Grammatical inference: an introductory example

Module X3IT040, Colin de la Higuera, Nantes & Le Mans, 2020

UNIVERSITÉ DE NANTES

Statistical and symbolic language modeling

ATAL

LS2N LABORATOIRE DES SCIENCES DU NUMÉRIQUE DE NANTES

A synopsis

- There are many runs of algorithms one can take as first examples of what a grammatical inference algorithm is about.
- The following one is « fun » and contains many key ideas.

Bibliography

D. Carmel and S. Markovitch. Model-based learning of interaction strategies in multi-agent systems. *Journal of Experimental and Theoretical Artificial Intelligence*, 10(3):309–332, 1998

D. Carmel and S. Markovitch. Exploration strategies for model-based learning in multiagent systems. *Autonomous Agents and Multi-agent Systems*, 2(2):141–172, 1999

The problem:

- An agent must take cooperative decisions in a multi-agent world

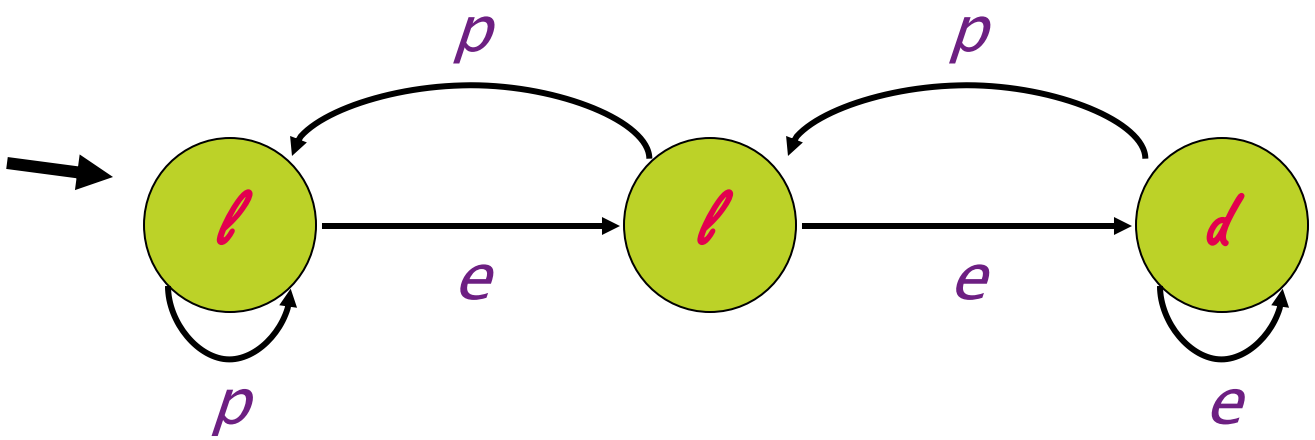
- His decisions will depend:
  - on what he hopes to win or lose
  - on the actions of other agents

Hypothesis:

The opponent follows a rational strategy (given by a *DFA* / Moore machine)

YOU:
listen or
doze

ME:
equations
or *pictures*



$e\ e\ p\ e\ p \rightarrow \ell$

$e\ e\ e \rightarrow d$

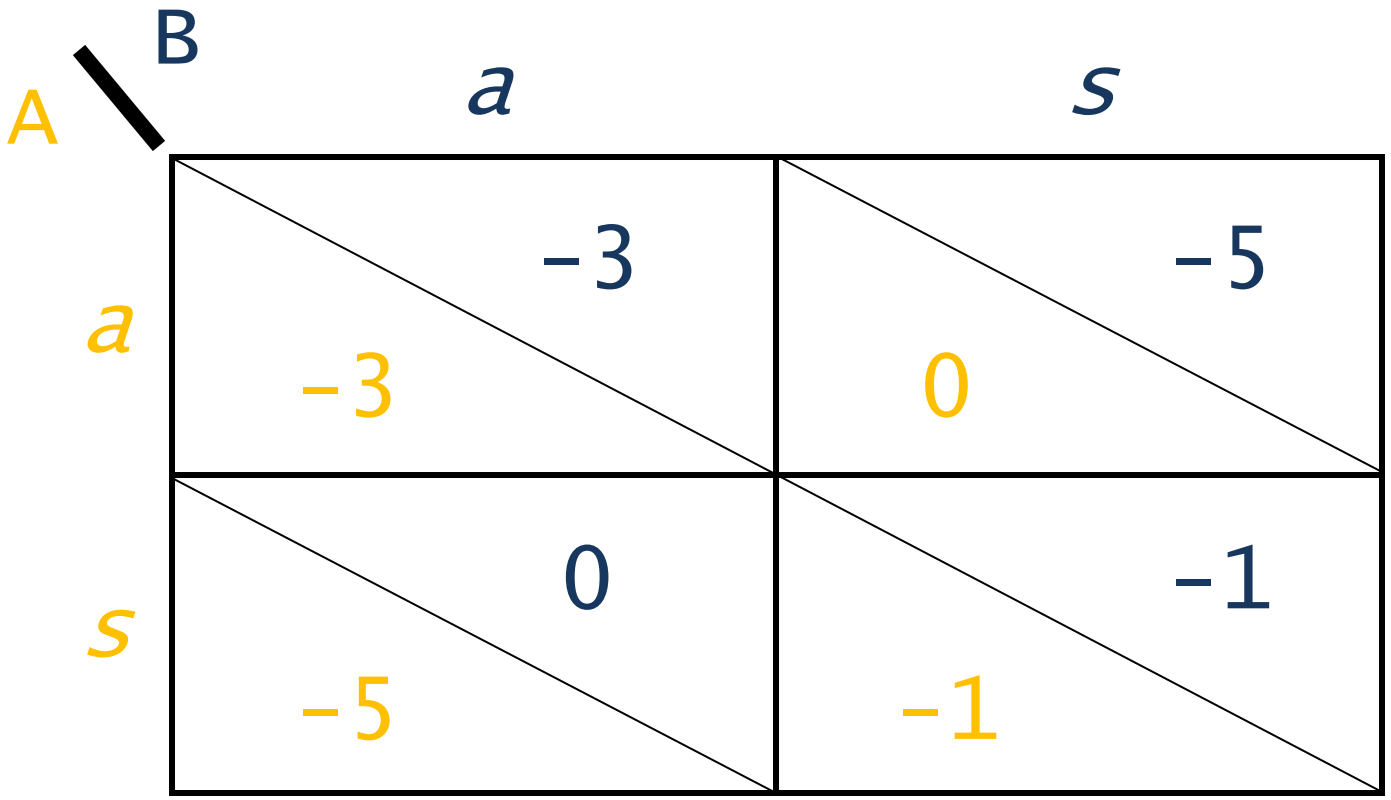
Example:

- Each prisoner can admit (*a*) or stay silent (*s*)

The prisoner's dilemma

  - If both admit: 3 years (prison) each
  - If A admits but not B: A=0 years, B=5 years
  - If B admits but not A: B=0 years, A=5 years
  - If neither admits: 1 year each

Example:



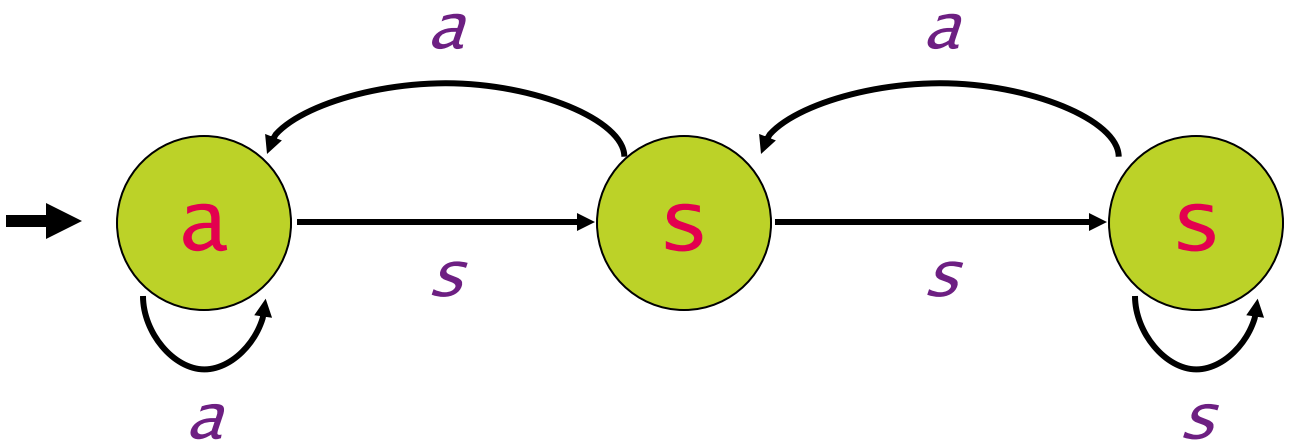
- In our version we study an iterated version against an opponent who follows a rational strategy
- Gain Function: limit of means (average over a very long series of moves)
- For example, if we get into a recurrent situation where we both admit, the gain will be -3

The general problem

- We suppose that the strategy of the opponent is given by a deterministic finite automaton (DFA)

- Can we imagine an optimal strategy?

Running example



Running example

- Then (game theory):

  Suppose we know the opponent's strategy

  – Consider the opponent's graph in which we value the edges by *our own gain* and find the best (infinite) path in the graph

Running example

Find the cycle of maximum mean weight

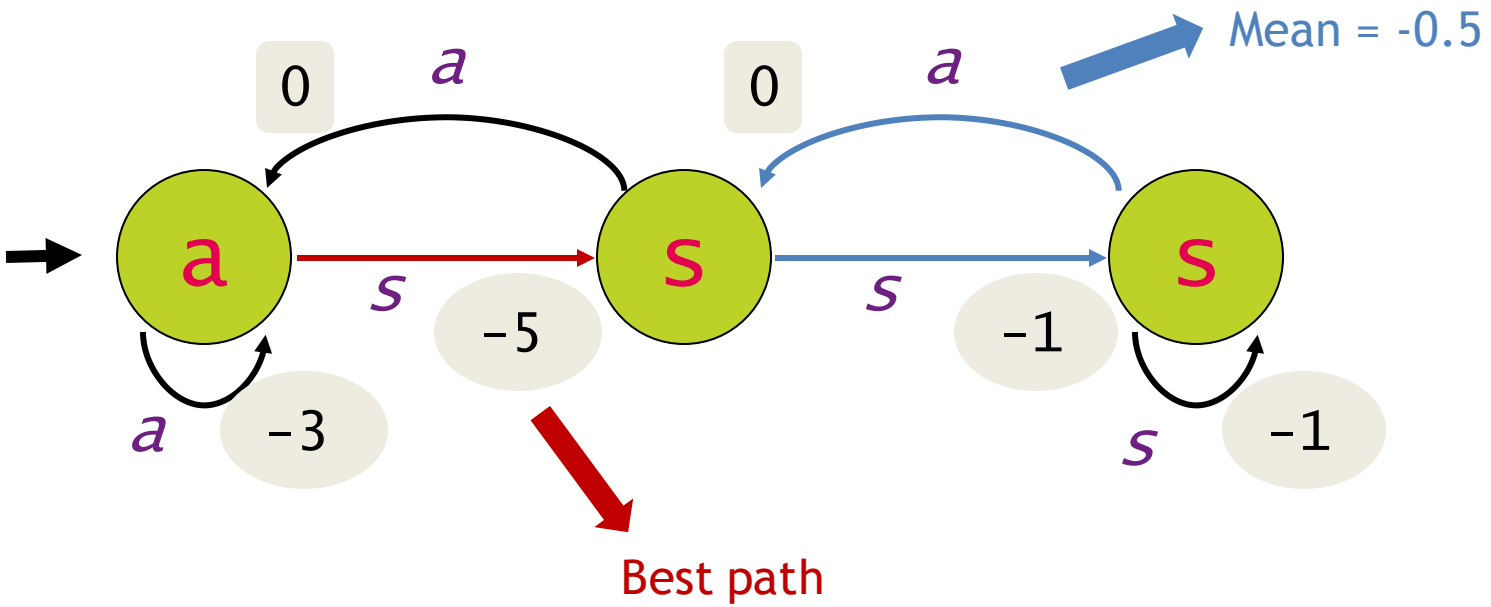
Find the best path leading to this cycle of maximum mean weight

Follow the path and stay in the cycle

Running example

Find the cycle of maximum mean weight → Find the best path leading to this cycle of maximum mean weight → Follow the path and stay in the cycle

	a	*s*
a	-3	0
s	-5	-1



Question

Can we play a game against this opponent and…

can we then **reconstruct** his strategy ?

The data (him, me)

HIM	ME
a	a
a	s
s	a
a	a
a	s
s	s
s	s
s	a
s	a

If I play asa, his move is a

$\lambda \rightarrow a$

$a \rightarrow a$

$as \rightarrow s$

$asa \rightarrow a$

$asaa \rightarrow a$

$asaas \rightarrow s$

$asaass \rightarrow s$

The logic of the algorithm

- The goal is to be able to parse and to have a partial solution consistent with the data
- The algorithm is loosely inspired by a number of grammatical inference algorithms
- It is greedy

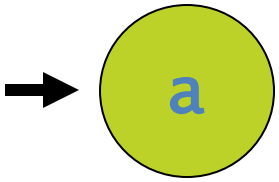
The algorithm

The first decision

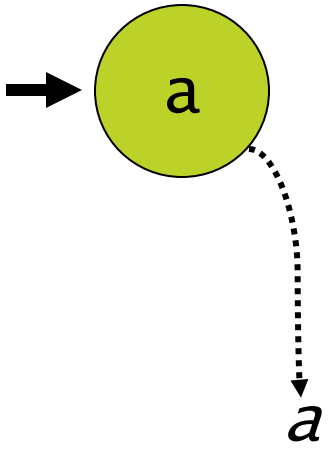
$\lambda \rightarrow a$

$a \rightarrow ?$

Sure:

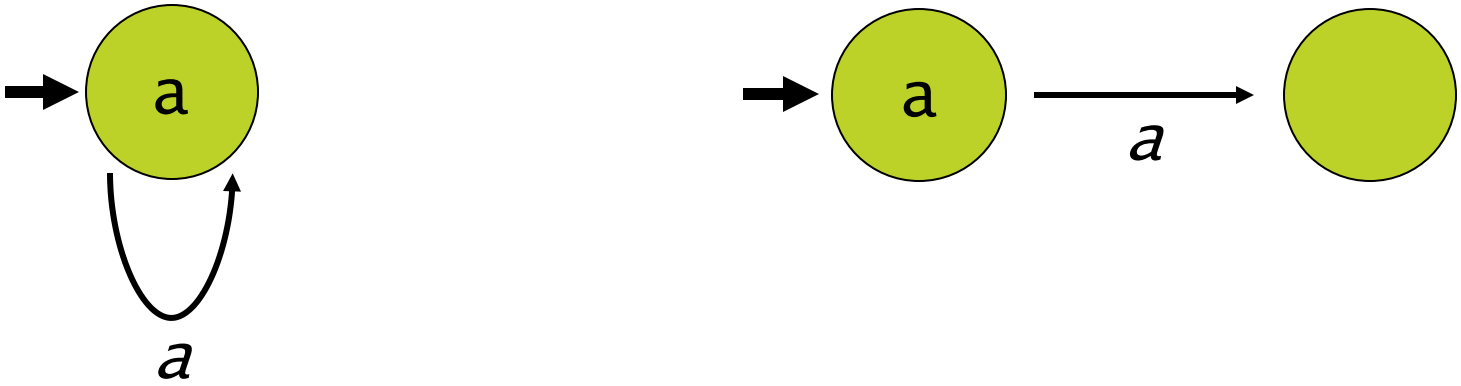


Have to deal with:



The algorithm

The candidates



Occam's razor

Entia non sunt multiplicanda praeter necessitatem

"Entities should not be multiplied unnecessarily"

The algorithm

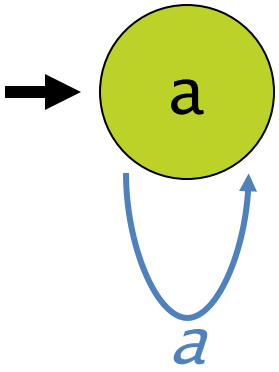
The second decision

$\lambda \rightarrow a$

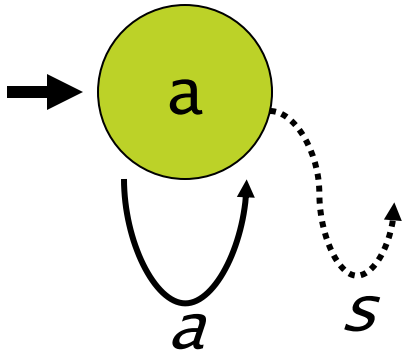
$a \rightarrow a$

$as \rightarrow ?$

Sure:



Have to deal with:

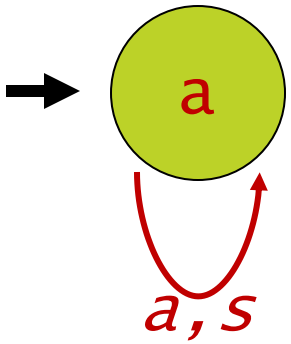


The algorithm

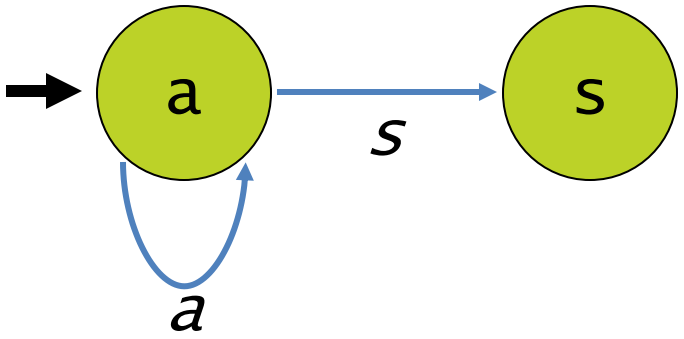
The third decision

$\lambda \rightarrow a$
$a \rightarrow a$
$as \rightarrow s$
$asa \rightarrow ?$

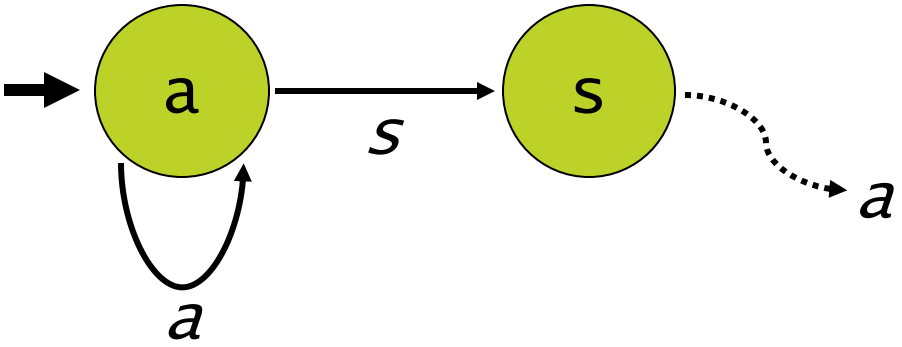
Inconsistent:



Consistent:

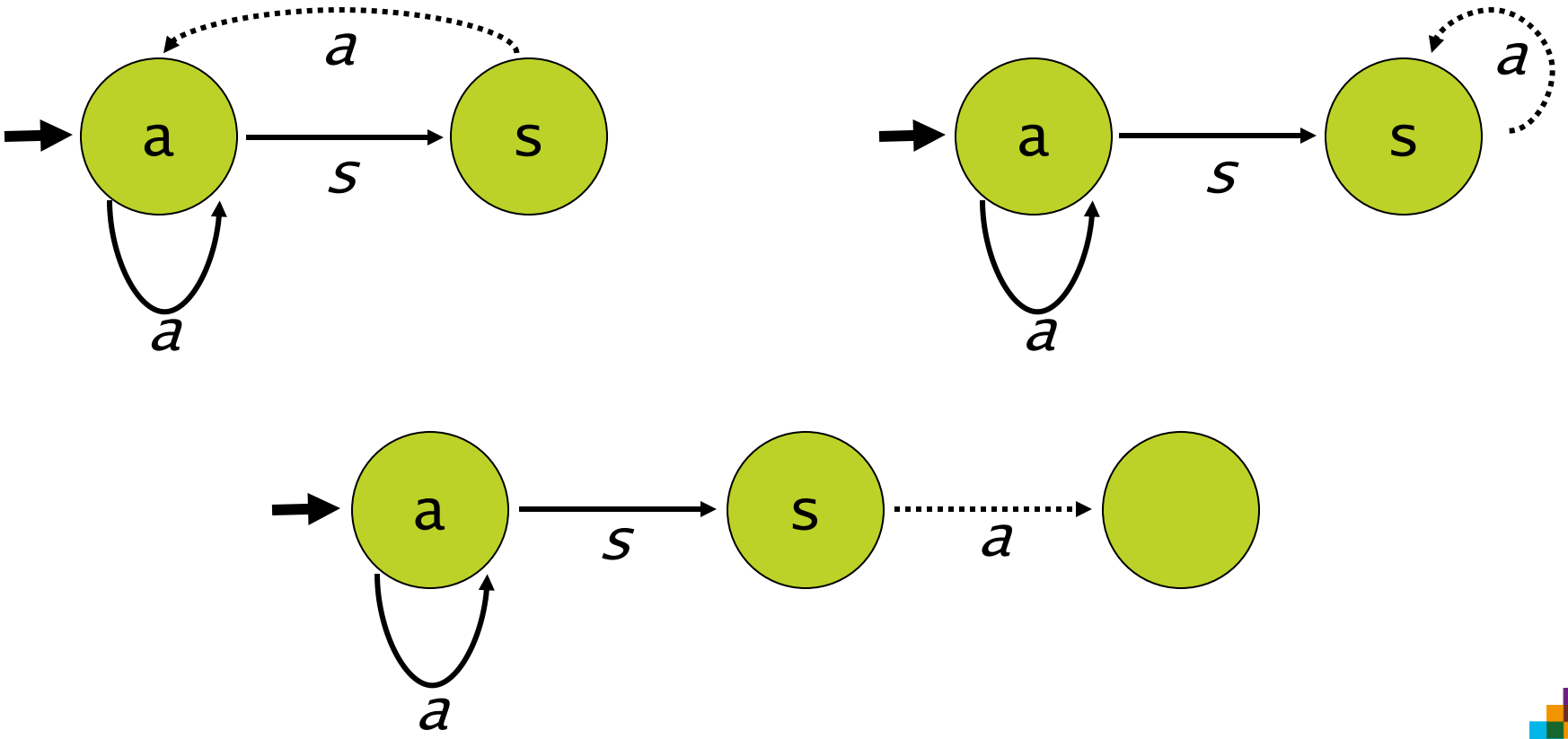


Have to deal with:



The algorithm

The three candidates



The algorithm

The fourth decision

$\lambda \rightarrow a$

$a \rightarrow a$

$as \rightarrow s$

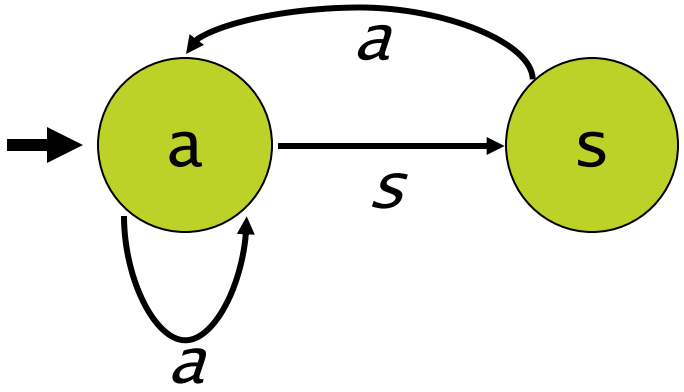
$asa \rightarrow a$

$asaa \rightarrow a$

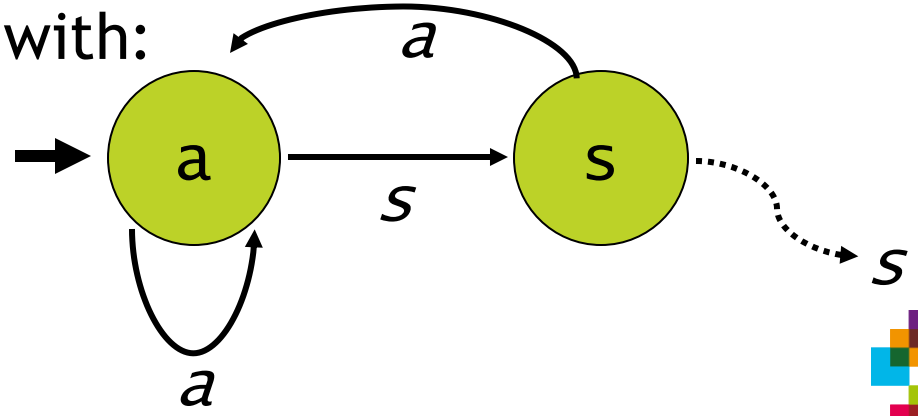
$asaas \rightarrow s$

$asaass \rightarrow ?$

Consistent:



Have to deal with:



The algorithm

The fifth decision

Inconsistent:

$\lambda \rightarrow a$

$a \rightarrow a$

$as \rightarrow s$

$asa \rightarrow a$

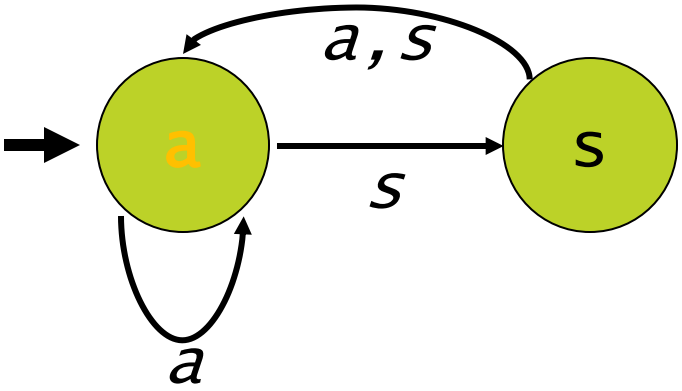
$asaa \rightarrow a$

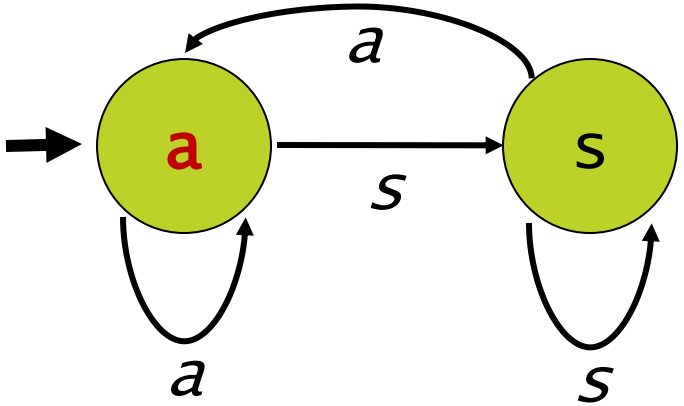
$asaas \rightarrow s$

$asaass \rightarrow s$

$asaasss \rightarrow s$

$asaasssa \rightarrow s$

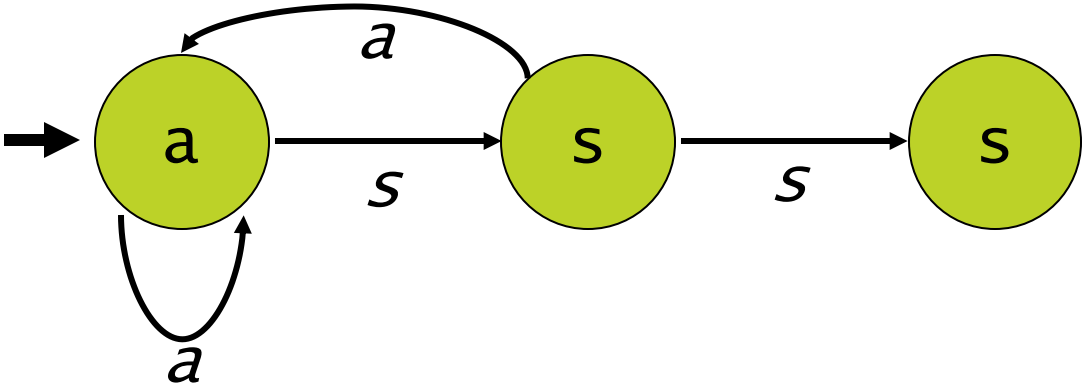




The algorithm

The fifth decision

Consistent:



$\lambda \rightarrow a$

$a \rightarrow a$

$as \rightarrow s$

$asa \rightarrow a$

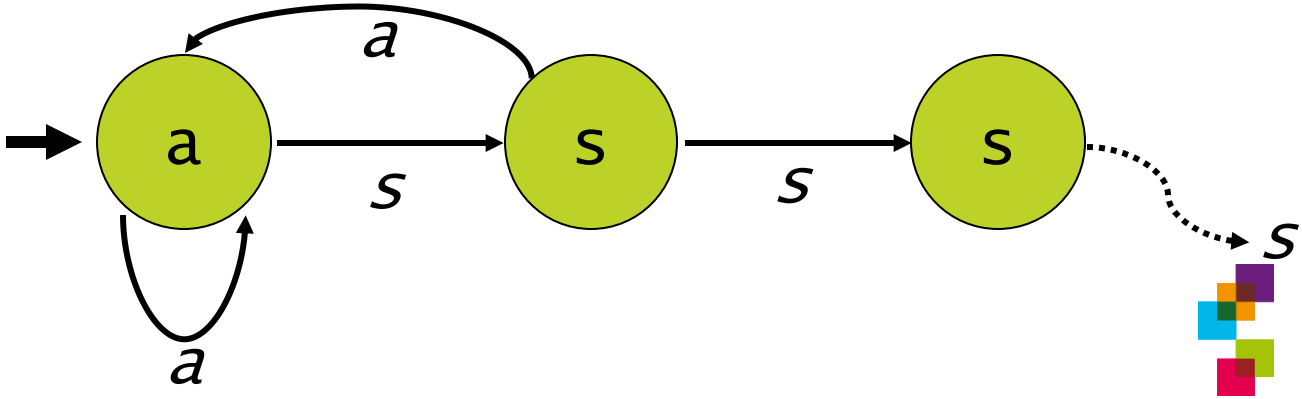
$asaa \rightarrow a$

$asaas \rightarrow s$

$asaass \rightarrow s$

$asaasss \rightarrow ?$

Have to deal with:



The algorithm

The sixth decision

Inconsistent:

$\lambda \rightarrow a$

$a \rightarrow a$

$as \rightarrow s$

$asa \rightarrow a$

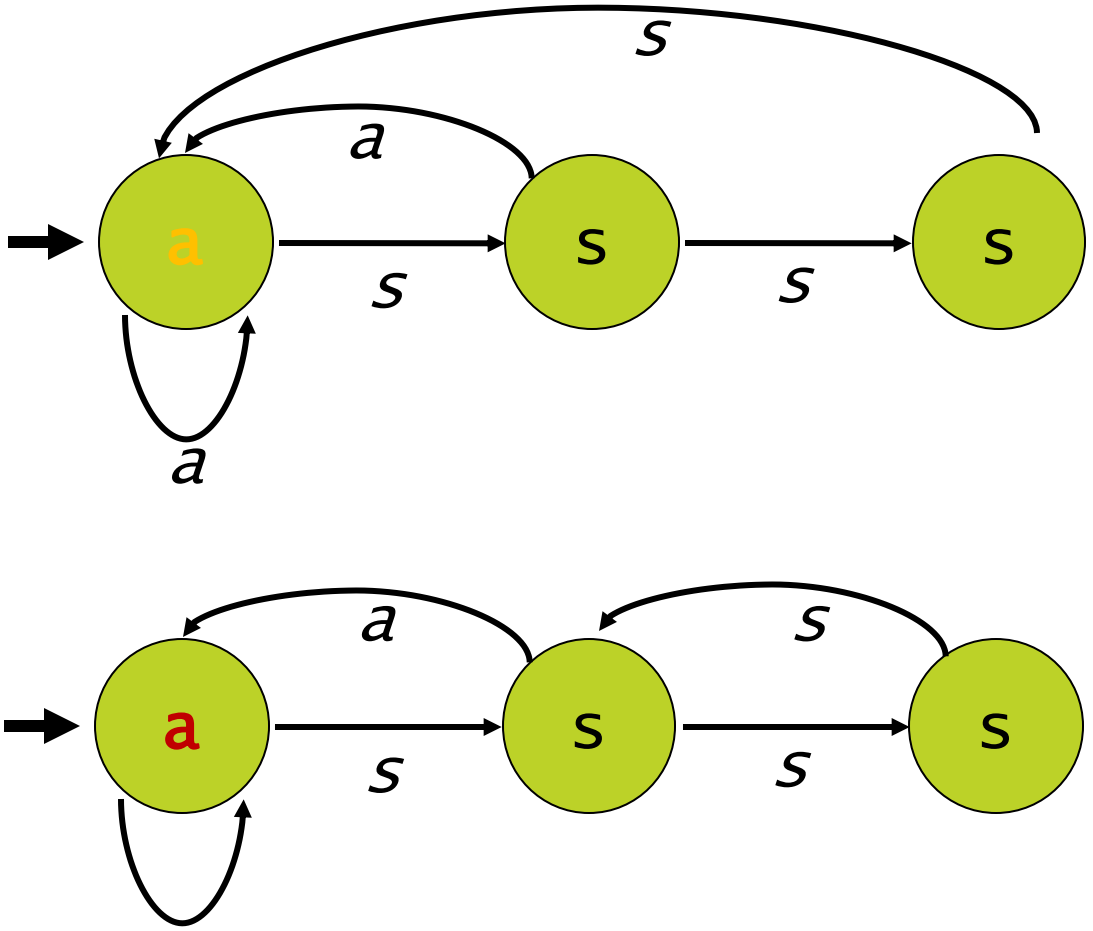
$asaa \rightarrow a$

$asaas \rightarrow s$

$asaass \rightarrow s$

$asaasss \rightarrow s$

$asaasssa \rightarrow s$

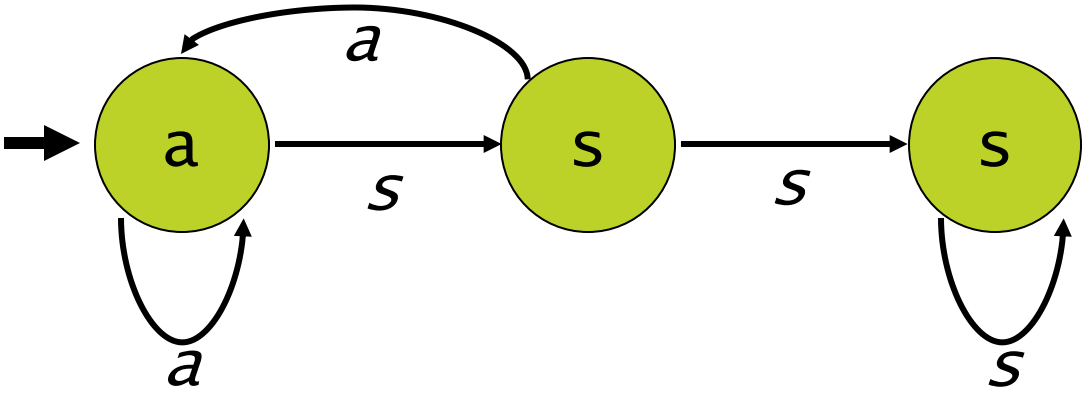


The algorithm

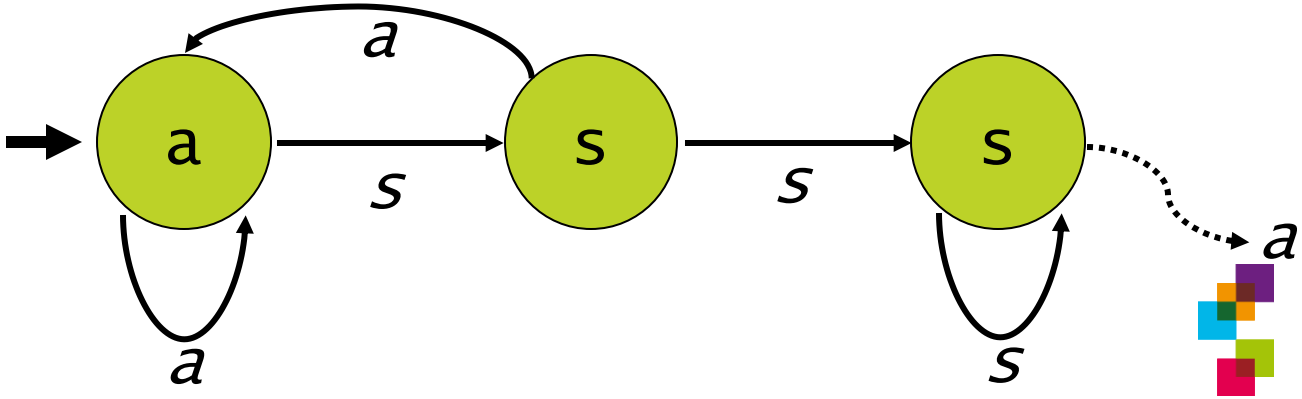
The sixth decision

Consistent:

$\lambda \rightarrow a$
$a \rightarrow a$
$as \rightarrow s$
$asa \rightarrow a$
$asaa \rightarrow a$
$asaas \rightarrow s$
$asaass \rightarrow s$
$asaasss \rightarrow s$
$asaasssa \rightarrow s$
$asaasssa \rightarrow ?$



Have to deal with:



The algorithm

$\lambda \rightarrow a$

$a \rightarrow a$

$as \rightarrow s$

$asa \rightarrow a$

$asaa \rightarrow a$

$asaas \rightarrow s$

$asaass \rightarrow s$

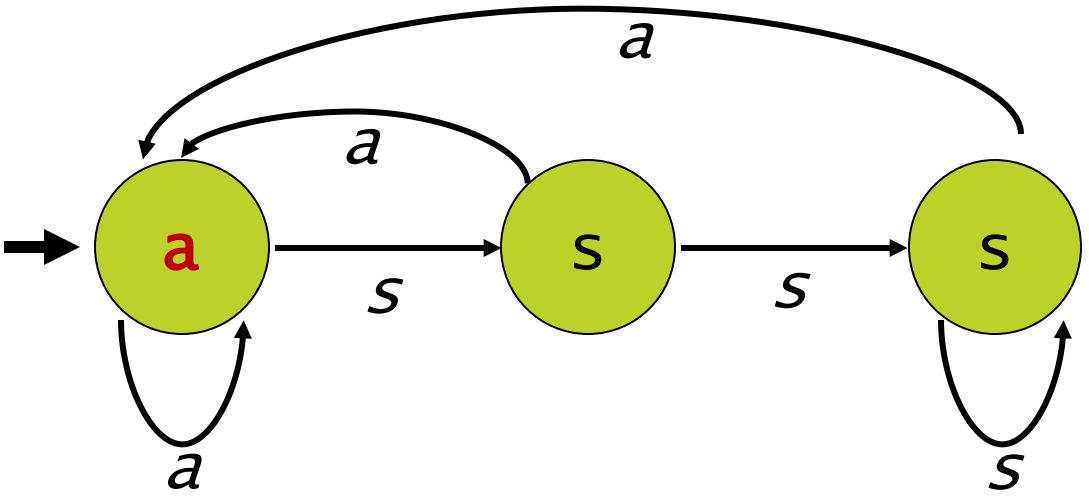
$asaasss \rightarrow s$

$asaasssa \rightarrow s$

$asaasssa \rightarrow s$

The seventh decision

Inconsistent:



The algorithm

$\lambda \rightarrow a$

$a \rightarrow a$

$as \rightarrow s$

$asa \rightarrow a$

$asaa \rightarrow a$

$asaas \rightarrow s$

$asaass \rightarrow s$

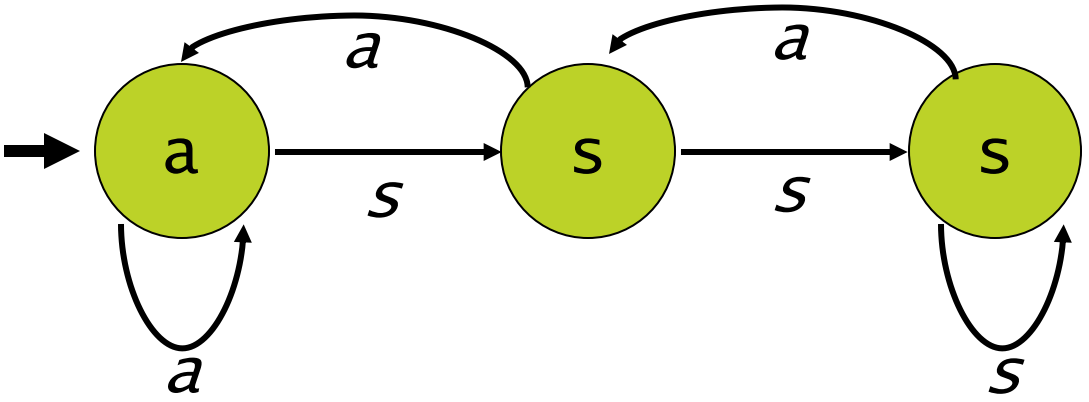
$asaasss \rightarrow s$

$asaasssa \rightarrow s$

$asaasssa \rightarrow s$

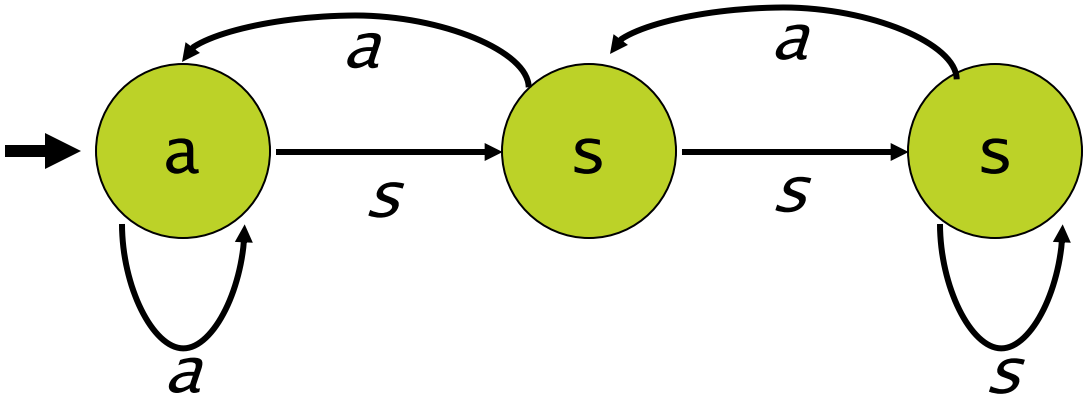
The seventh decision

Consistent:



The algorithm

The result



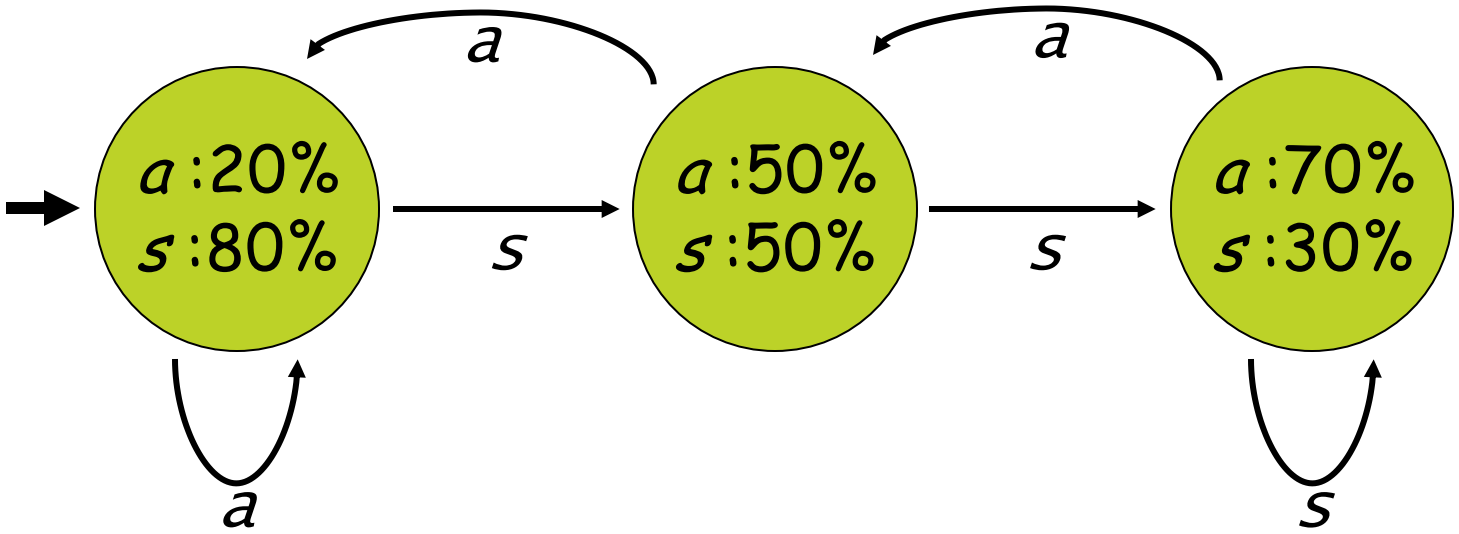
How do we get hold of the learning data?

a) through observation

b) through exploration (like here)

An open problem

The strategy is probabilistic:



Tit for tat