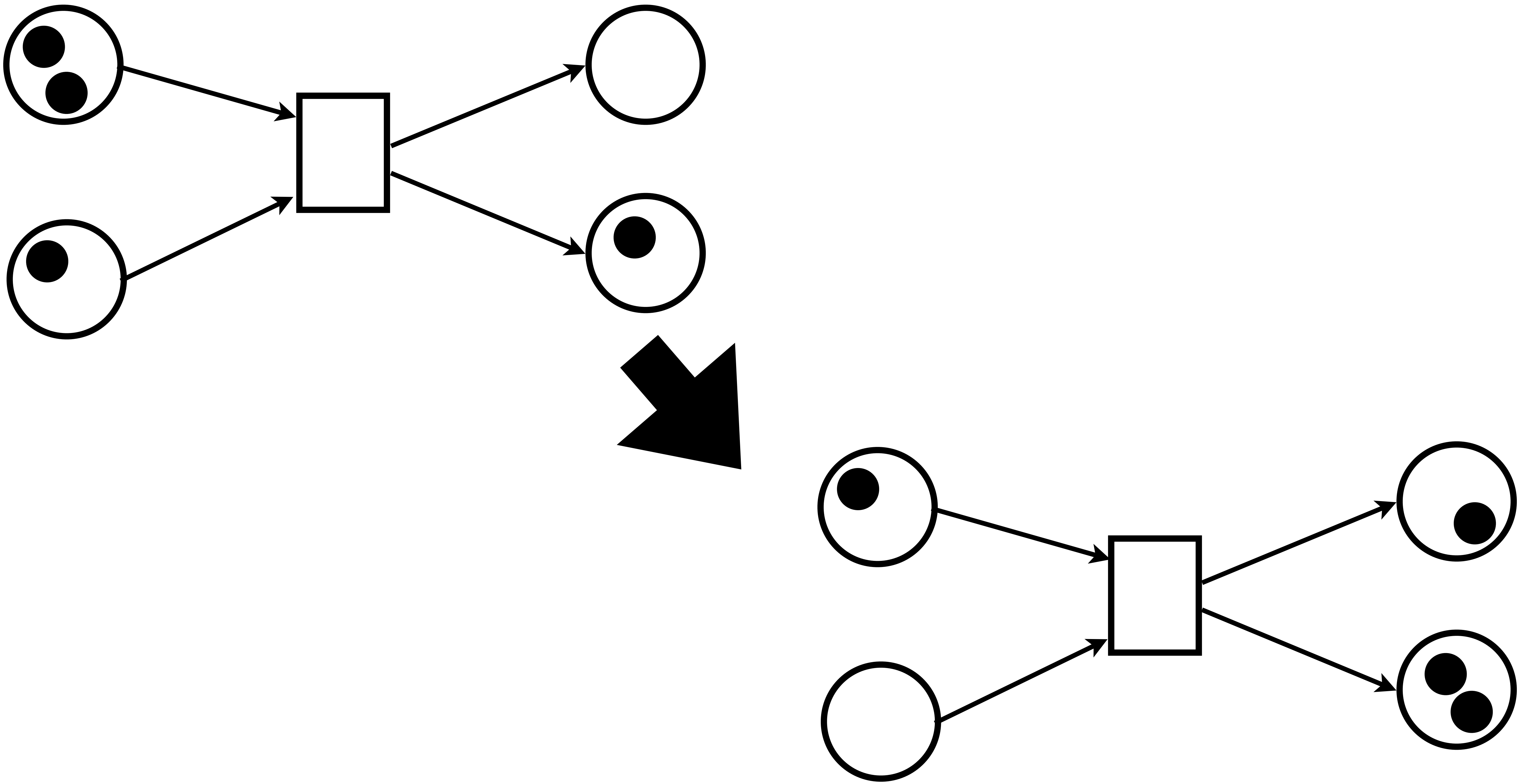
Token game: example



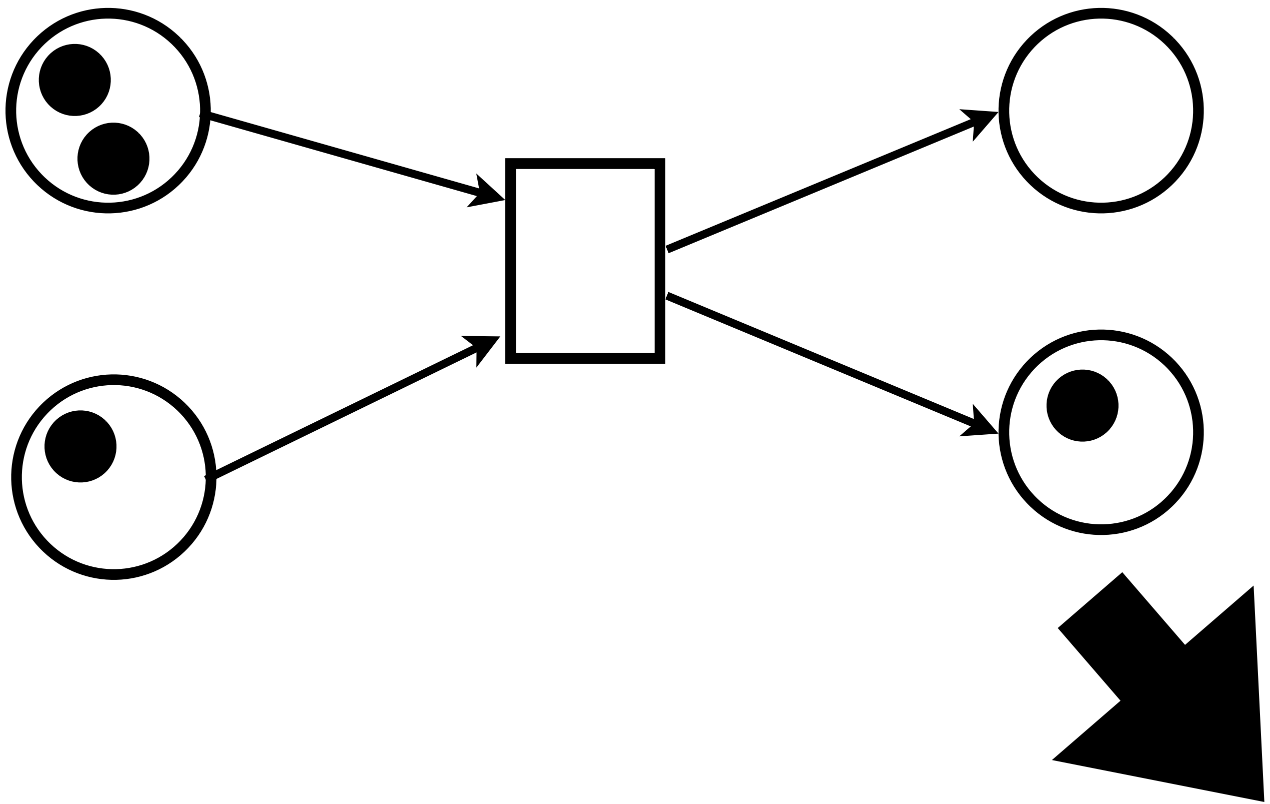
Token game: example



Token game: example

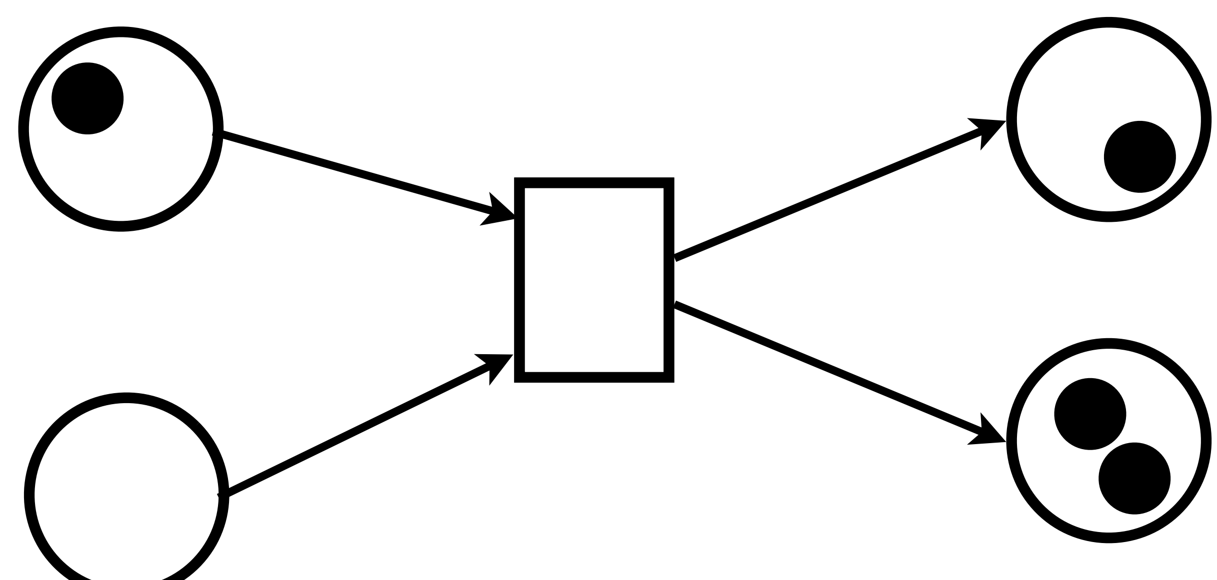


Token game: firing rule

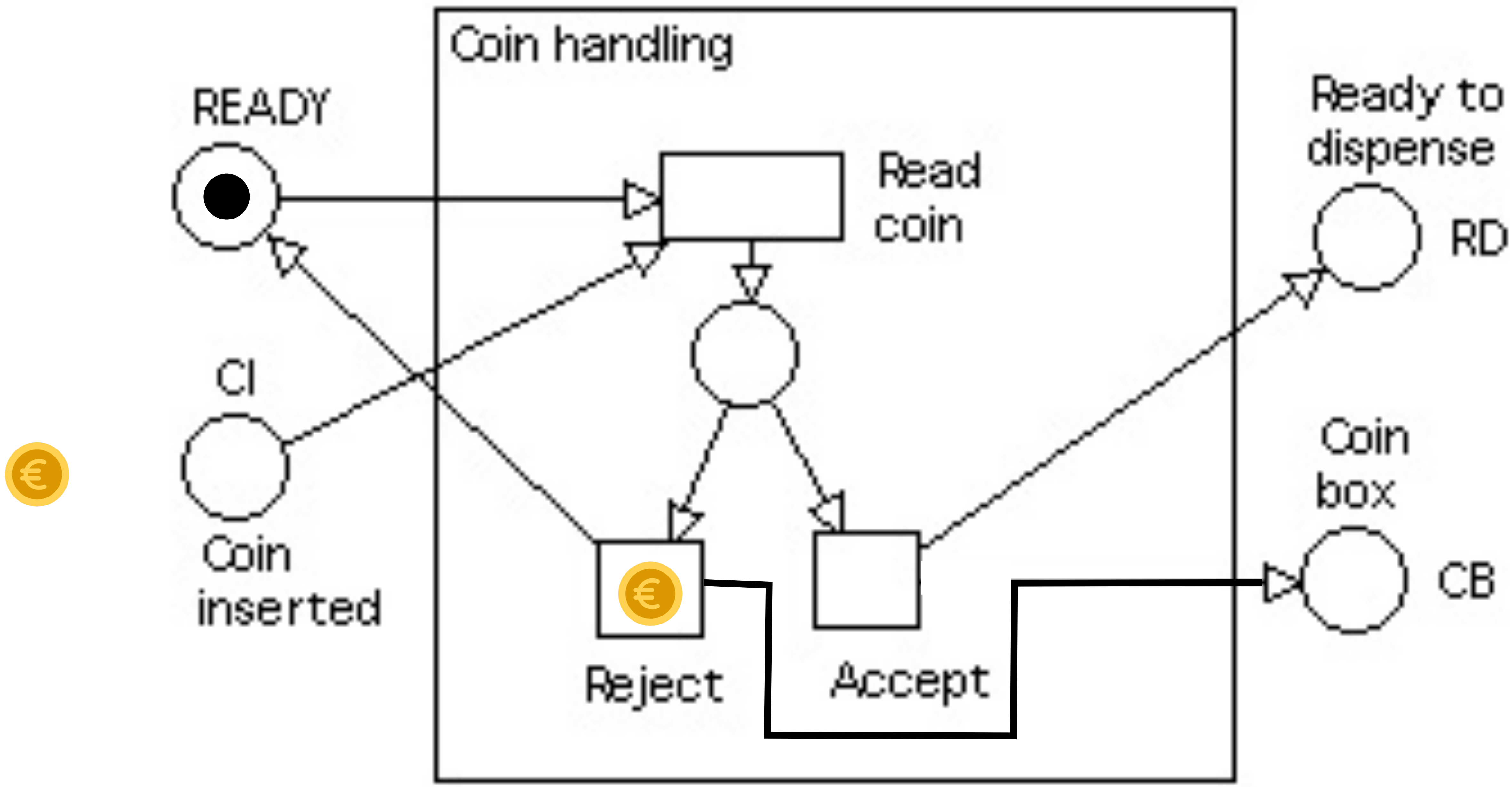


Collect one token from each “input” place

Produce one token into each “output” place

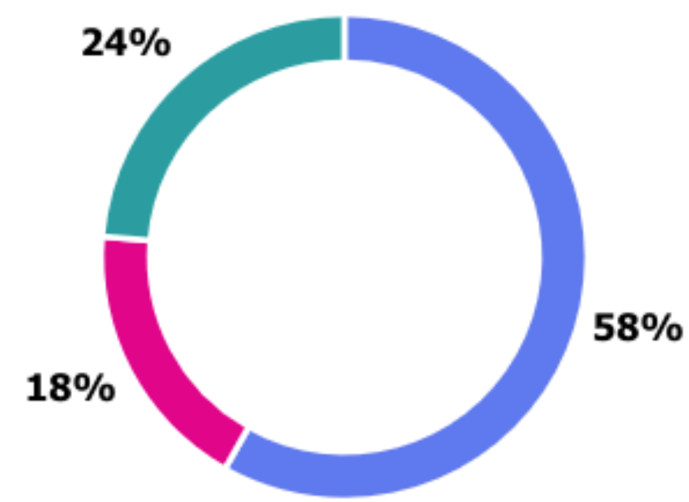


Example: Coin Handling



11. Do you agree that a multiset M of elements in A can be seen as a function from A to the set of natural numbers? Più dettagli

Yes	32
No	10
I do not understand the question	13



Notation: from sets...

Let S be a set.
Let $\wp(S)$ denote the set of sets over S.

Elements $A \in \wp(S)$ (i.e., $A \subseteq S$)
are in bijective correspondence with
functions $f : S \to \{0, 1\}$

$x \in A$ iff $f_A(x) = 1$

Sets vs Multisets

Set

Multiset

Order of elements does not matter

Each element appears at most once

Order of elements does not matter

Each element can appear multiple times

Notation: ... to multisets

Let $\mu(S)$ (or $S^{\oplus}$) denote the set of multisets over S.

Elements $M \in \mu(S)$ are in bijective correspondence with functions $M : S \to \mathbb{N}$

$f_M(x)$ is the number of instances of x in M
$x \in M$ iff $f_M(x) > 0$

Example: sets

$f_A($  $) = 1$

$f_A($  $) = 1$

$f_A($  $) = 1$

$f_A($  $) = 0$

Example: multisets

$M($  $) = 4$

$M($  $) = 2$

$M($  $) = 1$

$M($  $) = 0$

Notation: multisets

Multiset $M = \{ k_1 x_1, k_2 x_2, ..., k_n x_n \}$

element

multiplicity
(positive, omitted if 1)

{ 3 , 2 , 1  }

or just

{ 3 , 2 ,  }

Notation: multisets

Multiset $M = \{\ k_1 x_1, k_2 x_2, ..., k_n x_n\}$ as formal sum:

$$k_1 x_1 + k_2 x_2 + ... + k_n x_n$$

3  +2  + 

$$\sum_{i=1}^{n} k_i x_i$$

As set is just a special case: multiplicity is either 0 or 1

$$x_{i_1} + x_{i_2} + \ldots + x_{i_k}$$

$$\sum_{j=1}^{k} x_{i_k}$$

Marking

A **marking** $M : P \to \mathbb{N}$ denotes the number of tokens in each place

The marking of a Petri net represents its state

$M(a) = 0$ denotes the absence of tokens in place a

Notation: sets

Empty set:
$\emptyset = \{\ \}$ is such that $x \notin \emptyset$ for all $x \in S$

Set inclusion:
we write $A \subseteq B$ if $x \in A$ implies $x \in B$

Set strict inclusion:
we write $A \subset B$ if $A \subseteq B$ and $A \neq B$

Set union:
$A \cup B$ is the set s.t. $x \in (A \cup B)$ iff $x \in A$ or $x \in B$

Set difference:
$A - B$ is the set s.t. $x \in (A - B)$ iff $x \in A$ and $x \notin B$

Notation: multisets

Empty multiset:
$\emptyset$ is such that $\emptyset(x) = 0$ for all $x \in S$

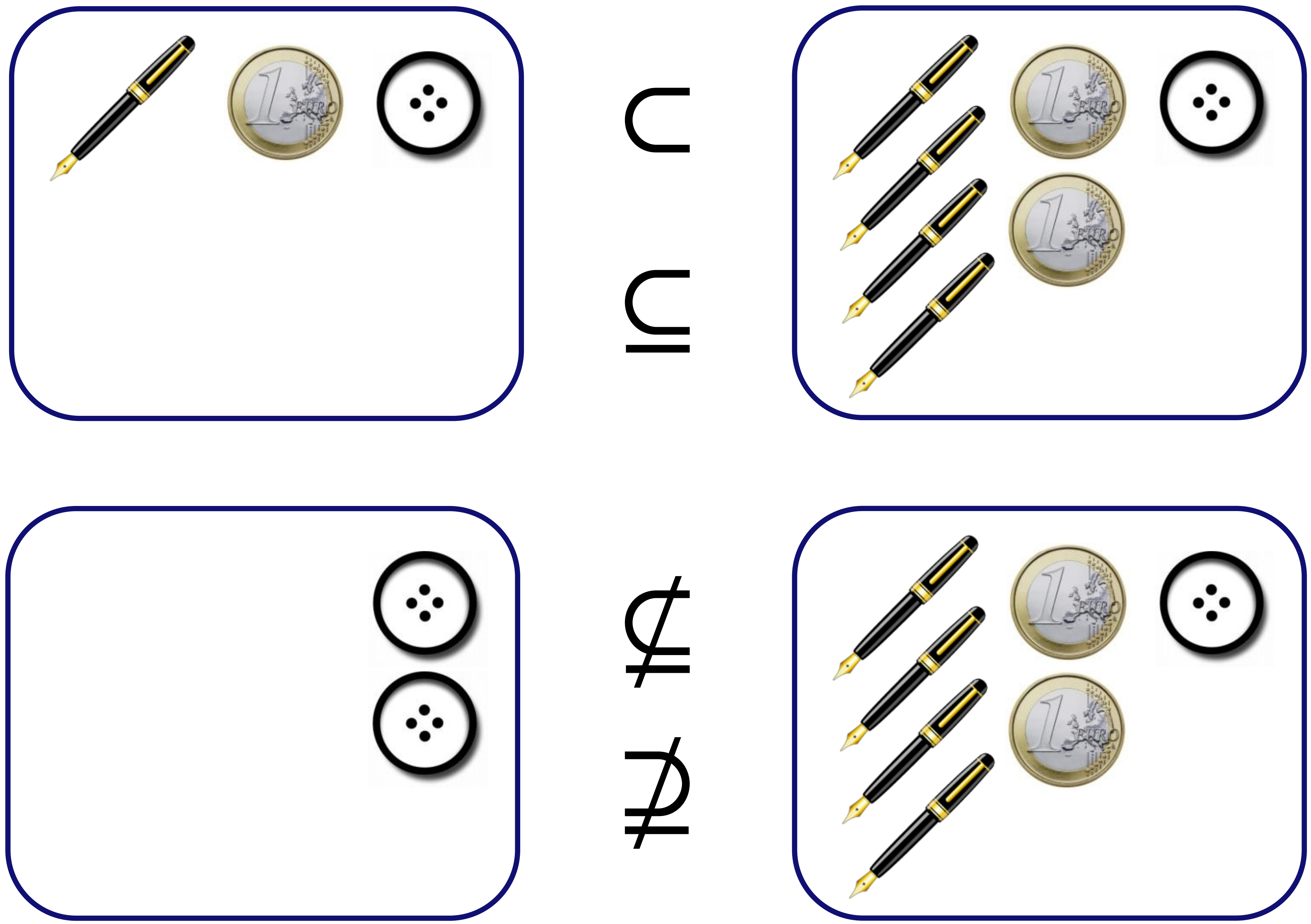
Multiset containment:
we write $M \subseteq M'$ if $M(x) \leq M'(x)$ for all $x \in S$

Multiset strict containment:
we write $M \subset M'$ if $M \subseteq M'$ and $M \neq M'$

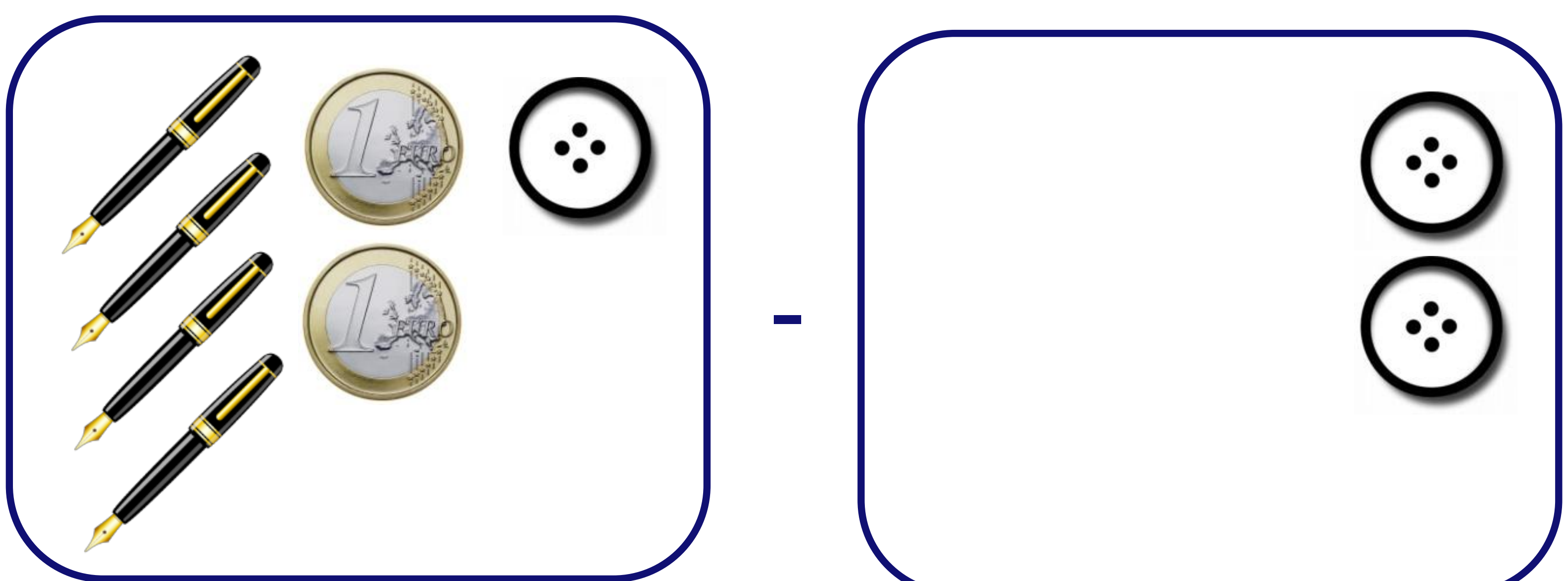
Multiset union:
$M + M'$ is the multiset s.t. $(M + M')(x) = M(x) + M'(x)$ for all $x \in S$

Multiset difference (defined only if $M \supseteq M'$):
$M - M'$ is the multiset s.t. $(M - M')(x) = M(x) - M'(x)$ for all $x \in S$

Multiset containment



Operations on Multisets



undefined

Question time

$$3a + 2b \stackrel{?}{\subseteq} 2a + 3b + c$$

$$3a + 2b \stackrel{?}{\supseteq} 2a + 3b + c$$

$$a + 2b \stackrel{?}{\subset} 2a + 3b$$

$$(a + 2b) + (2a + c) = ?$$

$$(2a + 3b) - (2a + b) = ?$$

$$(2a + 2b) - (a + c) = ?$$

Question time

$$3a + 2b \overset{?}{\subseteq} 2a + 3b + c$$ No

$$3a + 2b \overset{?}{\supseteq} 2a + 3b + c$$ No

$$a + 2b \overset{?}{\subset} 2a + 3b$$ Yes

$$(a + 2b) + (2a + c) = ?$$ $3a + 2b + c$

$$(2a + 3b) - (2a + b) = ?$$ $2b$

$$(2a + 2b) - (a + c) = ?$$ Not defined

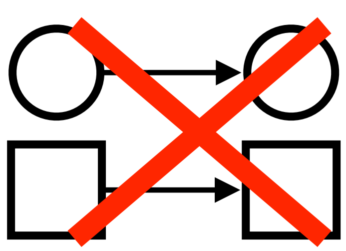
Petri nets

A **Petri net** is a tuple (P, T, F, M_0) where

- P is a finite set of **places**;

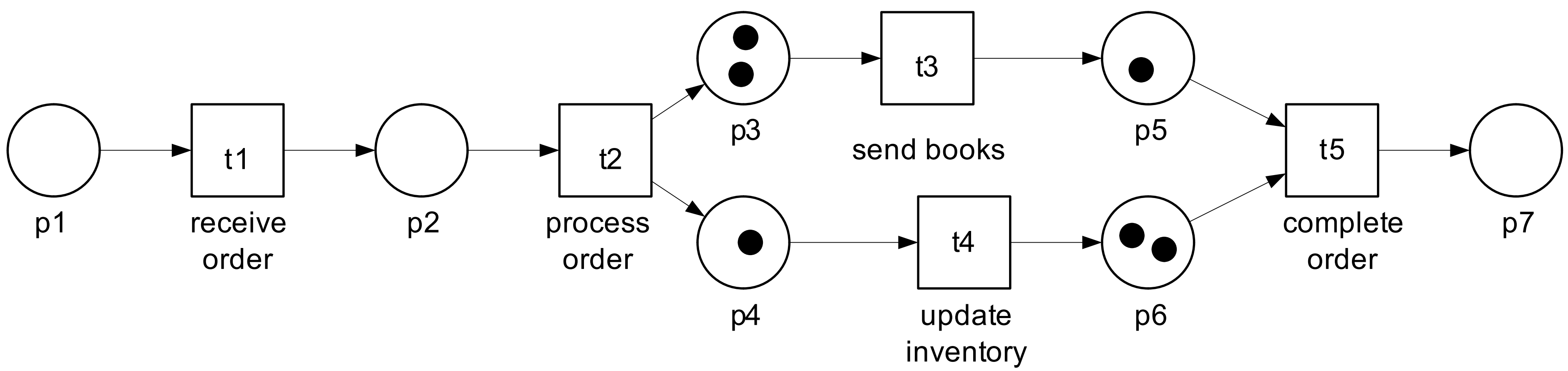
$P \cap T = \emptyset$

- T is a finite set of **transitions**;



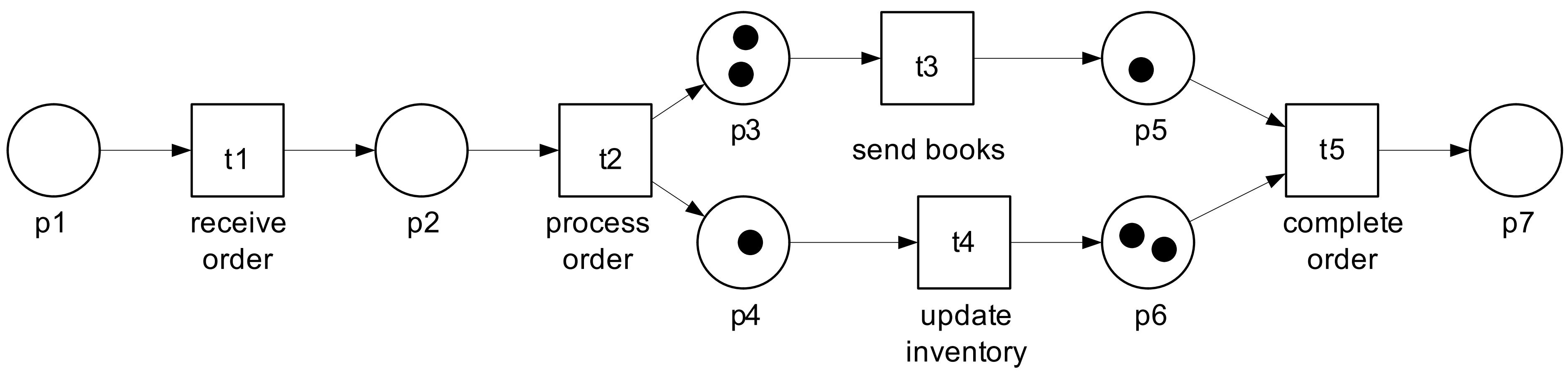
- $F \subseteq (P \times T) \cup (T \times P)$ is a **flow relation**;
- $M_0 : P \to \mathbb{N}$ is the **initial marking**.
  (i.e. $M_0 \in \mu(P)$)

Exercises



$$P = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\}$$
$$T = \{t_1, t_2, t_3, t_4, t_5\}$$
$$F = \{(p_1, t_1), (t_1, p_2), \ldots ?\}$$
$$M_0 = 2p_3 + \ldots ?$$

Exercises



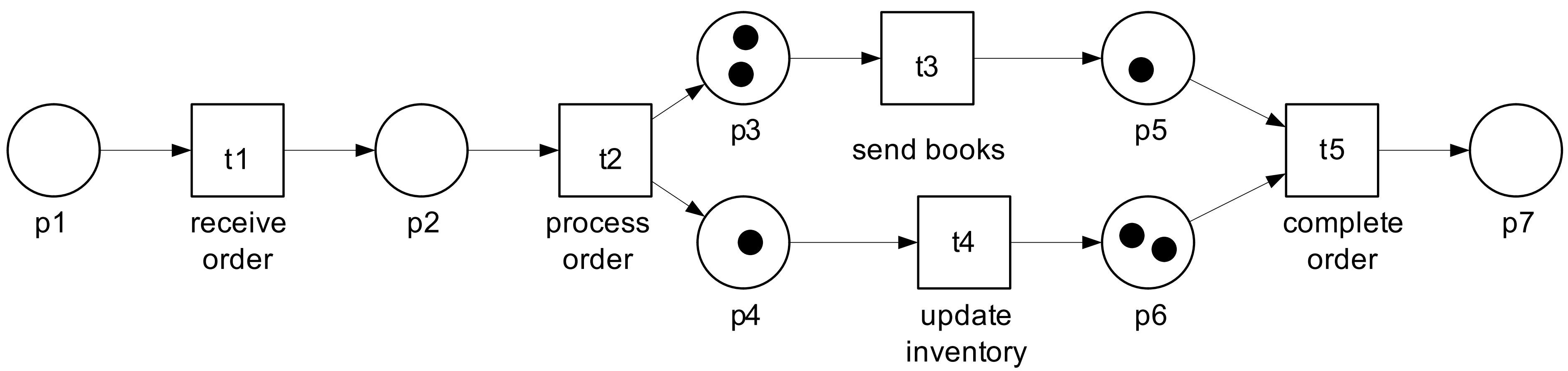
$$P = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\}$$
$$T = \{t_1, t_2, t_3, t_4, t_5\}$$
$$F = \{(p_1, t_1), (t_1, p_2), \ldots ?\}$$
$$M_0 = 2p_3 + p_4 + p_5 + 2p_6$$

Pre-set and post-set

Formally:

$$\bullet x = \{\ y \ \mid\ (y,x) \in F\ \} \quad \text{pre-set}$$
$$x\bullet = \{\ y \ \mid\ (x,y) \in F\ \} \quad \text{post-set}$$

Question time



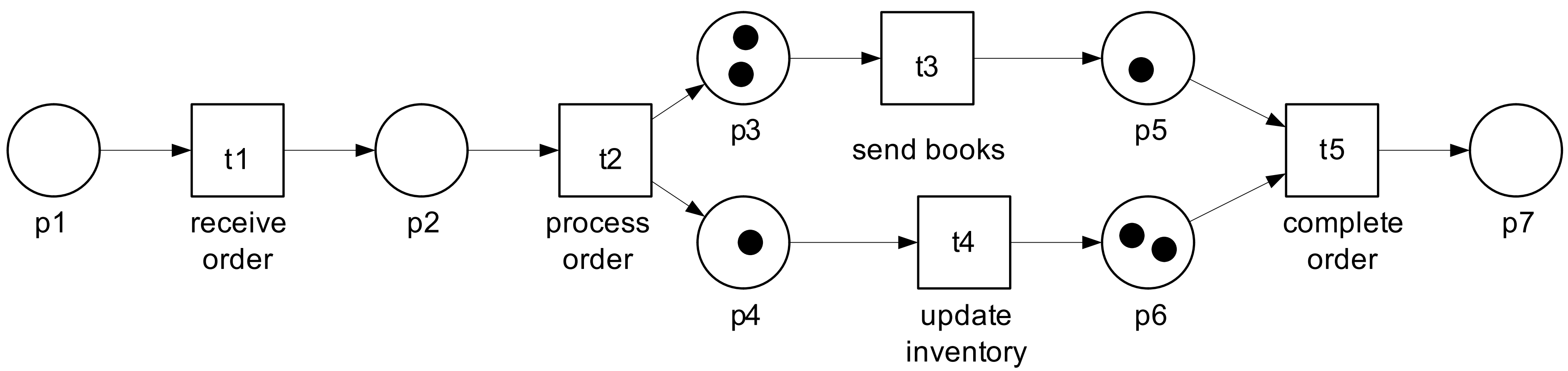
$$\bullet t_1 = ?$$
$$\bullet t_2 = ?$$
$$\bullet t_3 = ?$$
$$\bullet t_4 = ?$$
$$\bullet t_5 = ?$$

$$t_1\bullet = ?$$
$$t_2\bullet = ?$$
$$t_3\bullet = ?$$
$$t_4\bullet = ?$$
$$t_5\bullet = ?$$

$$\bullet p_1 = ?$$
$$\bullet p_2 = ?$$
$$\bullet p_3 = ?$$
$$\bullet p_4 = ?$$
$$\bullet p_5 = ?$$
$$\bullet p_6 = ?$$
$$\bullet p_7 = ?$$

$$p_1\bullet = ?$$
$$p_2\bullet = ?$$
$$p_3\bullet = ?$$
$$p_4\bullet = ?$$
$$p_5\bullet = ?$$
$$p_6\bullet = ?$$
$$p_7\bullet = ?$$

Exercises



$$\bullet p_1 = ?\,\emptyset \qquad p_1\bullet = ?\{\, t_1 \,\}$$

$$\bullet p_2 = ?\{\, t_1 \,\} \qquad p_2\bullet = ?\{\, t_2 \,\}$$

$$\bullet t_1 = ?\{\, p_1 \,\} \qquad t_1\bullet = ?\{\, p_2 \,\}$$

$$\bullet t_2 = ?\{\, p_2 \,\} \qquad t_2\bullet = ?\{\, p_3, p_4 \,\}$$

$$\bullet t_5 = ?\{\, p_5, p_6 \,\} \quad t_5\bullet = ?\{\, p_7 \,\}$$

Enabling $M[t\rangle$

(a set can be seen as a multiset whose elements have multiplicity 1)

A transition t is **enabled** at marking M iff $\bullet t \subseteq M$
and we write $M \xrightarrow{t}$ (also $M\,[t\rangle$)

A transition is enabled if each of its input places contains at least one token

Firing $M[t\rangle M'$

A transition t that is enabled at M can **fire**.
The **firing** of t at M changes the state to

$$M' = M - \bullet t + t\bullet$$

and we write $M \xrightarrow{t} M'$ (also $M\,[t\rangle\,M'$)

When a transition fires
it consumes a token from each input place
it produces a token into each output place

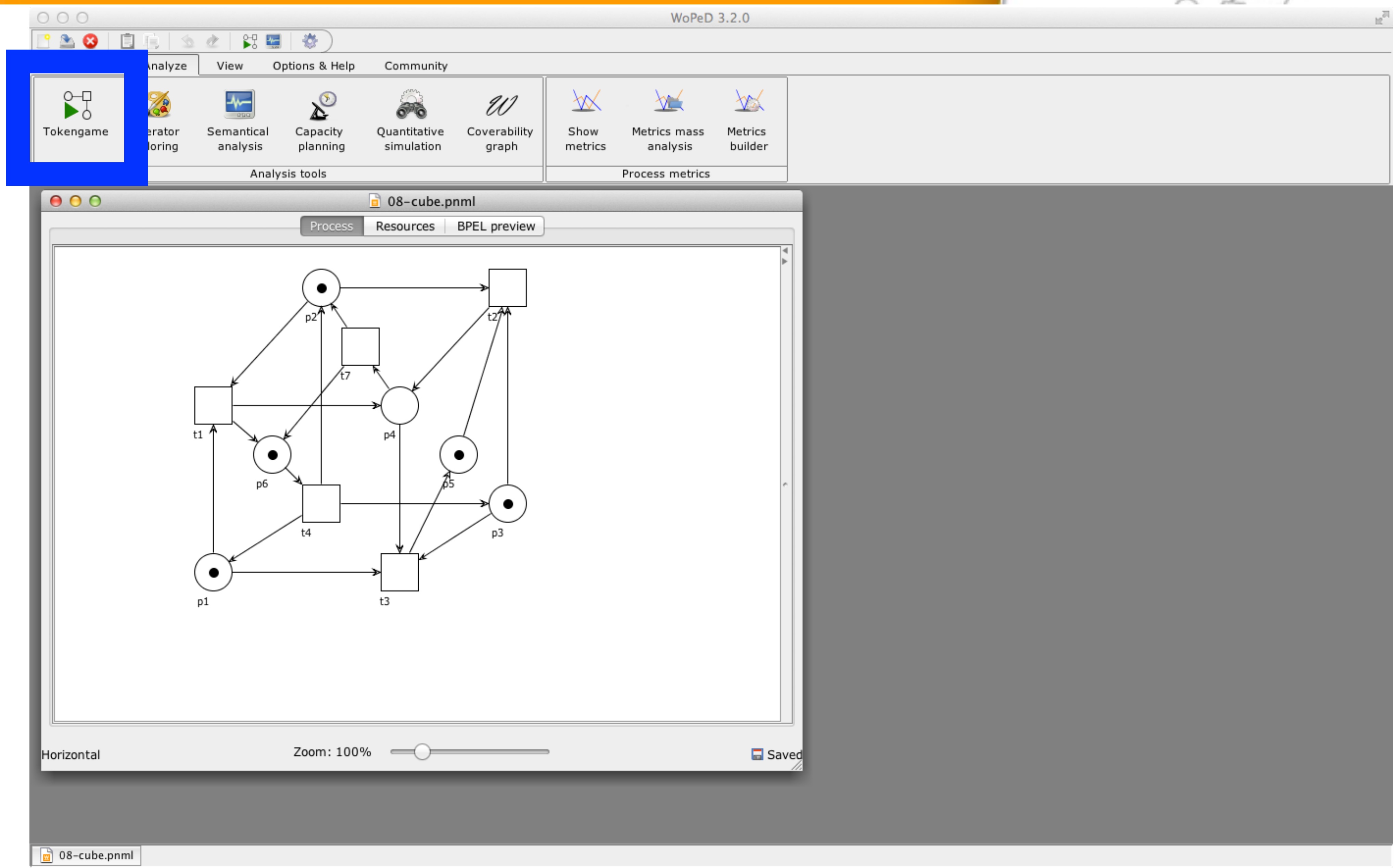
Some remarks

Firing is an atomic action

Our semantics is interleaving:
multiple transitions may be enabled,
but only one fires at a time

The network is static, but
the overall number of tokens may vary over time
(if transitions are fired for which the number of input places is not equal to the number of output places)


http://woped.dhbw-karlsruhe.de/woped/
WoPeD
Workflow Petri Net Designer
Download WoPeD at sourceforge!
W
WoPeD


WoPeD 3.2.0
nalyze
View
Options & Help
Community
Tokengame
erator
loring
Semantical analysis
Capacity planning
Quantitative simulation
Coverability graph
Show metrics
Metrics mass analysis
Metrics builder
Analysis tools
Process metrics
08-cube.pnml
Process
Resources
BPEL preview
p1
p2
p3
p4
p5
p6
t1
t2
t3
t4
t7
Horizontal
Zoom: 100%
Saved
08-cube.pnml

Notation

We write $M \rightarrow$ if $M \xrightarrow{t}$ for some transition t

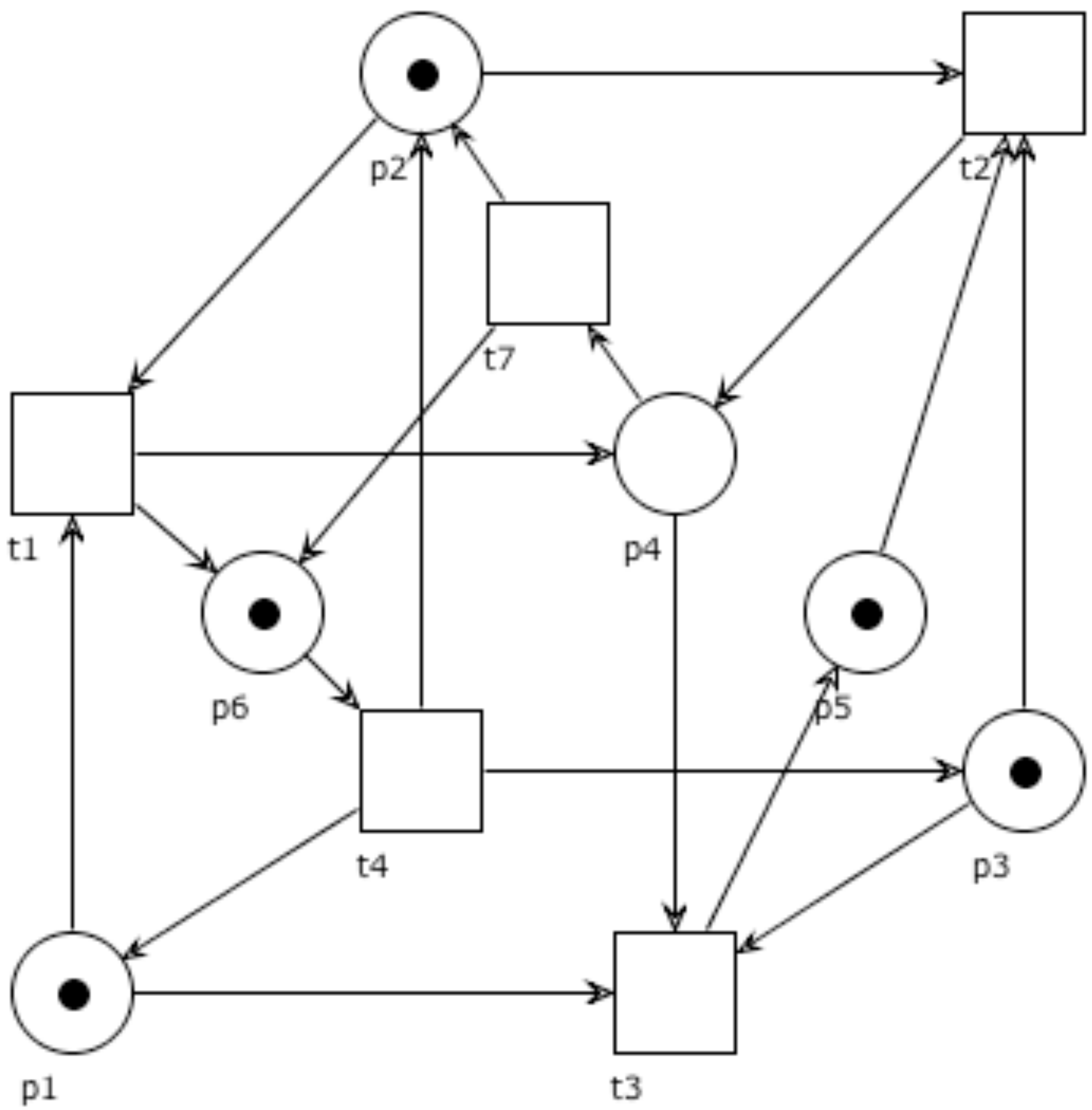
We write $M \rightarrow M'$ if $M \xrightarrow{t} M'$ for some transition t

We write $M \stackrel{t}{\not\rightarrow}$ if transition t is not enabled at M

We write $M \not\rightarrow$ if no transition is enabled at M

Example

$$M_0 = p_1 + p_2 + p_3 + p_5 + p_6$$



We can write that

- $M_0 \longrightarrow$ (there are enabled transitions, e.g., t1)
- $M_0 \longrightarrow p_1 + p_4 + p_6$ (by firing t2)
- $M_0 \overset{t_7}{\not\longrightarrow}$
- $p_1 + p_5 \not\longrightarrow$

Firing sequence

Let $\sigma = t_1 t_2 ... t_{n-1} \in T^*$ be a sequence of transitions.

We write $M \overset{\sigma}{\to} M'$ (and $M \overset{\sigma}{\to}$) if:

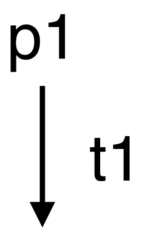
there is a sequence of markings $M_1, ..., M_n$

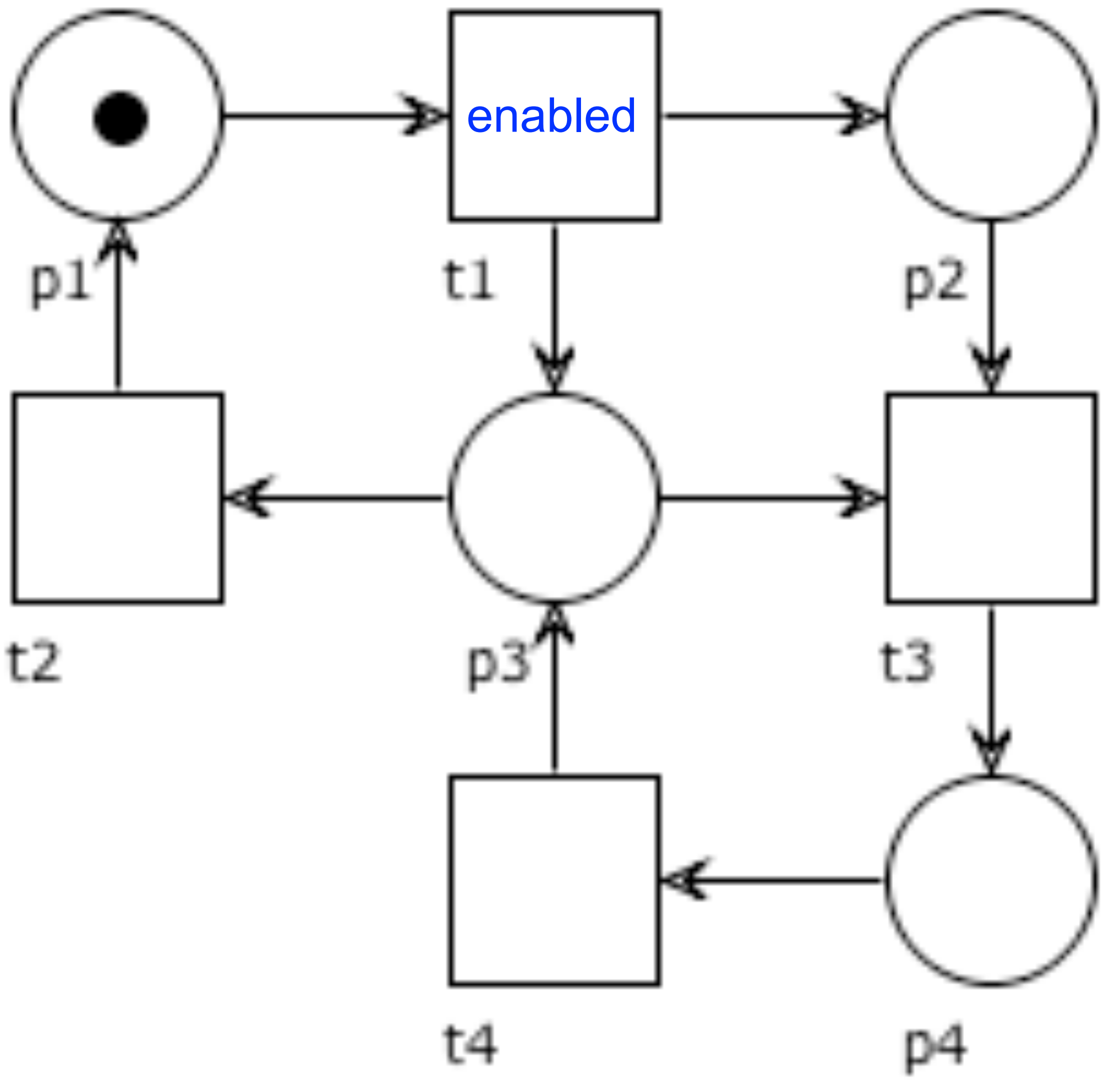
with $M = M_1$ and $M' = M_n$

and $M_i \xrightarrow{t_i} M_{i+1}$ for $1 \leq i < n$

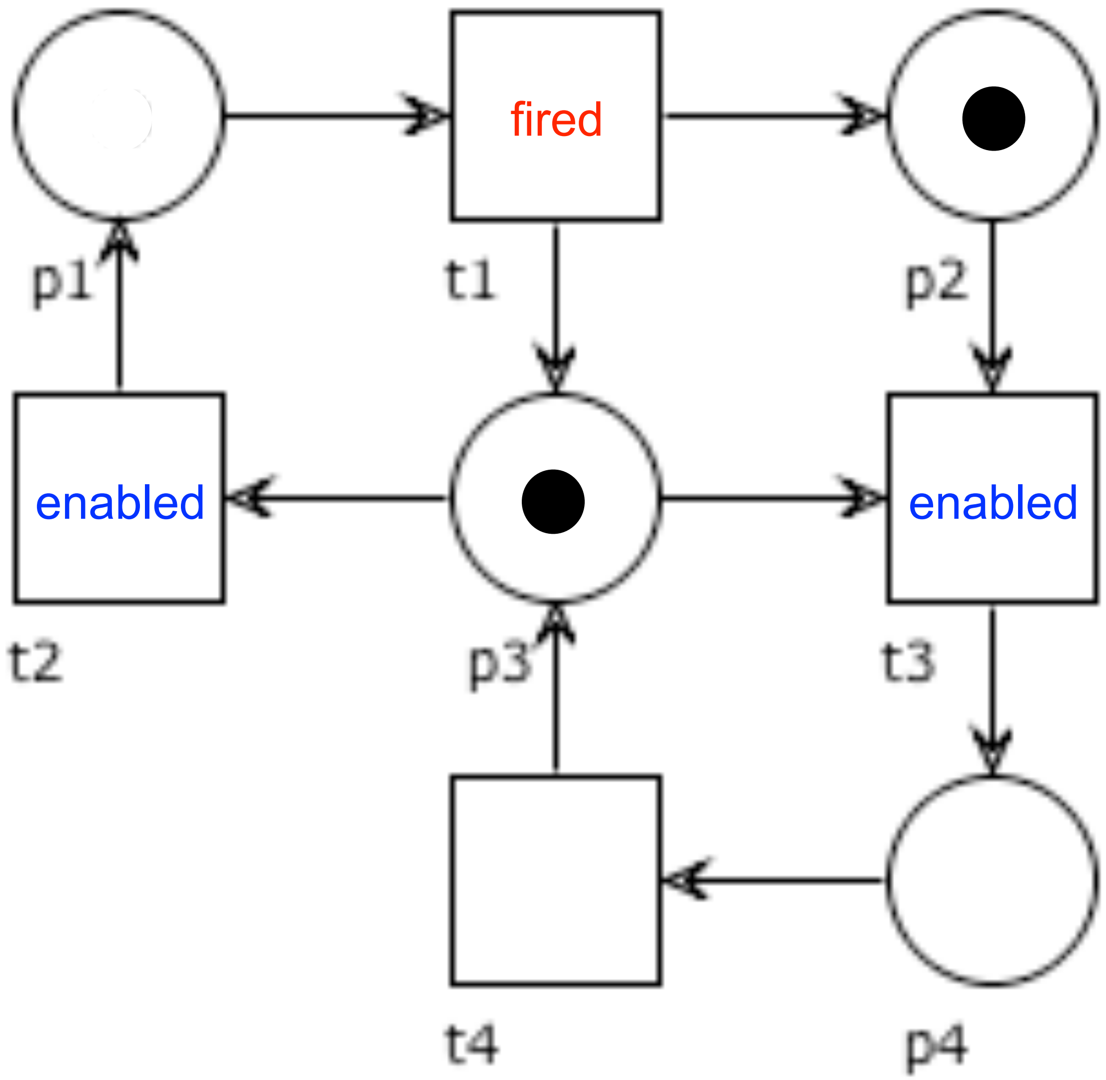
(i.e. $M = M_1 \xrightarrow{t_1} M_2 \xrightarrow{t_2} ... \xrightarrow{t_{n-1}} M_n = M'$)

Example



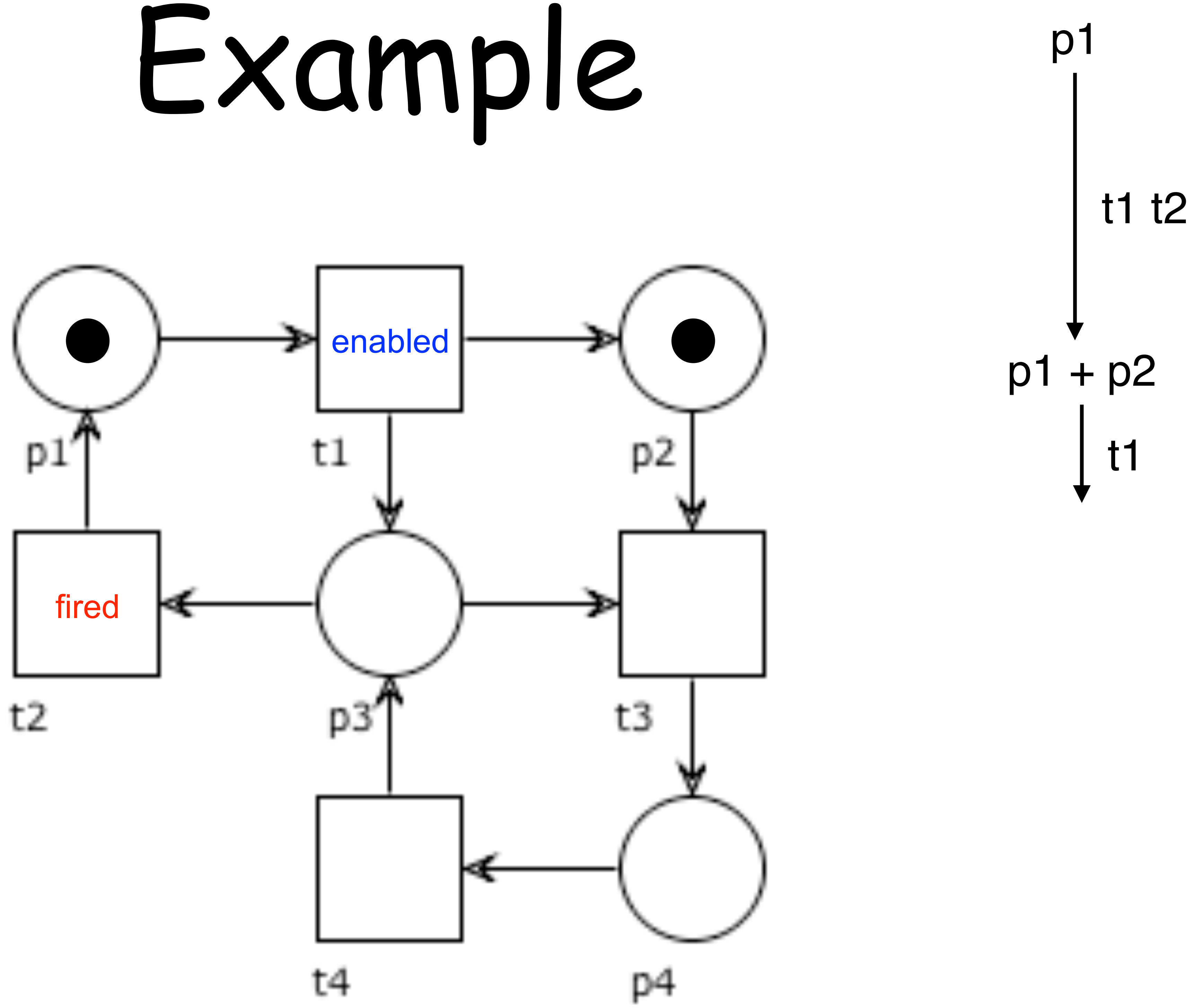


Example

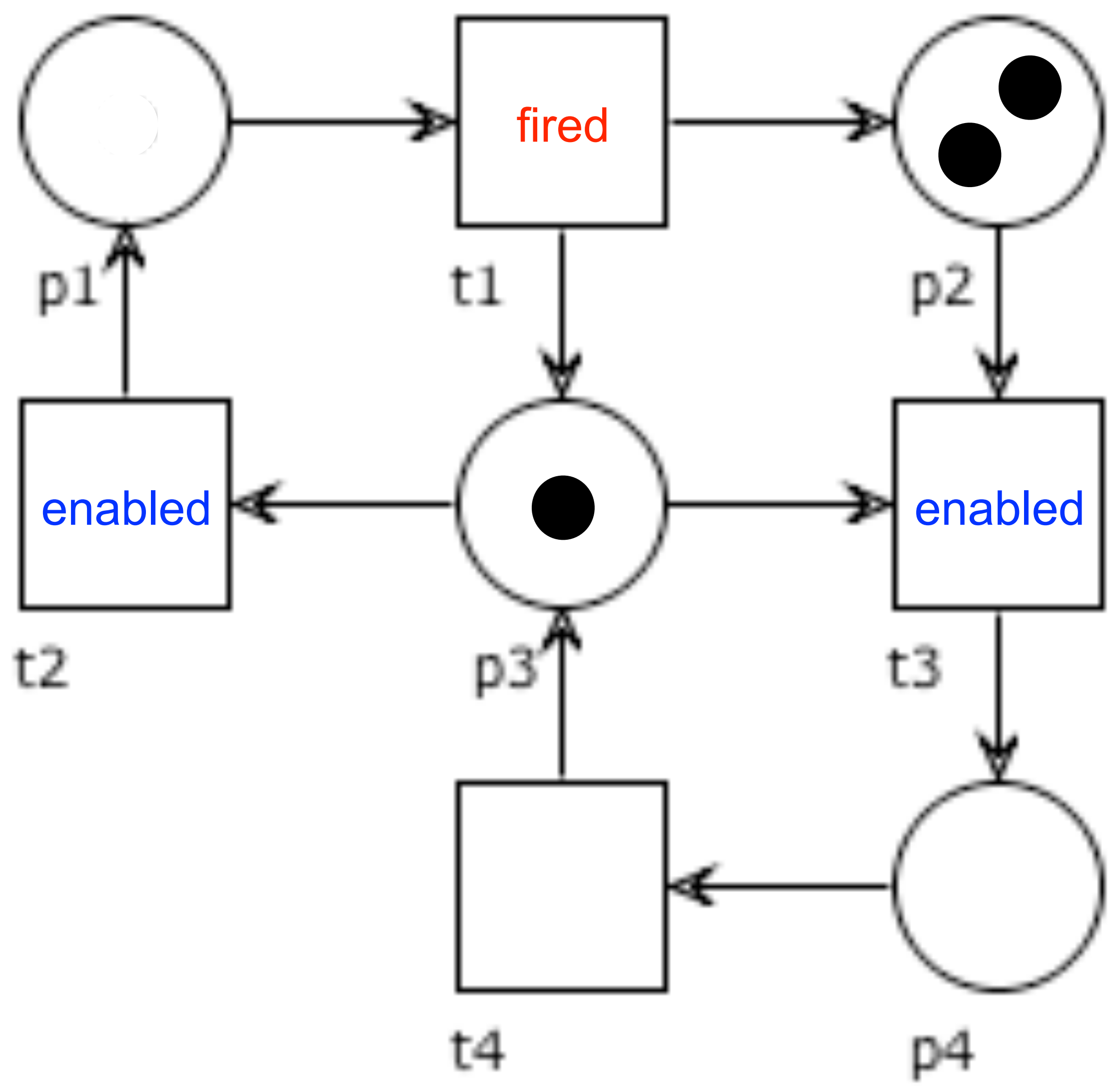


$$p1 \xrightarrow{t1} p2 + p3 \xrightarrow{t2\ /\ t3}$$

Example



Example



p1

$\downarrow$ t1 t2 t1

2p2 + p3

$\downarrow$ t2 / t3

Reachable markings $[M\rangle$

We write $M \stackrel{*}{\to} M'$ if $M \stackrel{\sigma}{\to} M'$ for some $\sigma \in T^*$

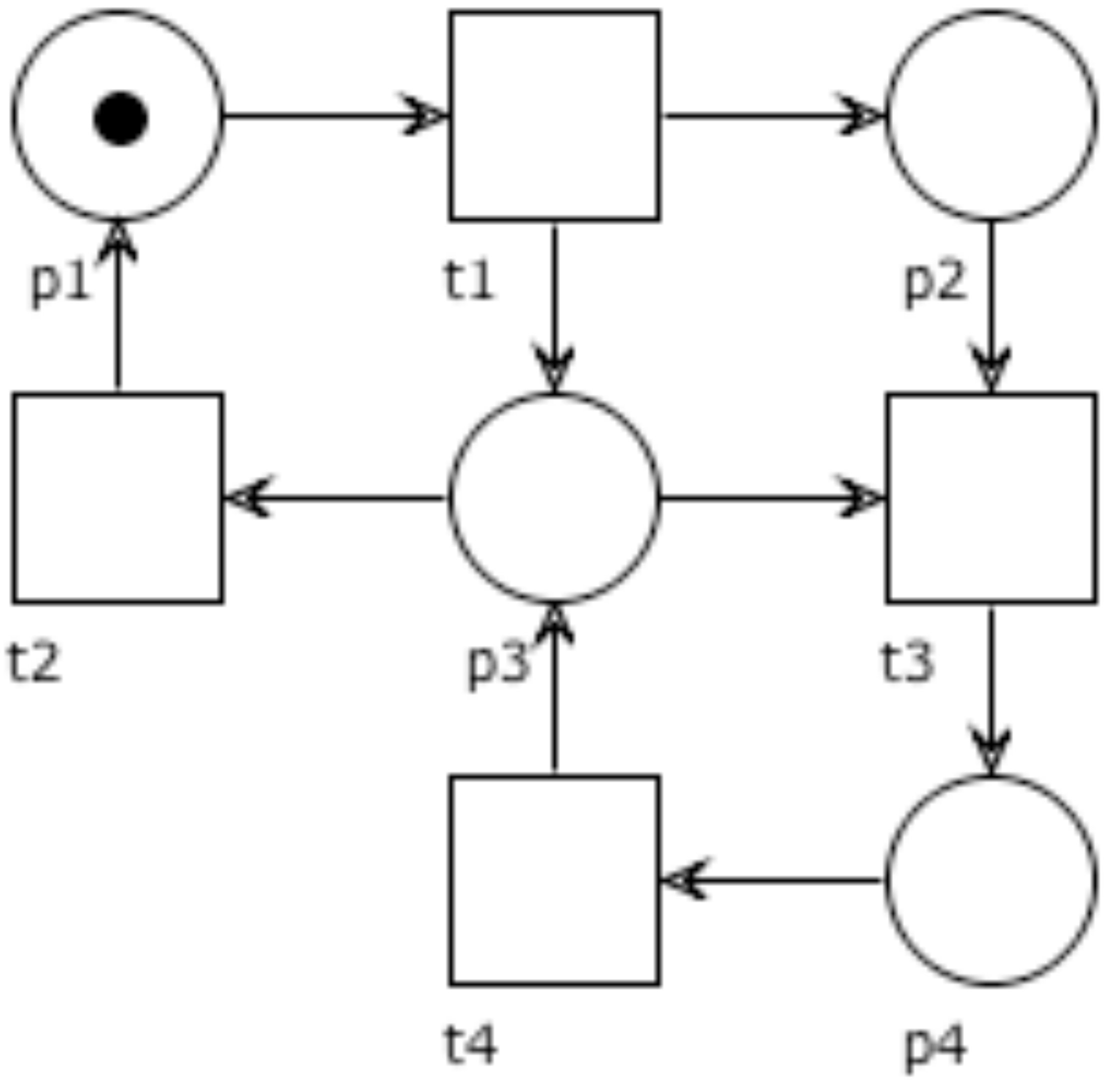
A marking M' is **reachable from** M if $M \stackrel{*}{\to} M'$

Note that $M \stackrel{\epsilon}{\to} M$ for ϵ the empty sequence

The set of markings reachable from M is often denoted:

$$reach(M) \text{ or also } [M\rangle$$

Example



Which of the following holds true?

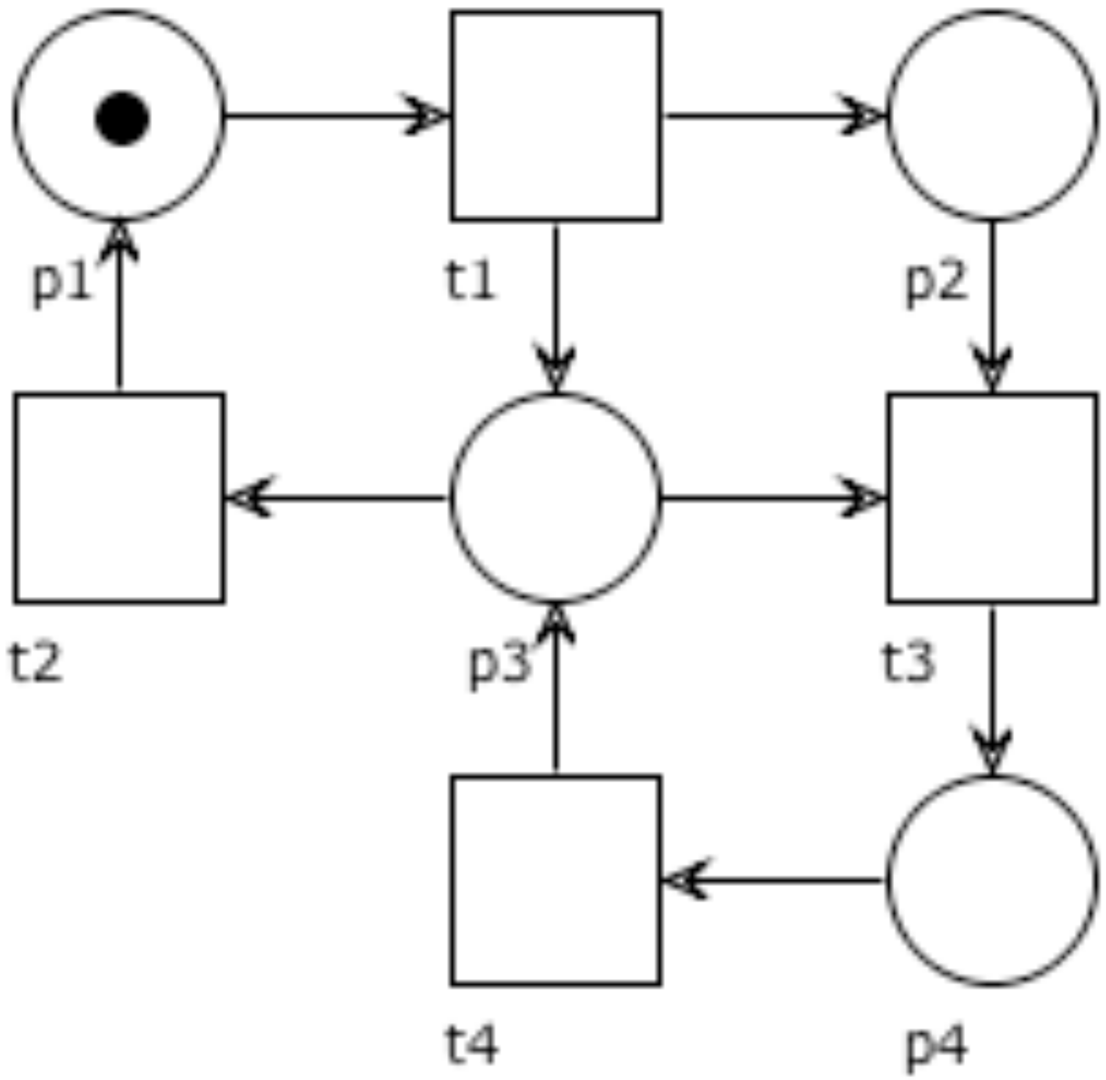
$$p_1 \xrightarrow{t_1\ t_2\ t_1\ t_3}$$

$$p_1 \xrightarrow{t_1\ t_3\ t_4\ t_2}$$

$$p_1 \xrightarrow{t_1\ t_3\ t_4\ t_3}$$

$$p_1 \xrightarrow{t_1\ t_3\ t_3}$$

Example



Which of the following holds true?

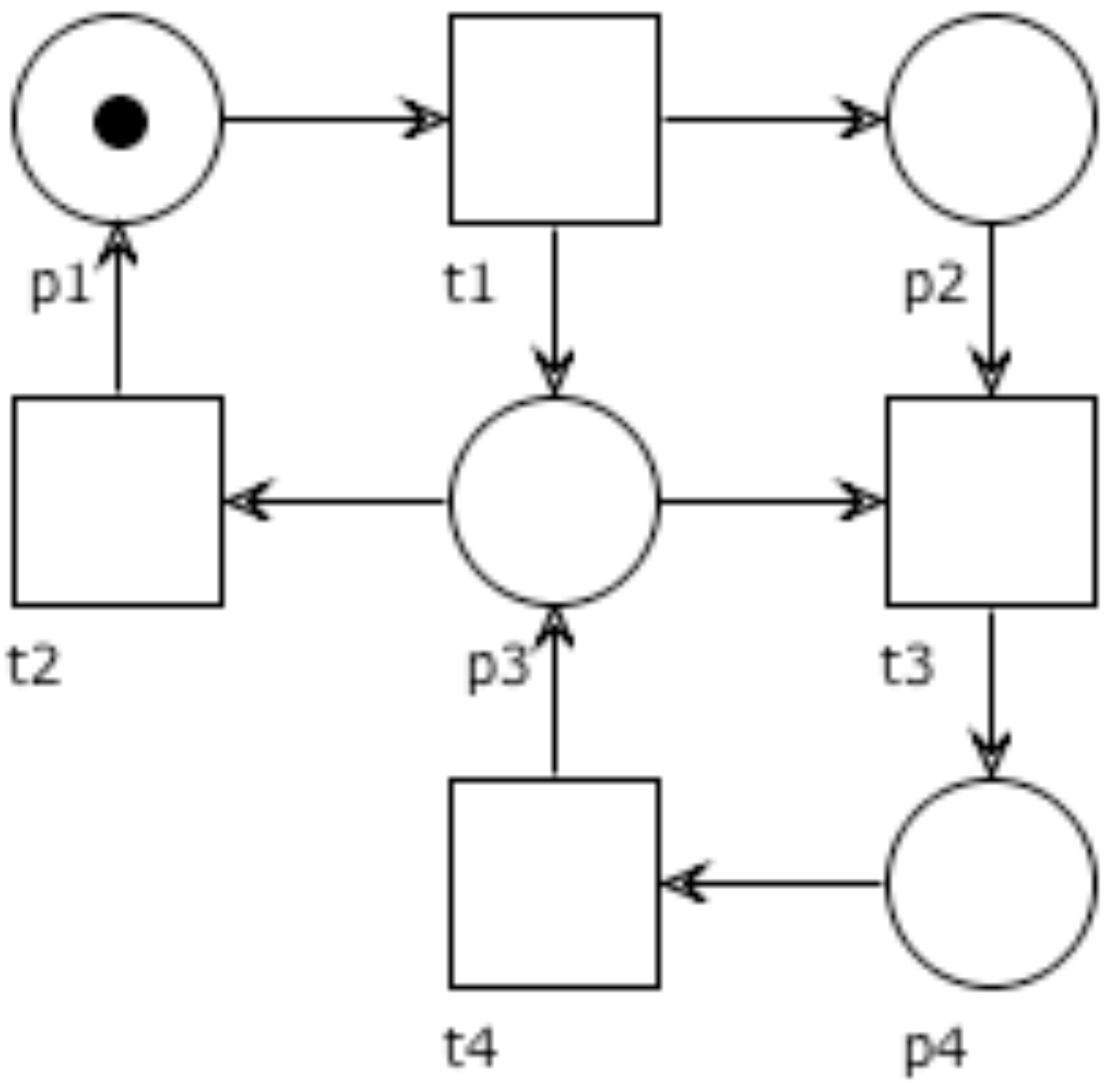
$p_1 \xrightarrow{t_1\ t_2\ t_1\ t_3}$ Yes

$p_1 \xrightarrow{t_1\ t_3\ t_4\ t_2}$ Yes

$p_1 \xrightarrow{t_1\ t_3\ t_4\ t_3}$ No (t_3 not enabled)

$p_1 \xrightarrow{t_1\ t_3\ t_3}$ No (t_3 not enabled)

Example



We have that:

$$p_1 \xrightarrow{t_1\ t_2\ t_1\ t_3} p_4$$

$$p_1 \xrightarrow{t_1\ t_3\ t_4\ t_2} p_1$$

Infinite sequence

Let $\sigma = t_1 t_2 ... \in T^\omega$ be an infinite sequence of transitions.

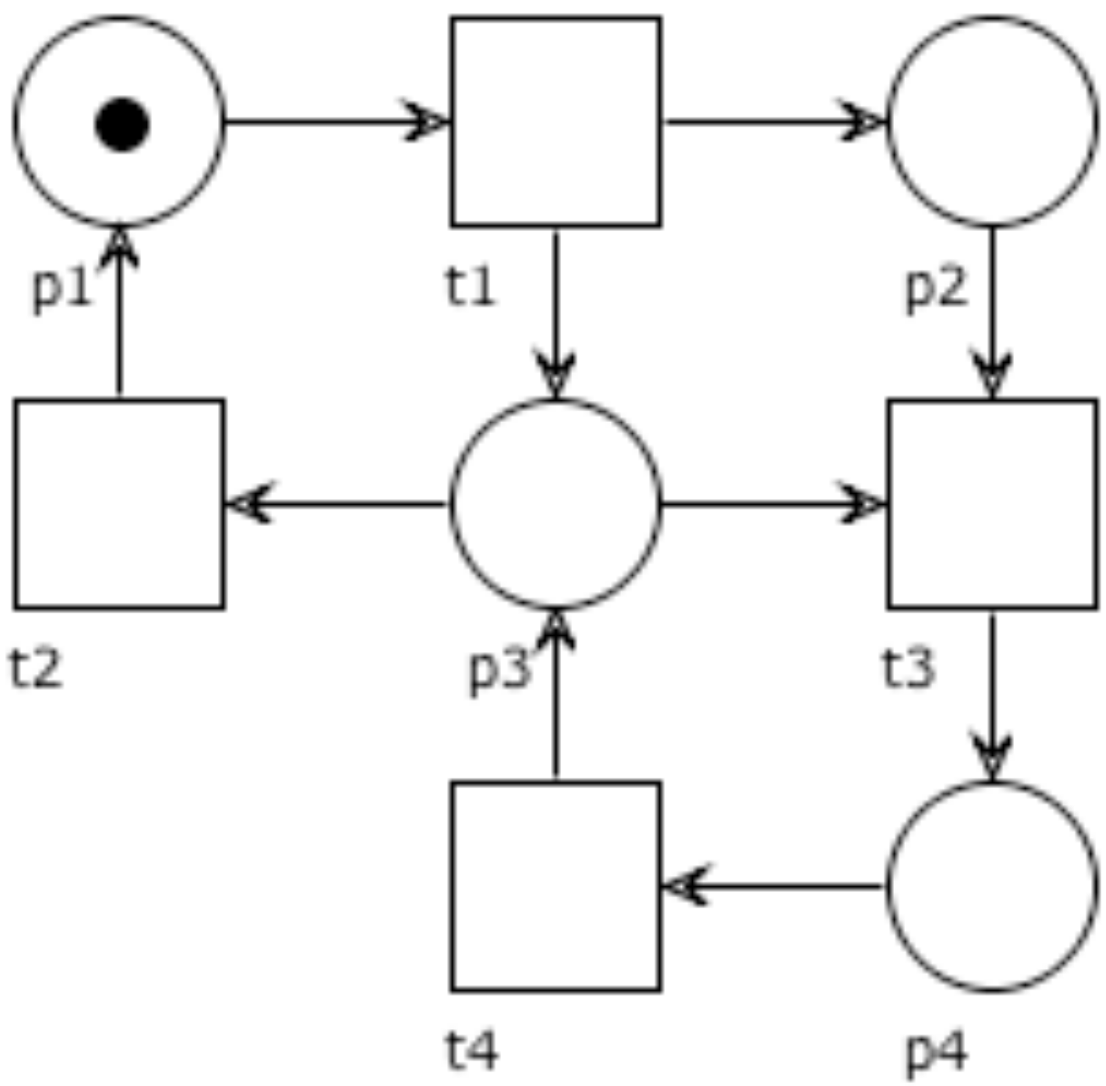
We write $M \stackrel{\sigma}{\to}$ if:

there is an infinite sequence of markings $M_1, M_2, ...$

with $M = M_1$ and $M_i \xrightarrow{t_i} M_{i+1}$ for $1 \leq i$

(i.e. $M = M_1 \xrightarrow{t_1} M_2 \xrightarrow{t_2} ...$)

Example



We have that:

$$p_1 \xrightarrow{t_1\ t_3\ t_4\ t_2\ t_1\ t_3\ t_4\ t_2\ t_1\ t_3\ t_4\ t_2 \cdots}$$

Concatenation & prefix

Concatenation: given $\sigma_1 = a_1 \cdots a_n$ $\sigma_2 = b_1 \cdots b_m$

$\sigma_1\sigma_2 = a_1 \cdots a_n b_1 \cdots b_m$ finite + finite = finite

Concatenation: given $\sigma_1 = a_1 \cdots a_n$ $\sigma_2 = b_1 b_2 \cdots$

$\sigma_1\sigma_2 = a_1 \cdots a_n b_1 b_2 \cdots$ finite + infinite = infinite

Prefix: finite

σ_1 is a proper prefix of σ if $\sigma_1\sigma_2 = \sigma$ for some $\sigma_2 \neq \varepsilon$

σ_1 is a prefix of σ if $\sigma_1 = \sigma$ or $\sigma_1\sigma_2 = \sigma$ for some $\sigma_2 \neq \varepsilon$

finite or infinite

Example: prefixes

$t_1t_4t_2t_3$	sequence	$t_1t_4t_7t_1t_4t_7t_1t_4t_7\cdots$
ϵ		ϵ
t_1	prefixes	t_1
t_1t_4		t_1t_4
$t_1t_4t_2$		$t_1t_4t_7$
$t_1t_4t_2t_3$		$t_1t_4t_7t_1$
		$t_1t_4t_7t_1t_4$
		$\cdots$
		$t_1t_4t_7t_1t_4t_7t_1t_4t_7\cdots$

Enabled sequence

We say that an occurrence sequence σ is **enabled** if $M \xrightarrow{\sigma}$

(σ can be finite or infinite)

Note that an infinite sequence can be represented as a map $\sigma : \mathbb{N} \to T$, where $\sigma(i) = t_i$

Enabledness

Proposition: $M \xrightarrow{\sigma}$ iff $M \xrightarrow{\sigma'}$ for every prefix σ' of σ

($\Rightarrow$) immediate from definition

($\Leftarrow$) trivial if σ is finite (σ itself is a prefix of σ)

When σ is infinite: taken any $i \in \mathbb{N}$ we need to prove that $t_i = \sigma(i)$ is enabled after the firing of the prefix $\sigma' = t_1 t_2 ... t_{i-1}$ of σ.

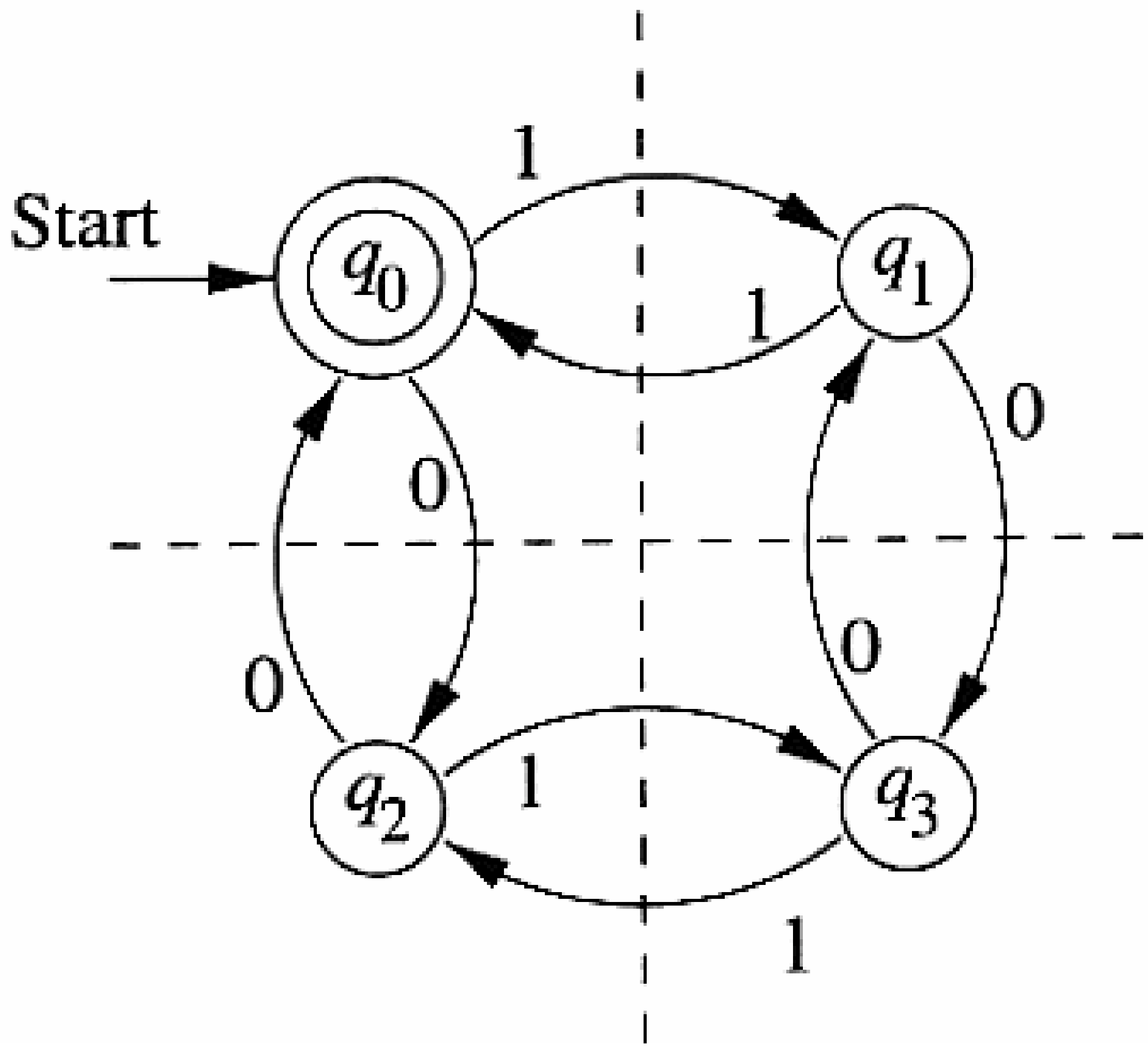
But this is obvious, because

$$M \xrightarrow{t_1} M_1 \xrightarrow{t_2} ... \xrightarrow{t_{i-1}} M_{i-1} \xrightarrow{t_i} M_i$$

is also a finite prefix of σ and therefore $M_{i-1} \xrightarrow{t_i}$

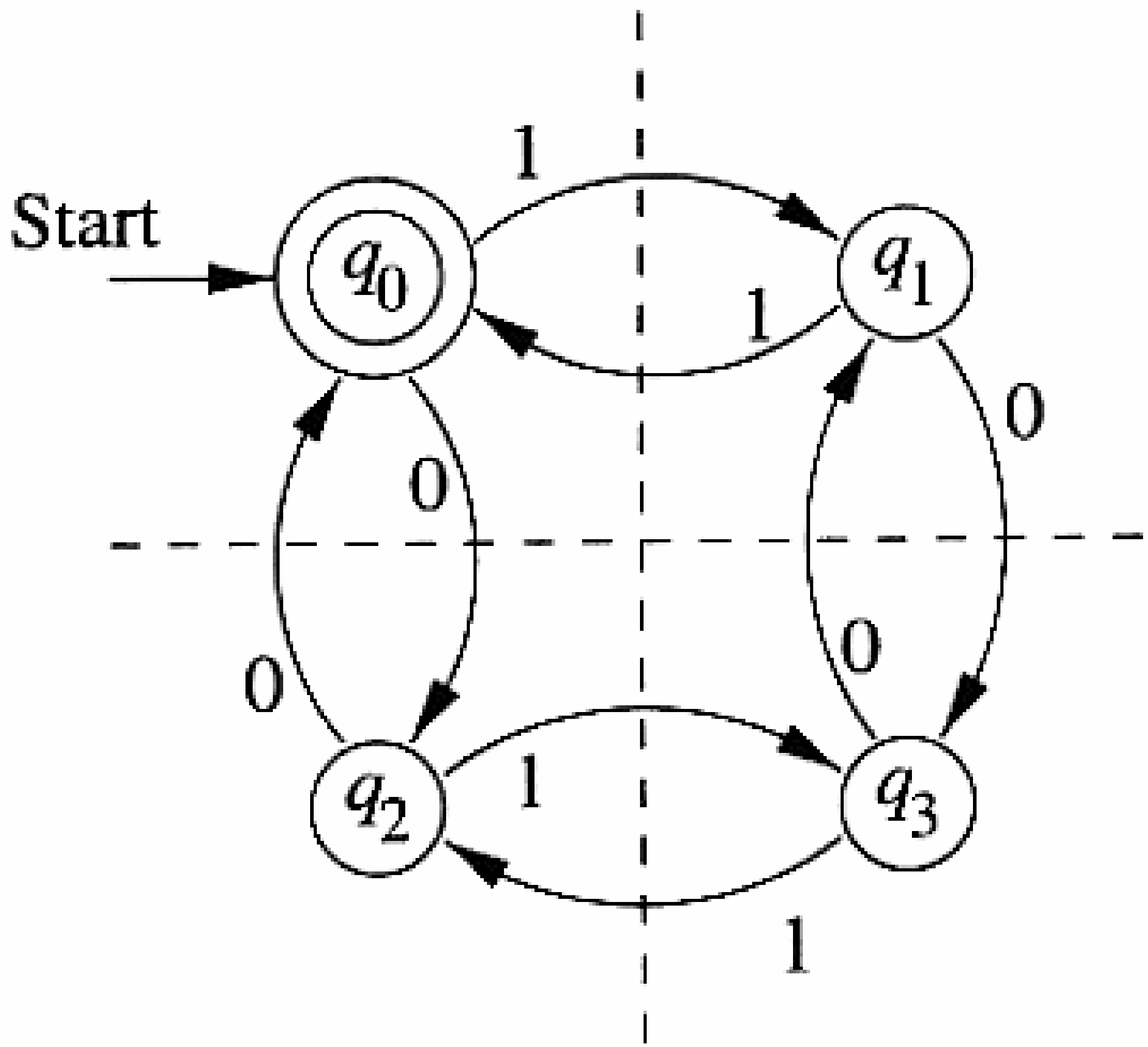
Exercises

DFA: example



	0	1
$\rightarrow *q_0$	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_1	q_2

DFA: exercise

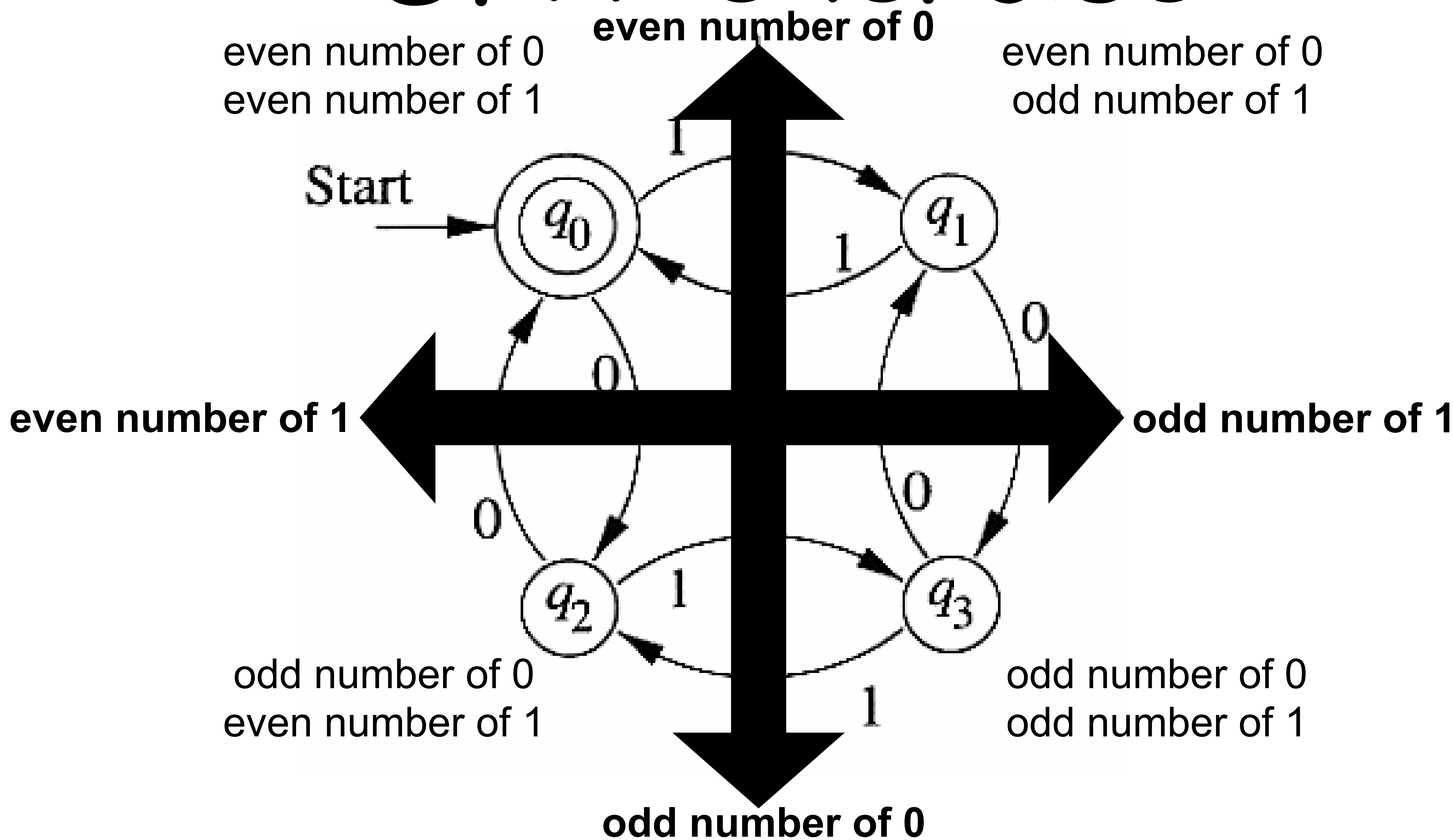


Does it accept 100 ? No

Does it accept 1010 ? Yes

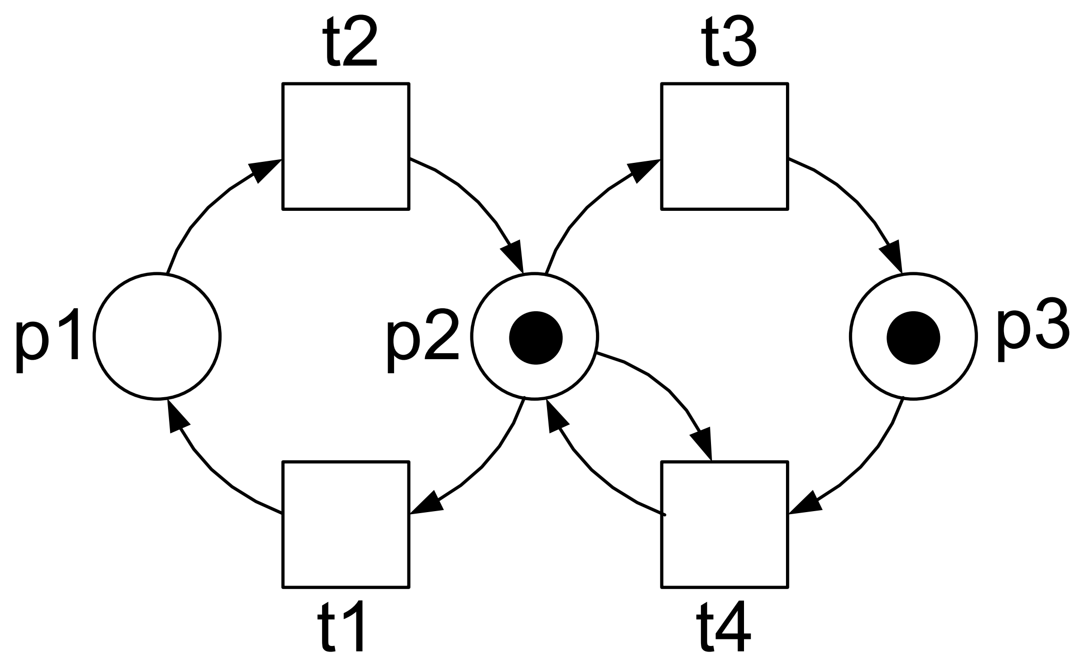
What is L(A) ?

DFA: exercise



What is L(A) ? **The set of all binary strings with an even number of occurrences of 0 and 1**

Exercises

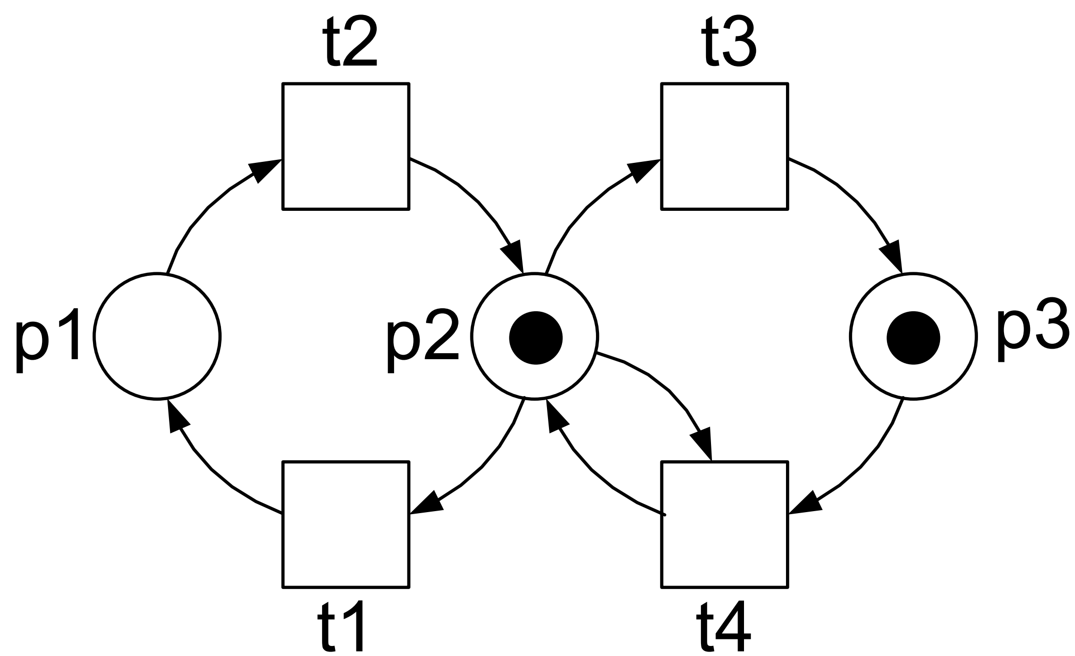


Determine the pre- and post-set of each element

Which transitions are currently enabled?
For each of them, which state would the firing lead to?

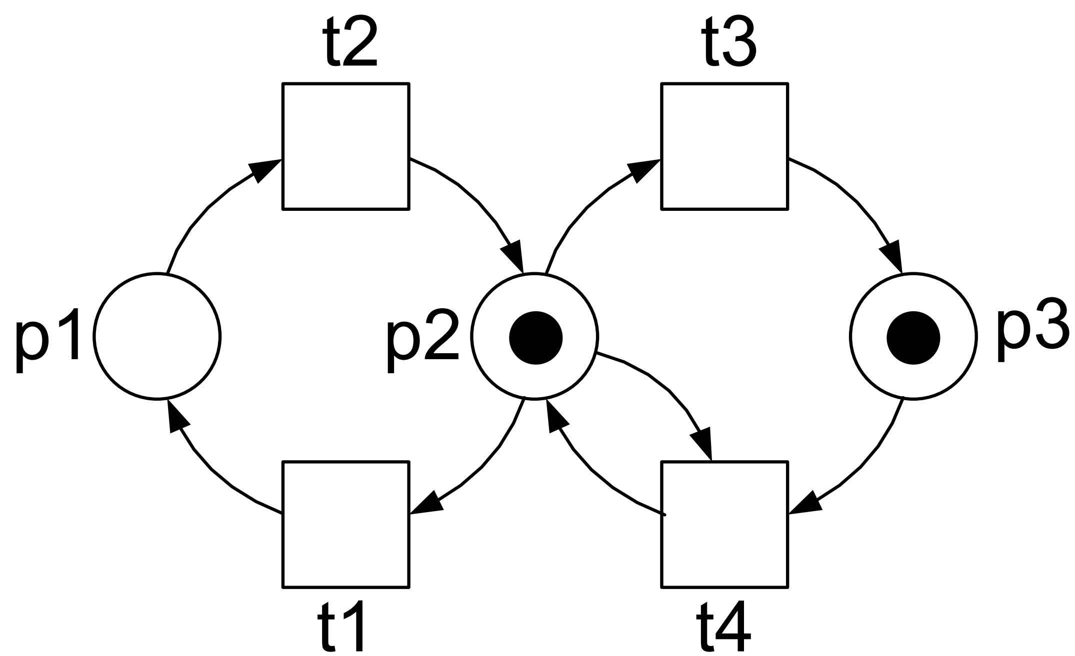
What are the reachable states?

Exercises



Determine the pre- and post-set of each element

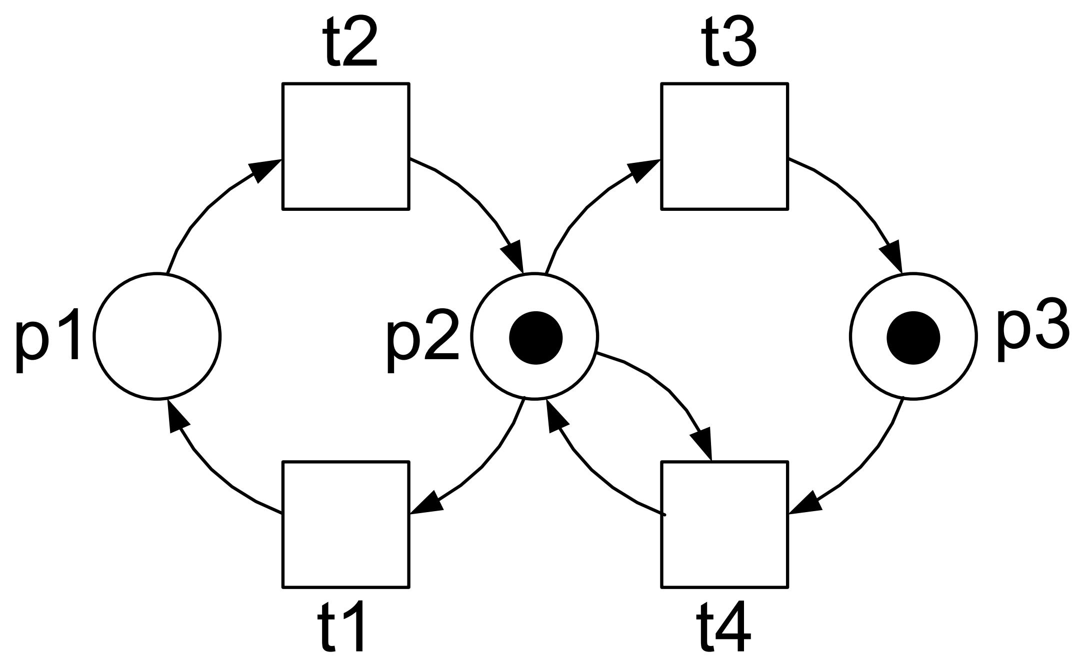
Exercises



Which transitions are currently enabled?

$$t_1, t_3, t_4$$

Exercises



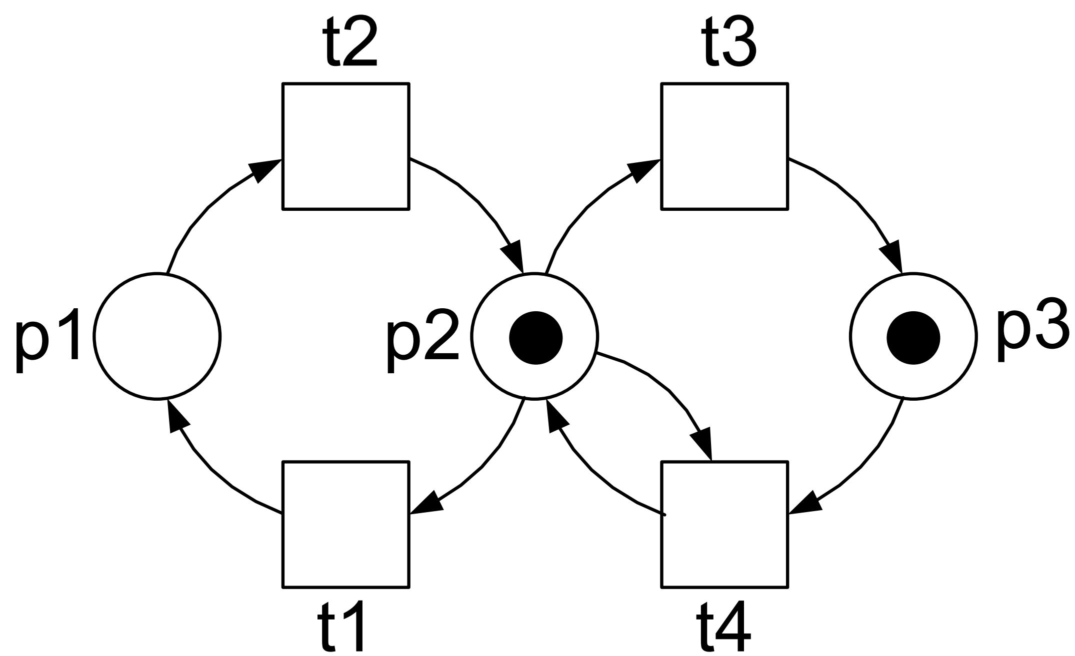
Which transitions are currently enabled?

$$t_1, t_3, t_4$$

For each of them, which state would its firing lead to?

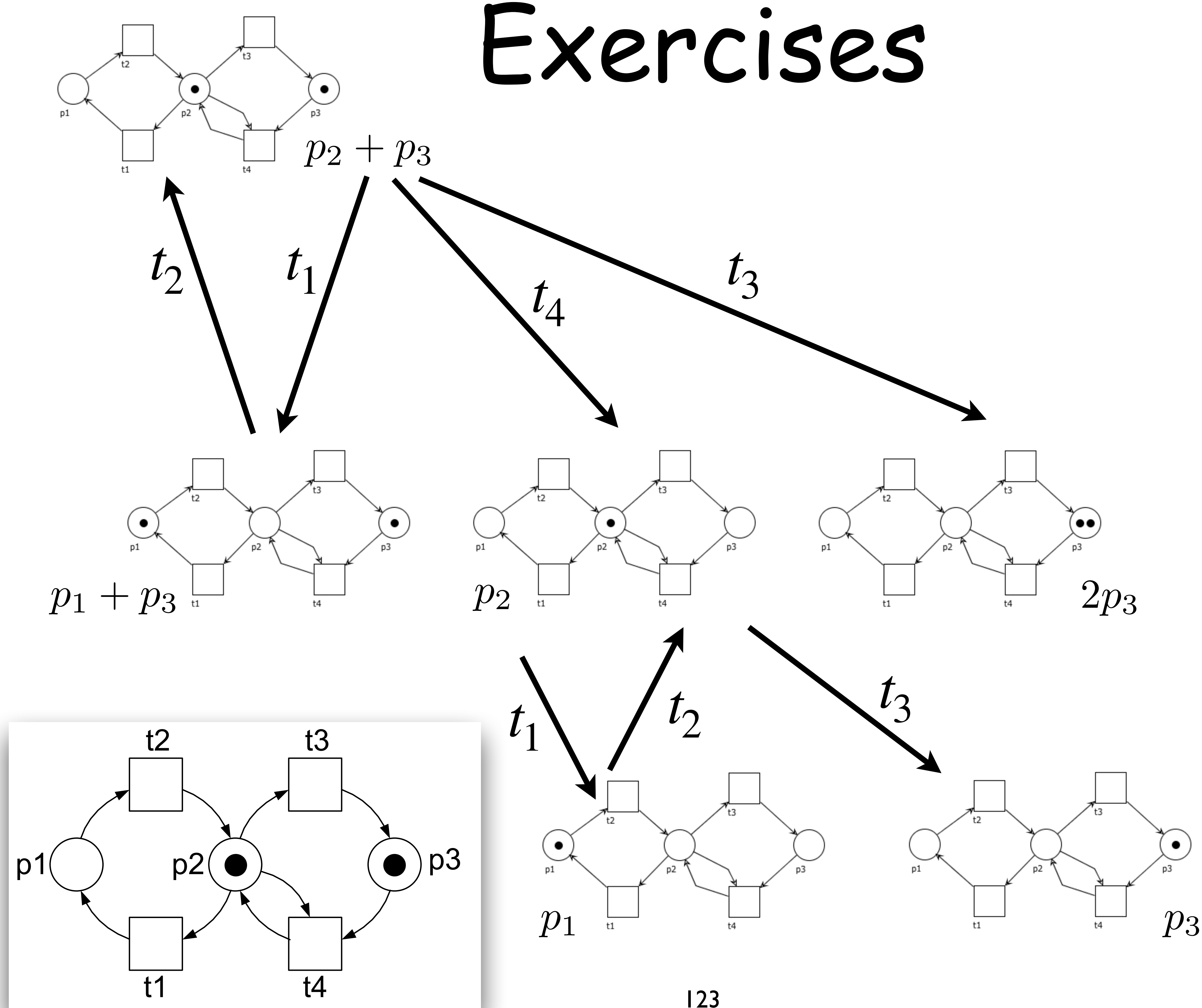
$$p_2 + p_3 \xrightarrow{t_1} p_1 + p_3 \qquad p_2 + p_3 \xrightarrow{t_3} 2p_3 \qquad p_2 + p_3 \xrightarrow{t_4} p_2$$

Exercises

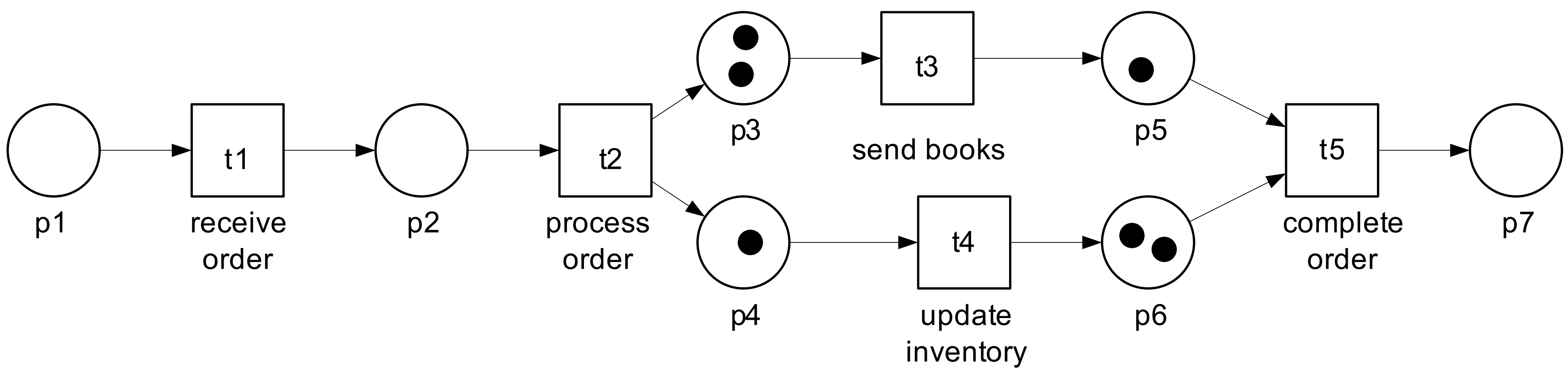


What are the reachable states?

Exercises



Exercises

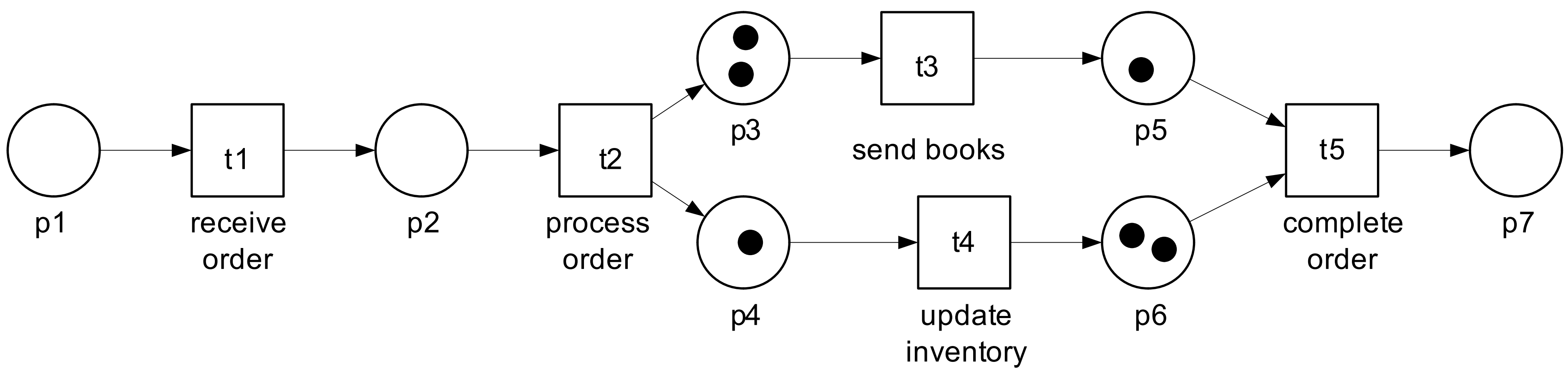


Which transitions are currently enabled?

For each of them, which state would the firing lead to?

What are the reachable states?

Exercises

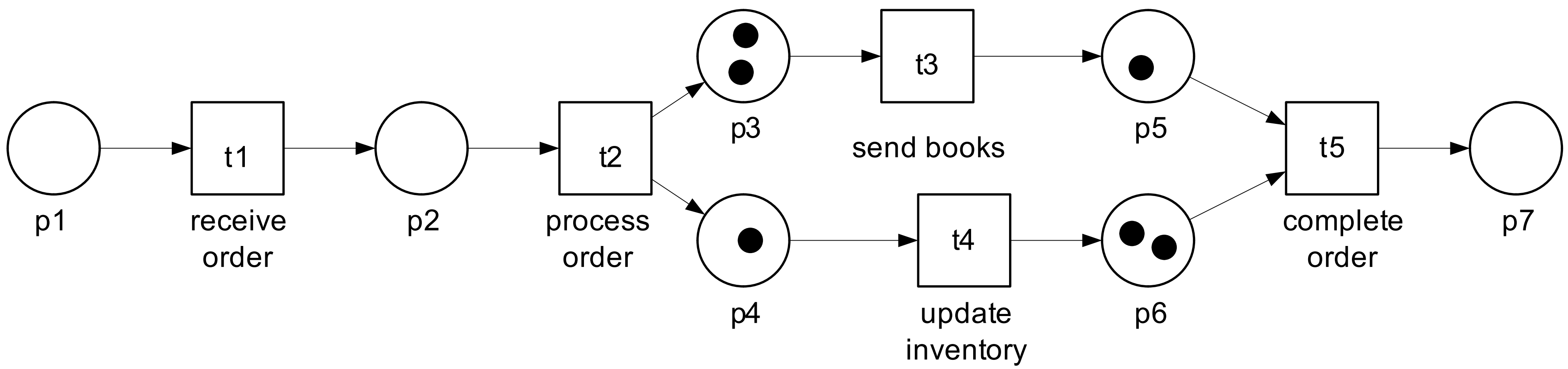


$$2p_3 + p_4 + p_5 + 2p_6$$

Which are the currently enabled transitions?

t_3, t_4, t_5

Exercises



$$2p_3 + p_4 + p_5 + 2p_6$$

Which are the currently enabled transitions?

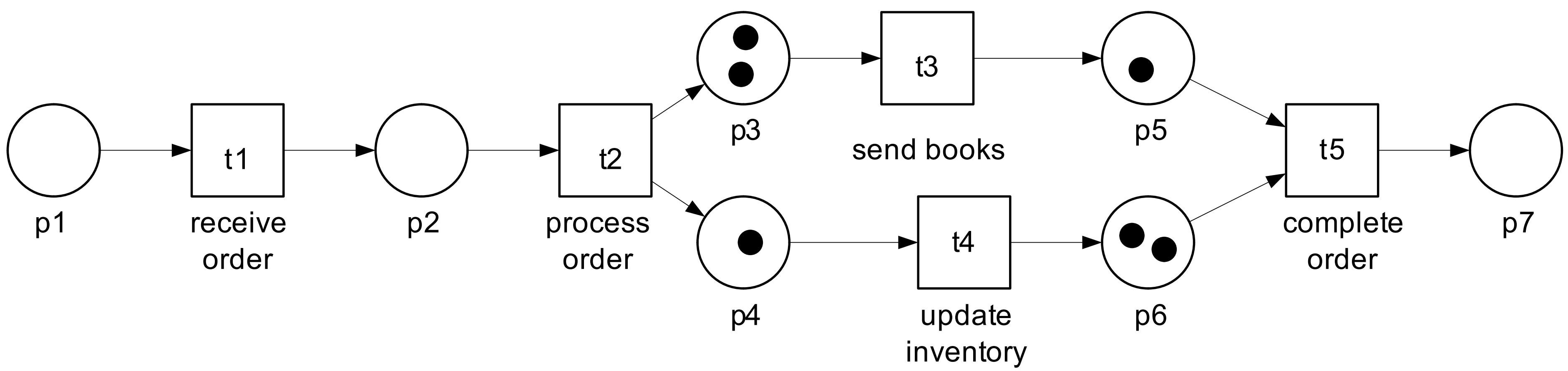
t_3 t_4 t_5

For each of them, which state would their firing lead to?

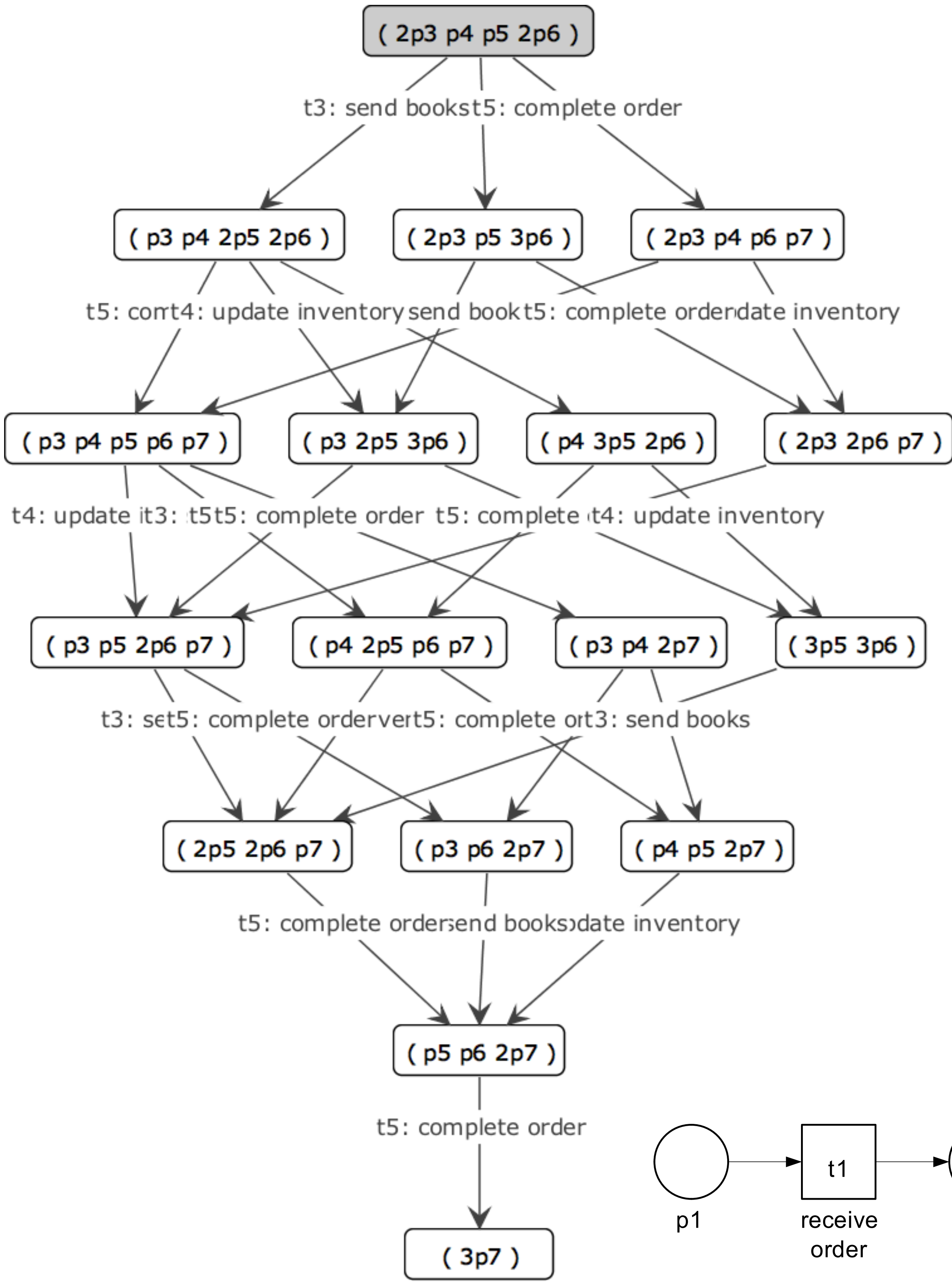
$p_3 + p_4 + 2p_5 + 2p_6$ $2p_3 + p_4 + p_6 + p_7$

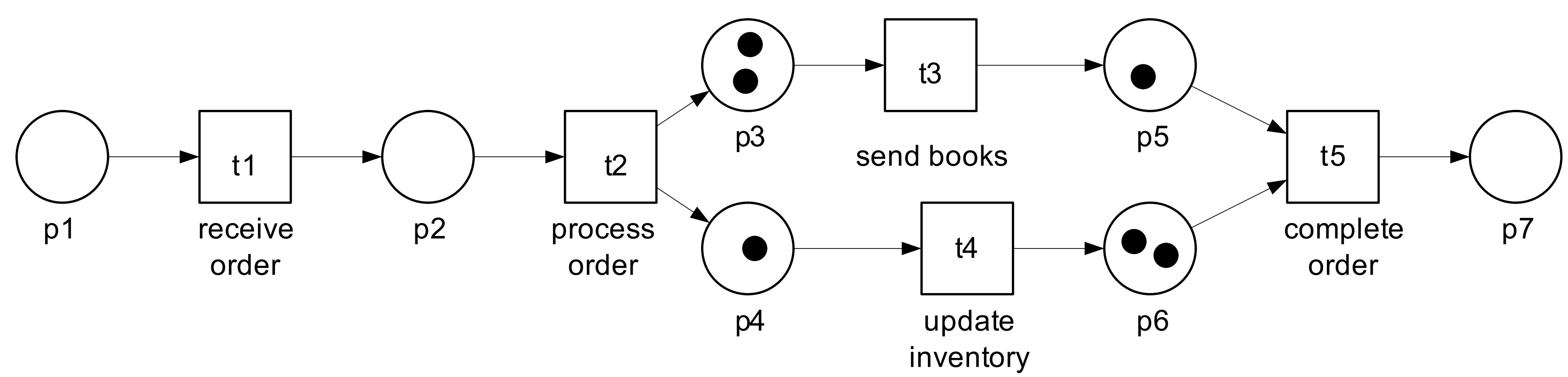
$$2p_3 + p_5 + 3p_6$$

Exercises



What are the reachable states?


(2p3 p4 p5 2p6)
t3: send books
t5: complete order
(p3 p4 2p5 2p6)
(2p3 p5 3p6)
(2p3 p4 p6 p7)
t4: update inventory
(p3 p4 p5 p6 p7)
(p3 2p5 3p6)
(p4 3p5 2p6)
(2p3 2p6 p7)
(p3 p5 2p6 p7)
(p4 2p5 p6 p7)
(p3 p4 2p7)
(3p5 3p6)
(2p5 2p6 p7)
(p3 p6 2p7)
(p4 p5 2p7)
(p5 p6 2p7)
(3p7)


p1
t1
receive order
p2
t2
process order
p3
t3
send books
p4
t4
update inventory
p5
p6
t5
complete order
p7