

Picobot



0

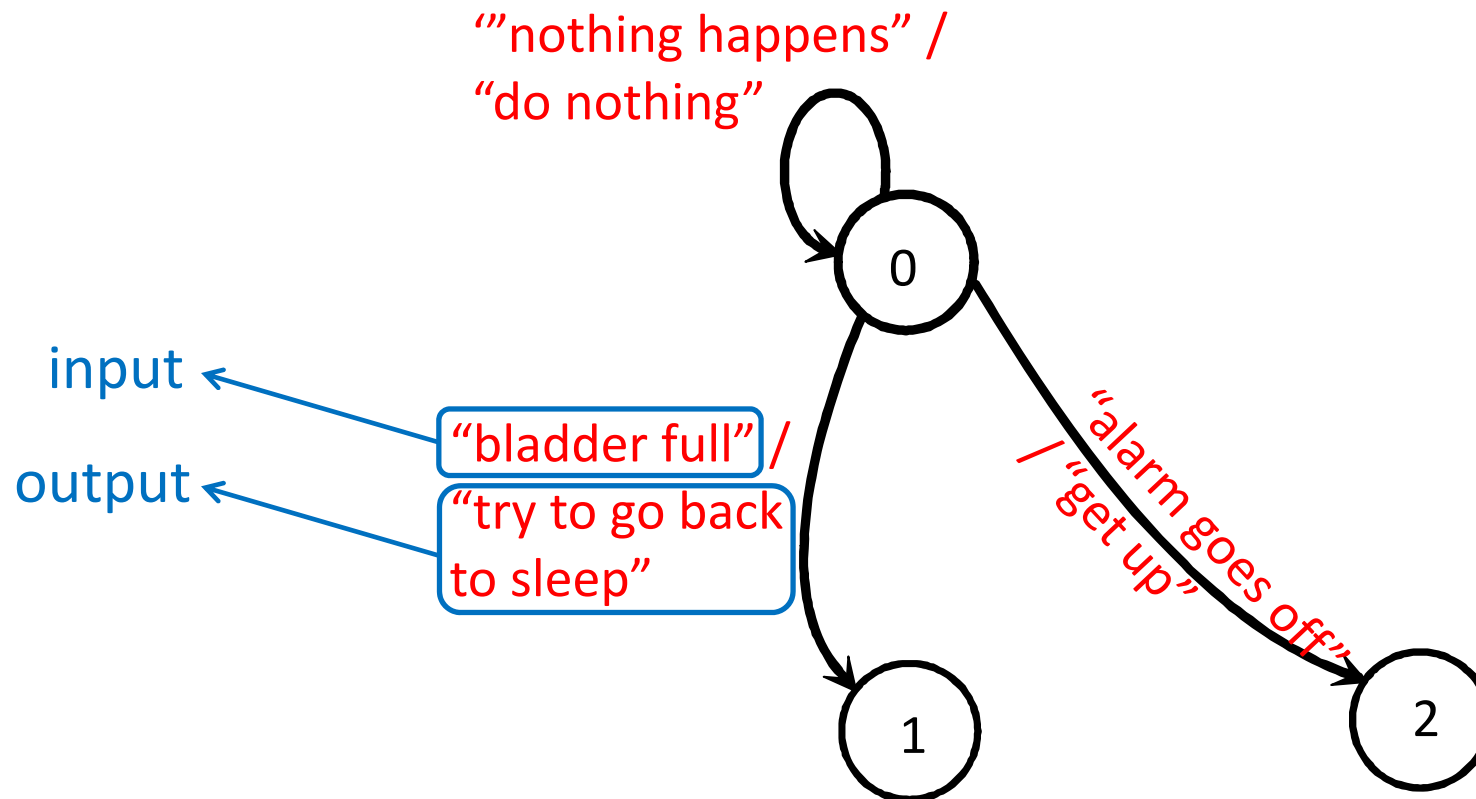
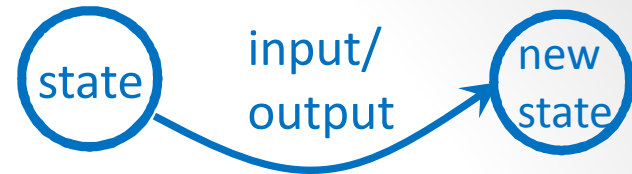
1

2

State 0: "in bed, asleep"

State 1: "in bed, awake"

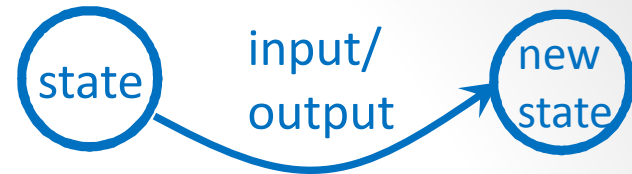
State 2: "in the bedroom, out of bed"



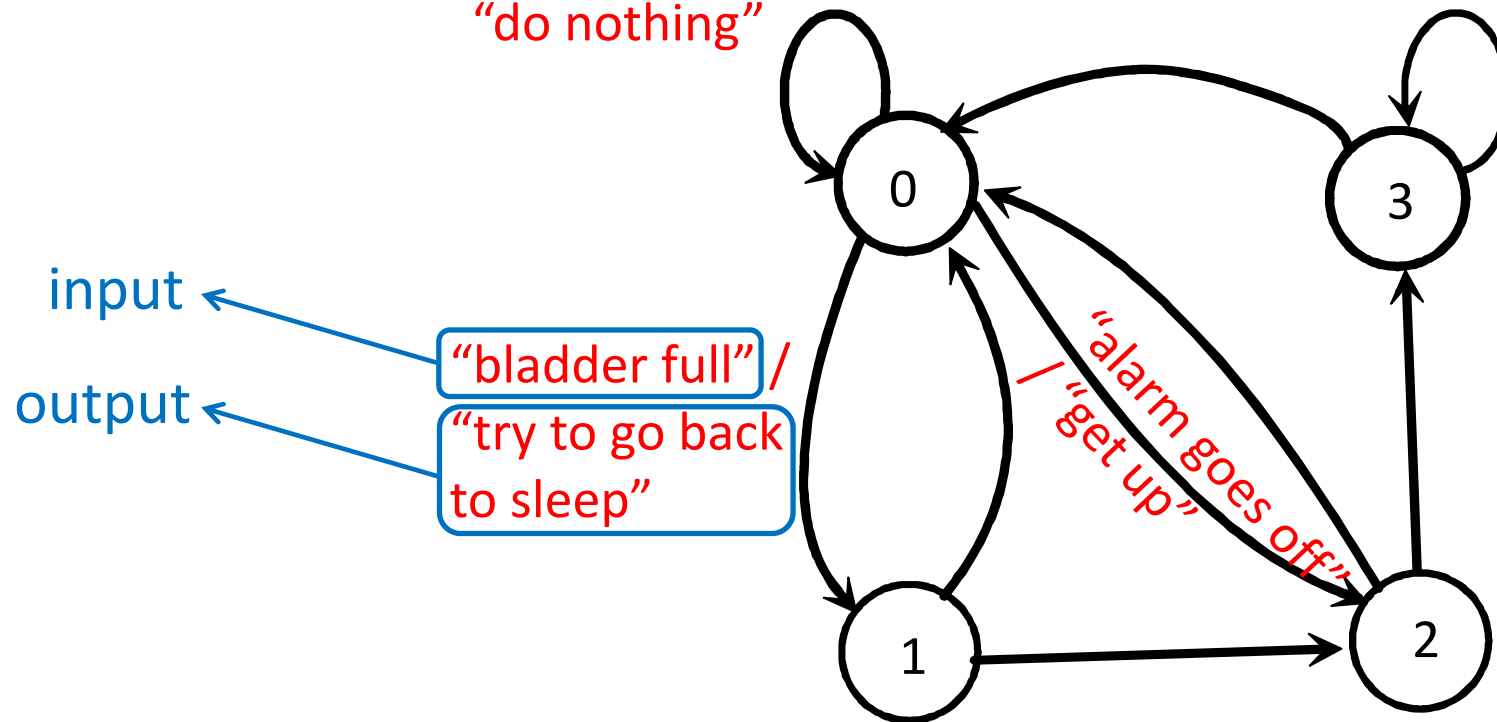
State 0: "in bed, asleep"

State 1: "in bed, awake"

State 2: "in the bedroom, out of bed"



“nothing happens” /
“do nothing”

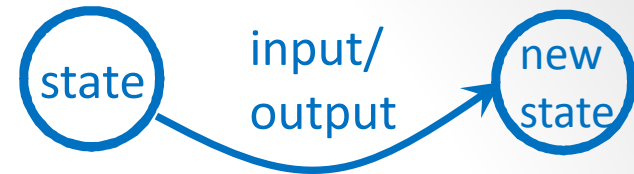


State 0: “in bed, asleep”

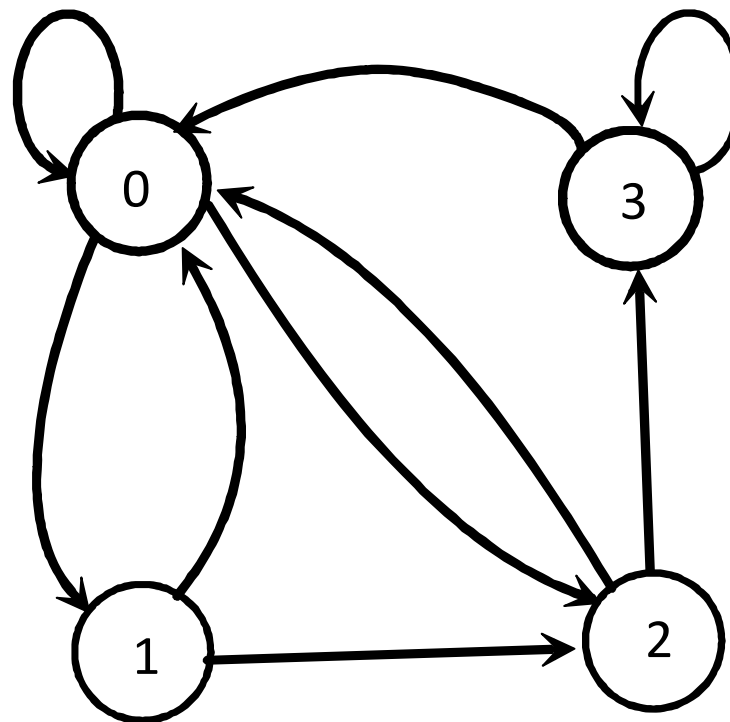
State 1: “in bed, awake”

State 2: “in the bedroom, out of bed”

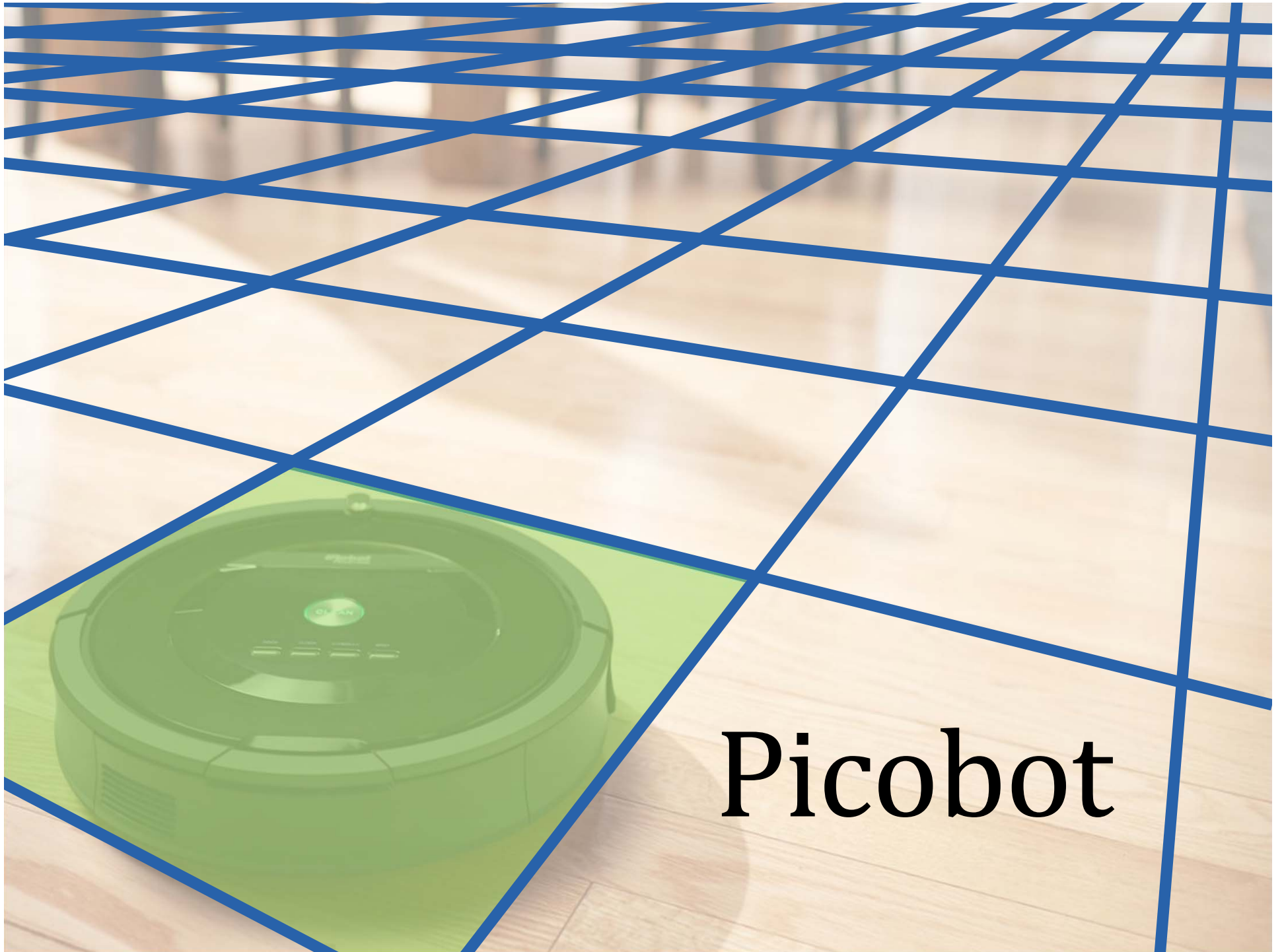
Finite State Machine (FSM)



Mealy FSM



state input \longrightarrow output new state



Picobot

Two languages in 4 weeks??

Python

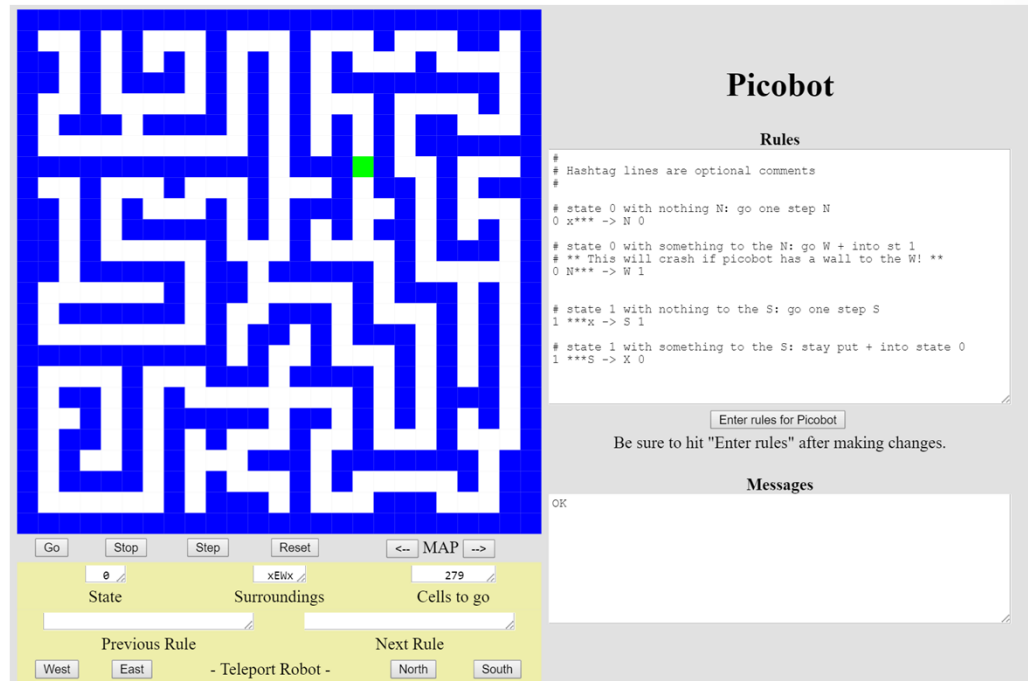
General-purpose language
you might see 20% by
the end of the
program

only 1% of its libraries!

Picobot

Special-purpose language
you'll see 100% in the next
10 minutes and learn
everything about it in the
next 1 hour

Picobot!



Picobot

Rules

```
# Hashtag lines are optional comments
#
# state 0 with nothing N: go one step N
0 x*** -> N 0
# state 0 with something to the N: go W + into st 1
# ** This will crash if picobot has a wall to the W! **
0 N*** -> W 1
# state 1 with nothing to the S: go one step S
1 ***x -> S 1
# state 1 with something to the S: stay put + into state 0
1 ***S -> X 0
```

Enter rules for Picobot

Be sure to hit "Enter rules" after making changes.

Messages

OK

Go Stop Step Reset MAP

0 xEnv 279

State Surroundings Cells to go

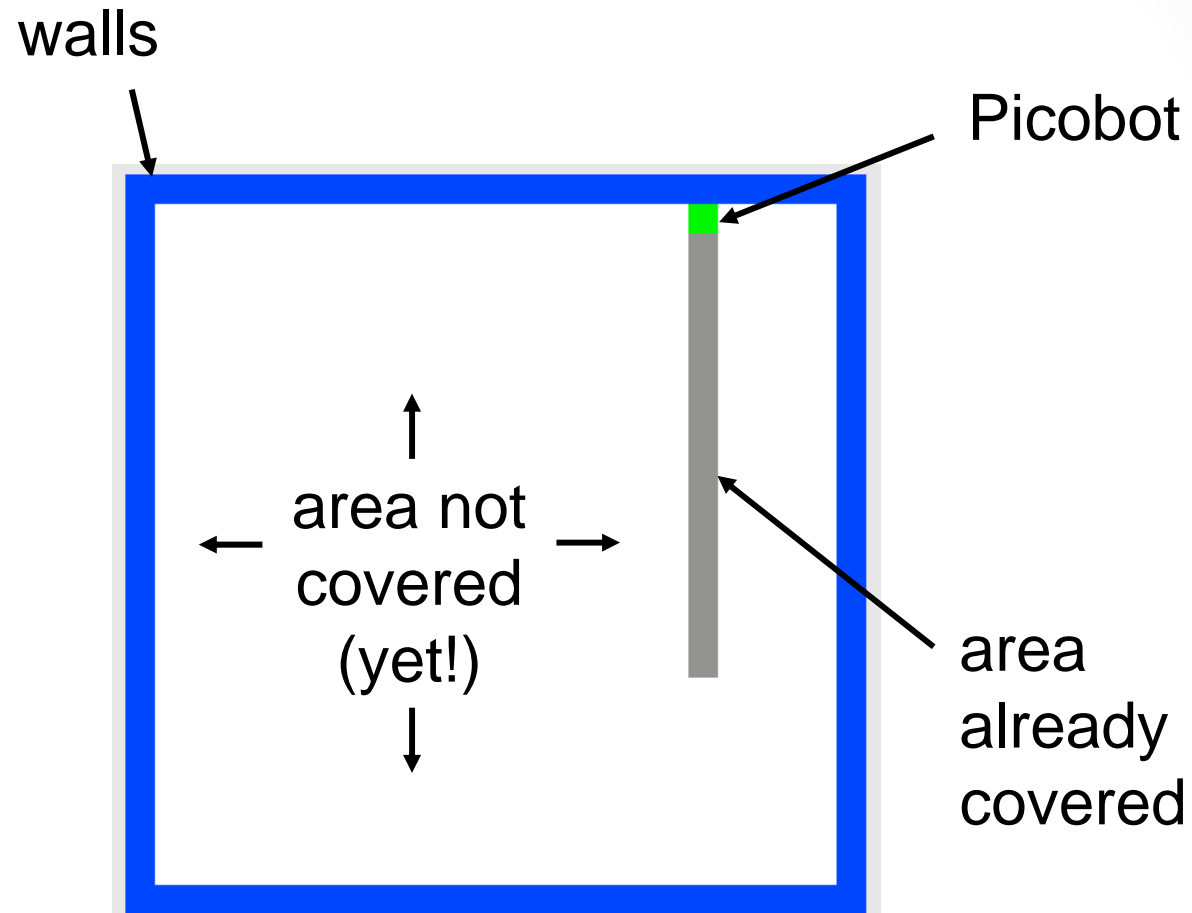
Previous Rule Next Rule

West East - Teleport Robot - North South

The Picobot simulator

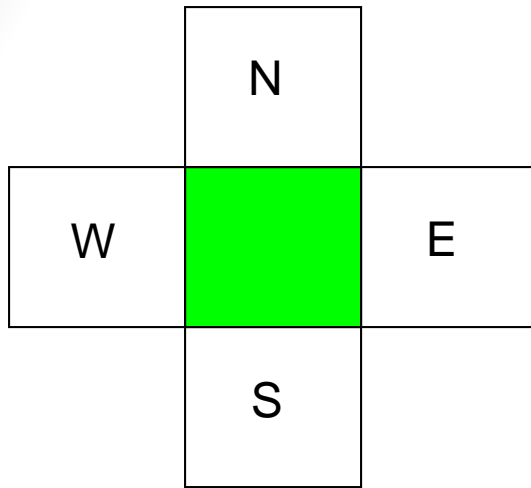
www.cs.hmc.edu/picobot

Picobot

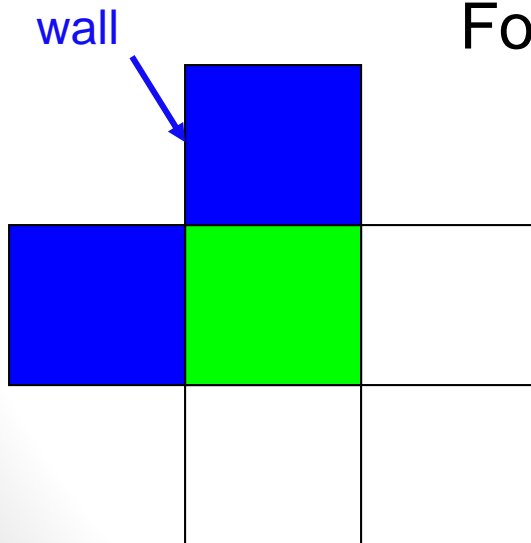


Goal: whole-environment coverage
with only *local sensing*...

Surroundings



Picobot can only sense things directly to the N, E, W, and S



For example, here its surroundings are

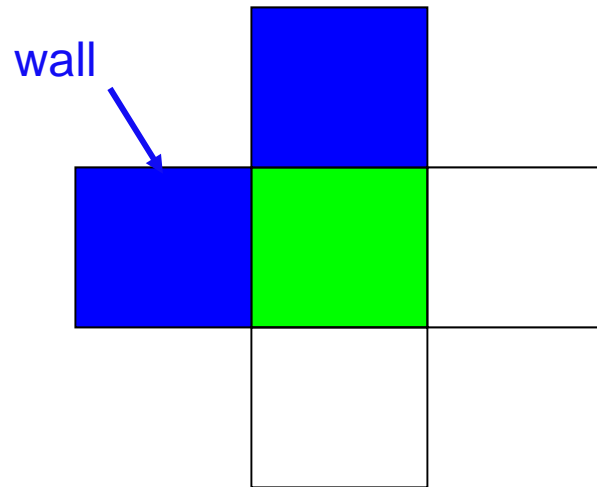
NxWx

N E W S

Surroundings are always in **NEWS** order.

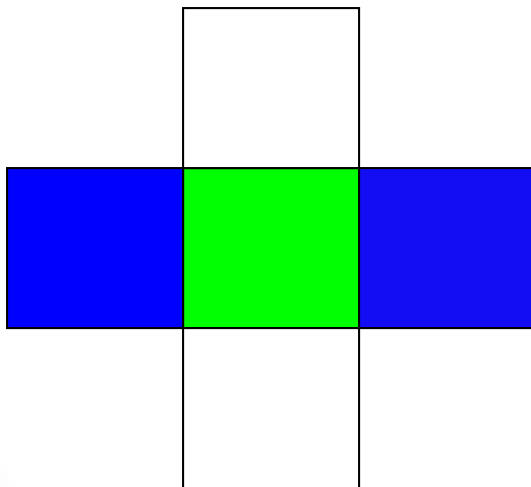
What are these surroundings?

Surroundings are always in **NEWS** order.



N E W S

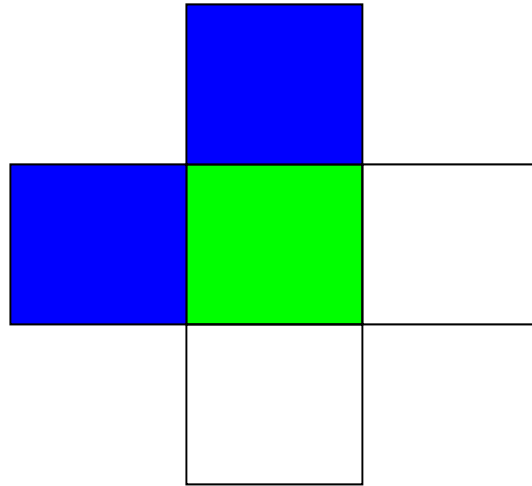
NxWx



- A. NxWx
- B. xEWx
- C. ExWx
- D. NxxS
- E. None of these

What are these surroundings?

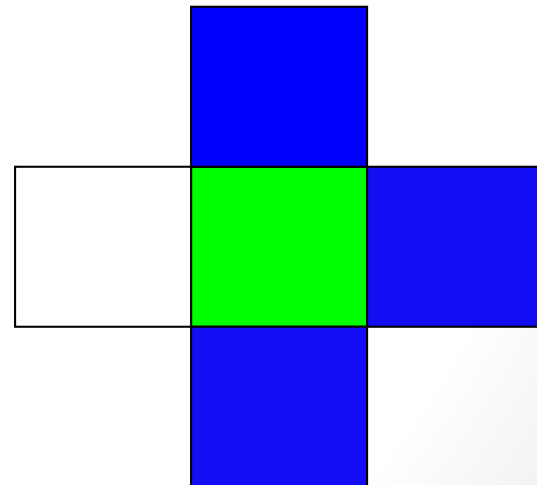
Surroundings are always in **NEWS** order.



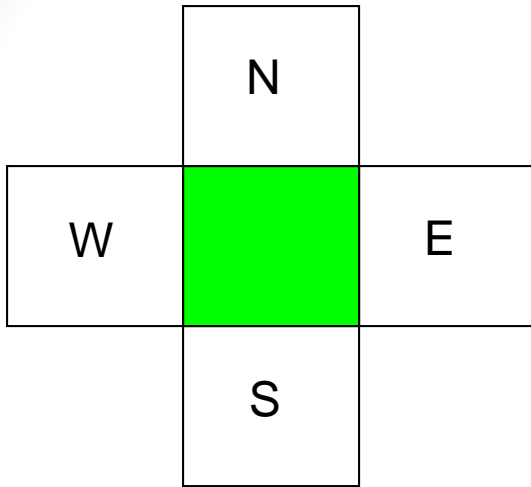
N E W S

NxWx

- A. NEWx
- B. NExS
- C. NxWS
- D. xxWx
- E. None of these



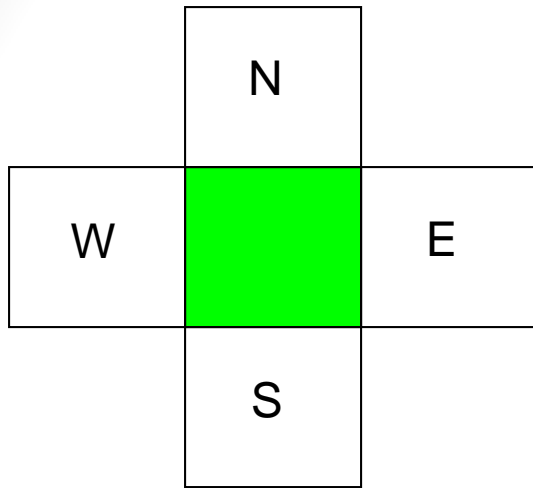
Surroundings



How many distinct surroundings are there?

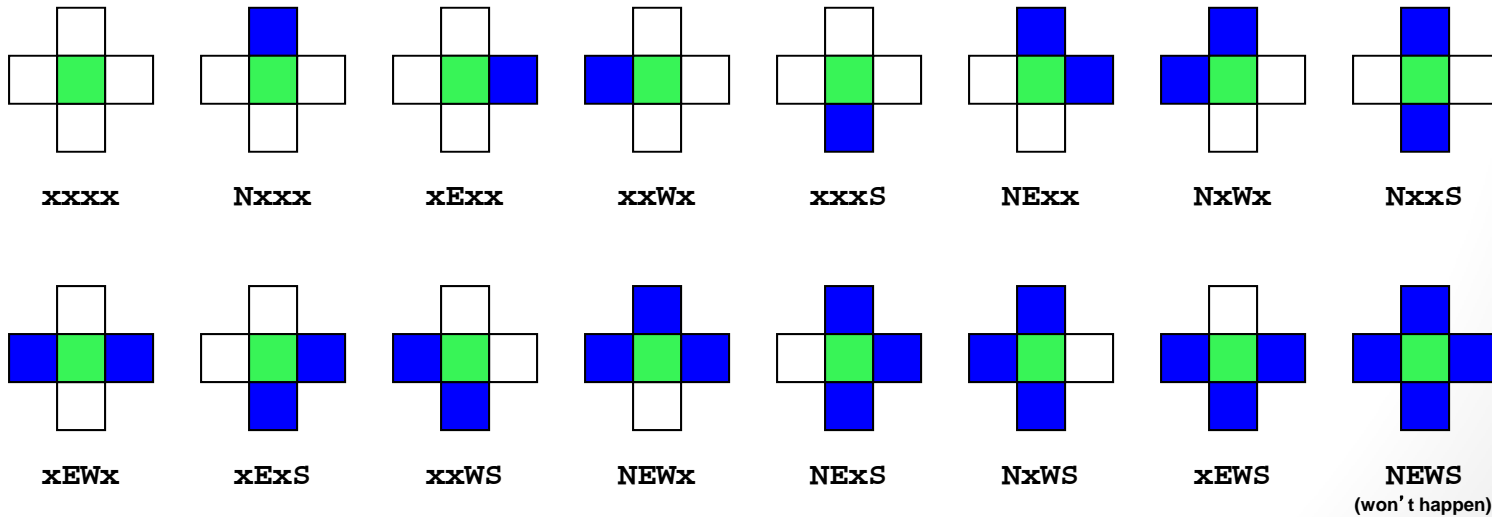
- A. 4
- B. 8
- C. 16
- D. 32
- E. 128

Surroundings

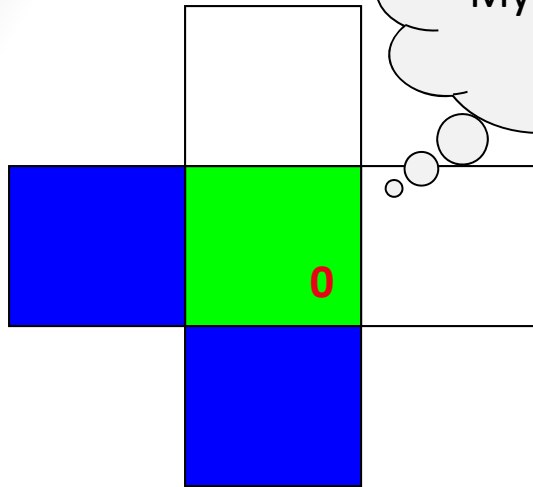


How many distinct surroundings are there?

$2^4 == 16$ possible ...



State



I am going North!
My surroundings
are xxWS.

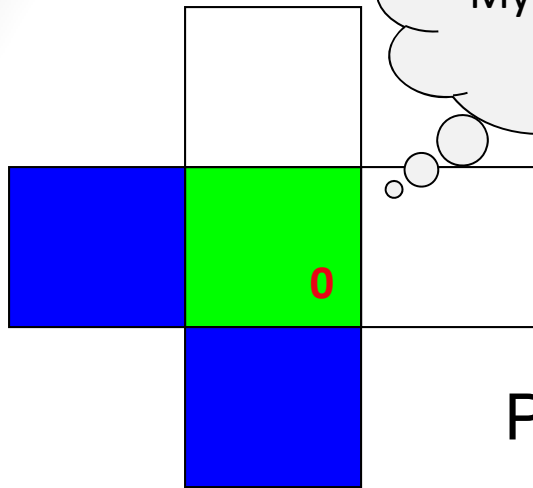
Picobot's memory is a single number, called its **state**.

State is picobot's way of remembering what has happened in the past!

Picobot always starts in **state 0**.

State and **surroundings** represent everything Picobot knows about the world

Rules



I am in state **0**.
My surroundings
are **xxWS**.

Picobot moves according to a set of rules:

state

surroundings

direction

new state

0

xxWS

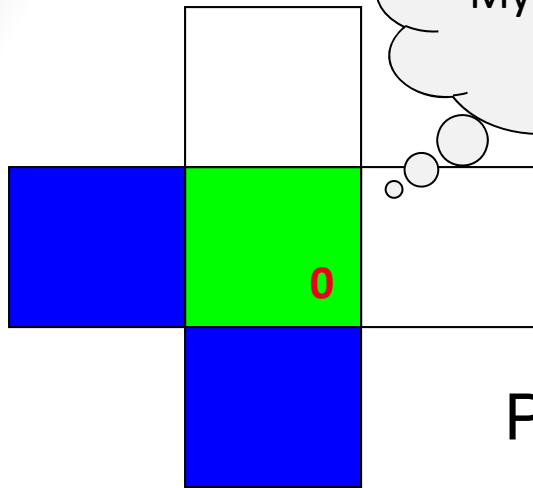


N

0

What does this rule mean? Summarize in your own words.

Rules



I am in state 0.
My surroundings
are xxWS.

Aha!

I should move N.

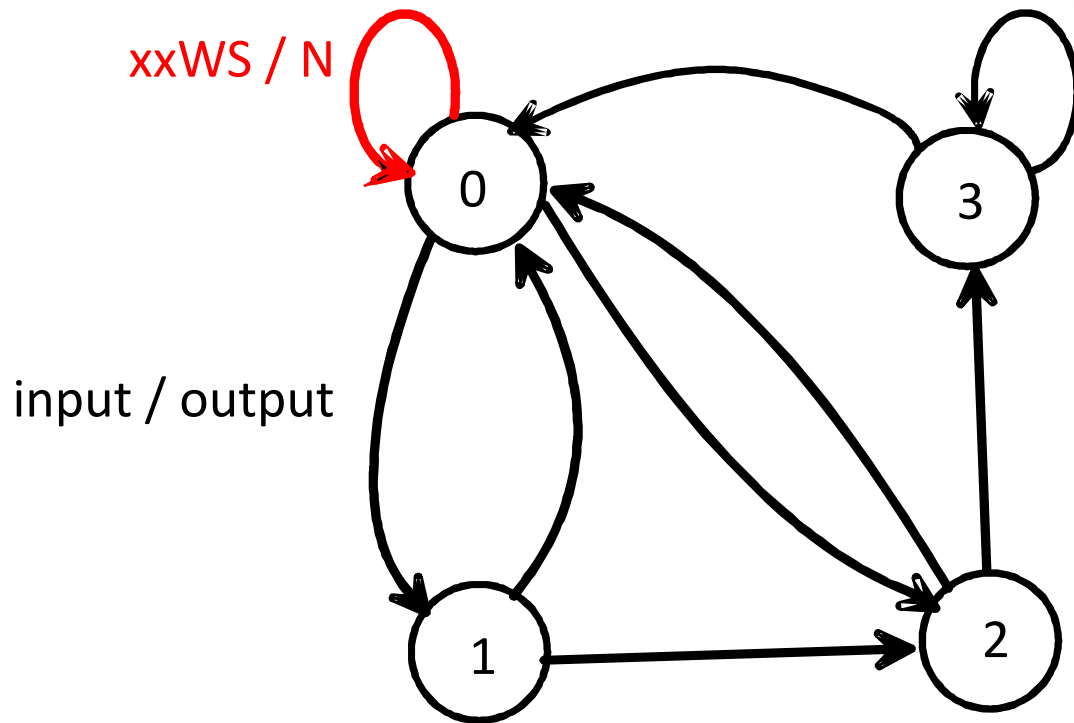
I should enter state 0.

Picobot moves according to a set of rules:

state	surroundings		direction	new state
0	xxWS	→	N	0

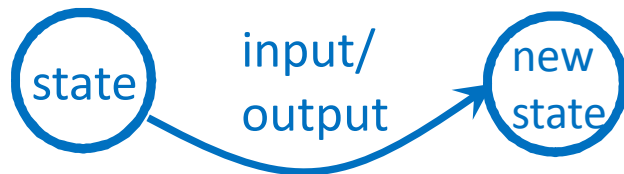
*If I'm in state 0
seeing xxWS,*

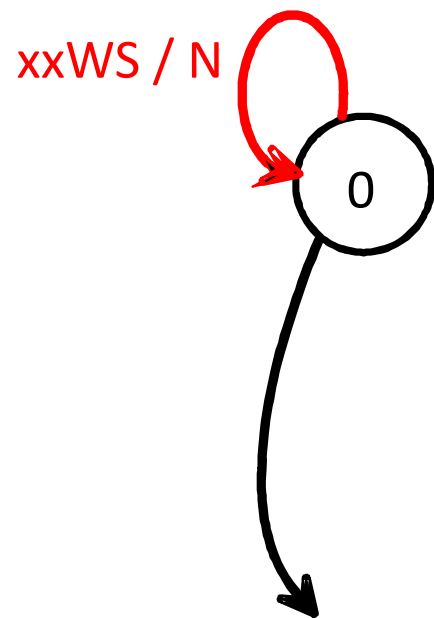
*Then I move **N**orth, and
"change" to state 0.*



state	surroundings	direction	new state
0	xxWS	→	0

state input output new state





state

surroundings

direction

new state

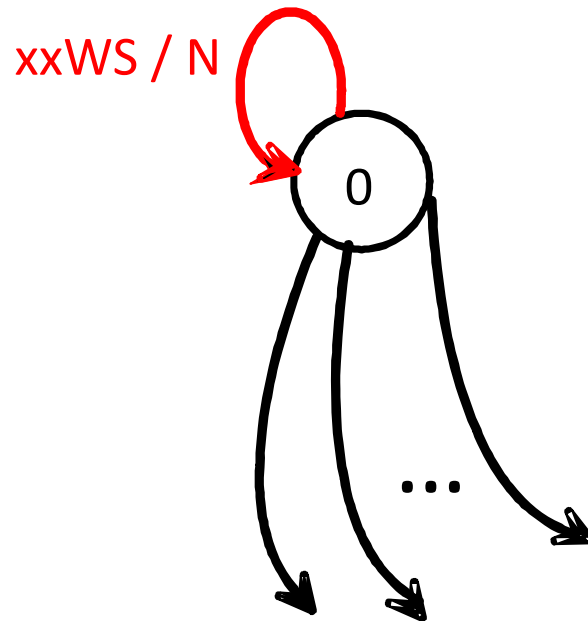
0

xxWS



N

0



How many
outgoing arrows
are there for
each state?

state

surroundings

direction

new state

0

xxWS



N

0

state	surroundings		direction	new state
0	xxxx	→	N	0
0	xxxS	→	N	0
0	xxWx	→	N	0
0	xxWS	→	N	0
0	xExx	→	N	0
0	xExS	→	N	0
0	xEWx	→	N	0
0	xEWS	→	N	0
0	Nxxx	→	X	1
0	NxxS	→	X	1
0	NxWx	→	S	0
0	NxWS	→	S	0
0	NExx	→	X	1
0	NExS	→	X	1
0	NEWx	→	S	0
0	NEWS	→	S	0

state surroundings direction new state

0	xxxx	→	N	0
0	xxxS	→	N	0
0	xxWx	→	N	0
0	xxWS	→	N	0
0	xExx	→	N	0
0	xExS	→	N	0
0	xEWx	→	N	0
0	xEWS	→	N	0
0	Nxxx	→	X	1
0	NxxS	→	X	1
0	NxWx	→	S	0
0	NxWS	→	S	0
0	NExx	→	X	1
0	NExS	→	X	1
0	NEWx	→	S	0
0	NEWS	→	S	0

state surroundings direction new state

0 x*** → N 0

0	Nxxx	→	X	1
0	NxxS	→	X	1
0	NxWx	→	S	0
0	NxWS	→	S	0
0	NExx	→	X	1
0	NExS	→	X	1
0	NEWx	→	S	0
0	NEWS	→	S	0

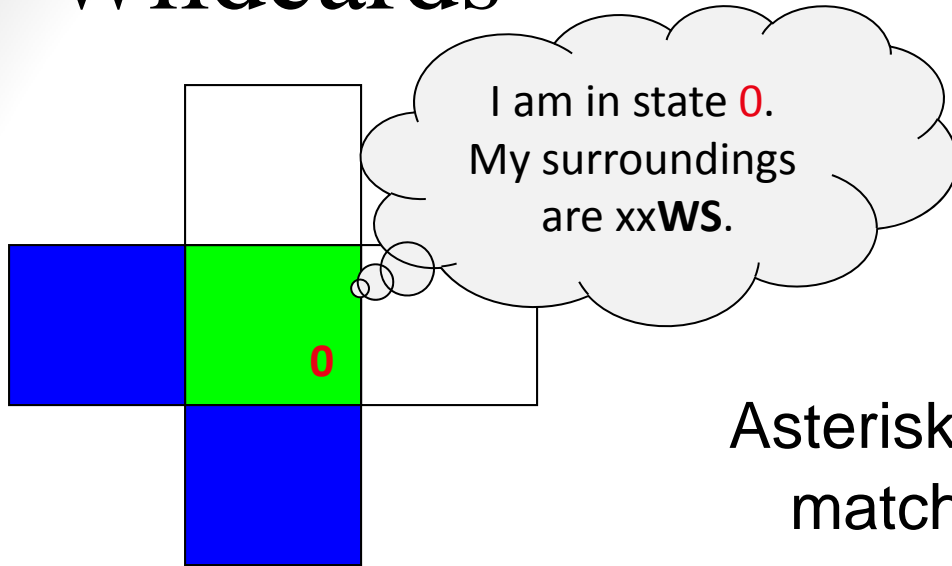
Wildcards

state	surroundings		direction	new state
0	X***	→	N	0

N must be empty

EWS may be wall *or* empty space

Wildcards



Aha!
This matches the
surroundings *x**

Asterisks * are wild cards. They
match walls **or** empty space:

state	surroundings	direction	new state
0	x****	→	0

N must be empty

EWS may be wall **or** empty space

state

surroundings

direction

new state

0

x***



N

0

state surroundings direction new state

0 x*** → N 0

0	xxxx	→	N	0
0	xxxS	→	N	0
0	xxWx	→	N	0
0	xxWS	→	N	0
0	xExx	→	N	0
0	xExS	→	N	0
0	xEWx	→	N	0
0	xEWS	→	N	0

state	surroundings		direction	new state
0	N*x*	→	X	1

How many rules does this expand to?

- A. 1
- B. 2
- C. 4
- D. 8
- E. 16

state	surroundings		direction	new state
-------	--------------	--	-----------	-----------

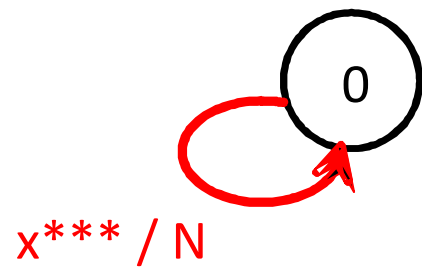
0	N*x*	→	X	1
---	------	---	---	---

0	Nxxx	→	X	1
0	NxxS	→	X	1
0	NExx	→	X	1
0	NExS	→	X	1

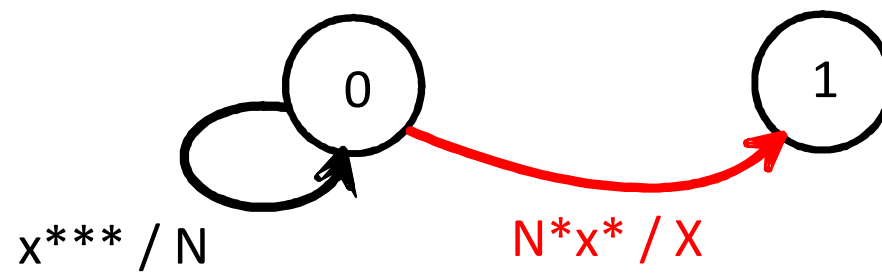
state	surroundings		direction	new state
0	x***	->	N	0
0	N*x*	->	X	1
0	N*W*	->	S	0

This covers all 16 options,
without duplication

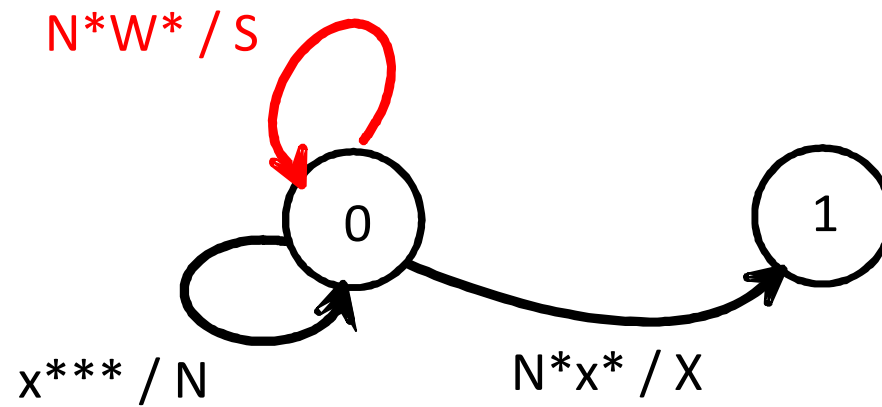
state	surroundings		direction	new state
0	x***	->	N	0
0	N*x*	->	X	1
0	N*W*	->	S	0



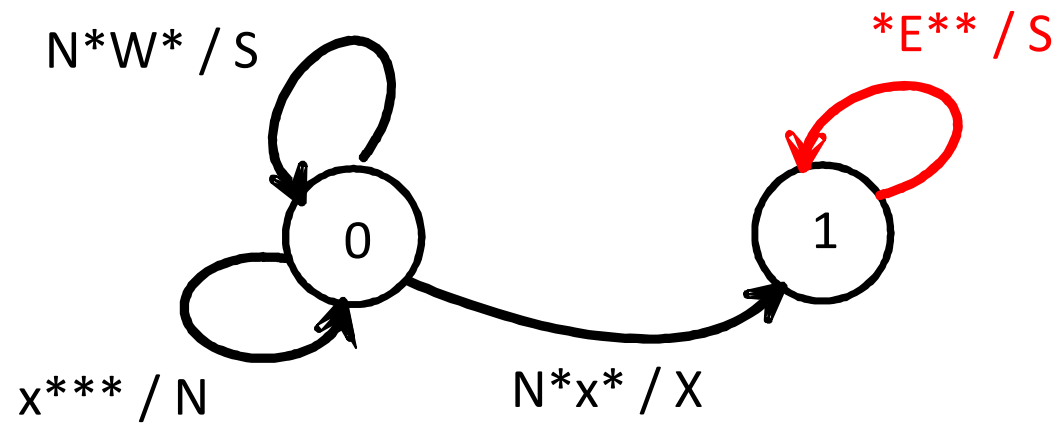
state	surroundings		direction	new state
0	x***	->	N	0
0	N*x*	->	X	1
0	N*W*	->	S	0



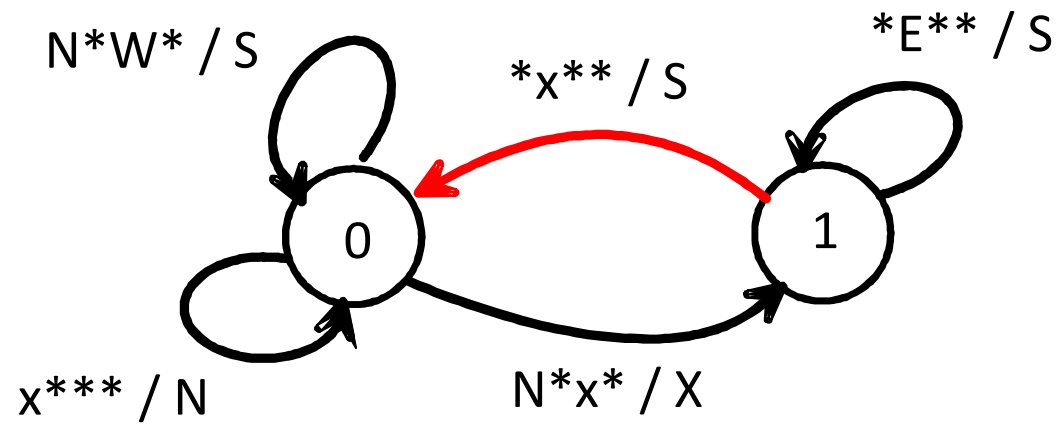
state	surroundings		direction	new state
0	x***	->	N	0
0	N*x*	->	X	1
0	N*W*	->	S	0



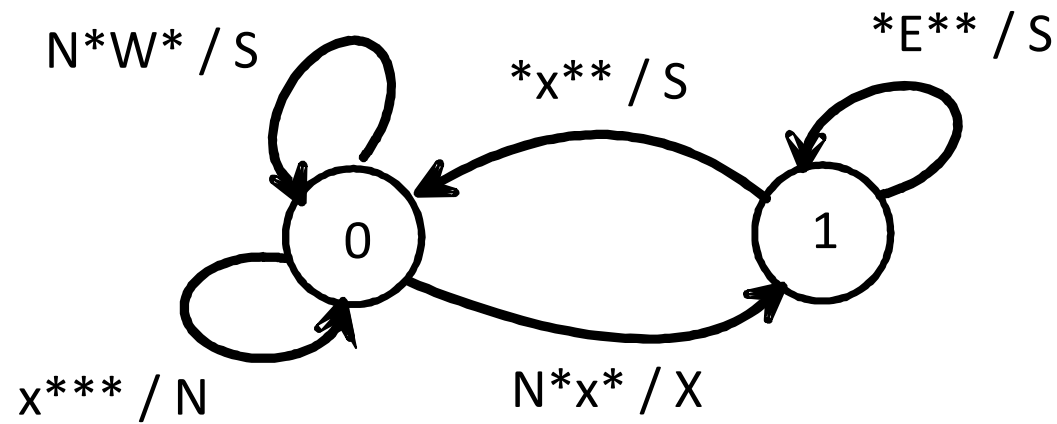
state	surroundings		direction	new state
0	x***	->	N	0
0	N*x*	->	X	1
0	N*W*	->	S	0
1	*E**	->	S	1



state	surroundings		direction	new state
0	x***	->	N	0
0	N*x*	->	X	1
0	N*W*	->	S	0
1	*E**	->	S	1
1	*x**	->	S	0



state	surroundings		direction	new state
0	x***	->	N	0
0	N*x*	->	X	1
0	N*W*	->	S	0
1	*E**	->	S	1
1	*x**	->	S	0



state

surroundings

direction

new state

Now let's build a Picobot
program step-by-step ...

Model

Picobot checks all rules each time.

When it finds a matching rule, that rule runs.

Only one rule is allowed per state and surroundings.

(1) What single rule sends the Picobot to the North (top) of the empty room ?

	state	surroundings		direction	new state
A:	0	x***	->	N	0
B:	0	xEW*	->	N	0
C:	0	x**x	->	N	0
D:	0	***x	->	S	0

Model

Picobot checks all rules each time.

When it finds a matching rule, that rule runs.

Only one rule is allowed per state and surroundings.

Picobot

Rules

```
#  
# Hashtag lines are optional comments  
#  
0 x*** -> N 0
```

Enter rules for Picobot

Be sure to hit "Enter rules" after making changes.

Messages

OK

Go Stop Step Reset <-- MAP -->

0 xxxx 528
State Surroundings Cells to go

none none

RECORDED Previous Rule

Next Rule

SCREENCAST MATELport Robot -

North

South

(2) What second rule can you add to avoid Picobot from crashing in that N wall?

state	surroundings		direction	new state
0	x***	->	N	0

A:	0	x***	->	N	1
B:	0	N***	->	N	1
C:	0	N***	->	X	1
D:	0	***x	->	X	1

Model

Picobot checks all rules each time.

When it finds a matching rule, that rule runs.

Only one rule is allowed per state and surroundings.

Picobot

Rules

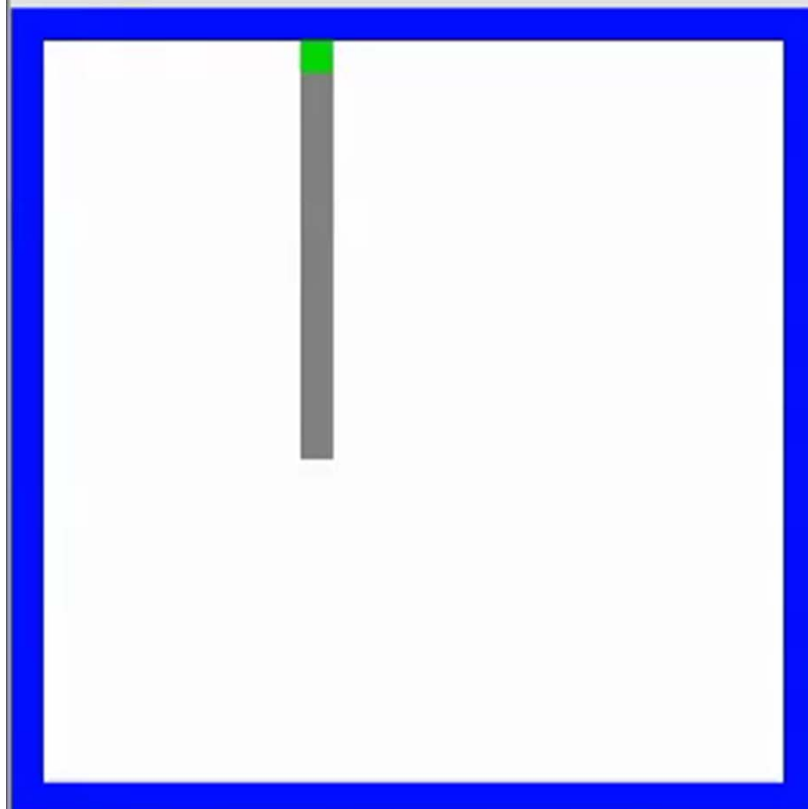
```
#  
# Hashtag lines are optional comments  
#  
0 x*** -> N 0
```

Enter rules for Picobot

Be sure to hit "Enter rules" after making changes.

Messages

```
No rule for state 0 and surr Nxxx  
Stopping.
```



Go Stop Step Reset <-- MAP -->

0 Nxxx 516
State Surroundings Cells to go

none none

RECORDED Previous Rule

Next Rule

SCREENCAST MATEIASort Robot -

North South

(3) How do we get back down?

state	surroundings		direction	new state
0	x***	->	N	0
0	N***	->	X	1
1	<u> ?</u>	->	S	1

- A. ***S
- B. N**X
- C. X**X
- D. XXX*
- E. ***X

Model

Picobot checks all rules each time.

When it finds a matching rule, that rule runs.

Only one rule is allowed per state and surroundings.

(3) How do we get back down?

state	surroundings		direction	new state
0	x***	->	N	0
0	N***	->	X	1
1	***x	->	S	1

Model

Picobot checks all rules each time.

When it finds a matching rule, that rule runs.

Only one rule is allowed per state and surroundings.

(4) How do we continue and go back up once reach the S wall?

state	surroundings		direction	new state
0	x***	->	N	0
0	N***	->	X	1
1	***x	->	S	1
1	***S	->	X	<u>?</u>

- A. 0
- B. 1
- C. 2
- D. x
- E. *

Model

Picobot checks all rules each time.

When it finds a matching rule, that rule runs.

Only one rule is allowed per state and surroundings.

(4) How do we continue and go back up?

state	surroundings		direction	new state
0	x***	->	N	0
0	N***	->	X	1
1	***x	->	S	1
1	***S	->	X	0

Model

Picobot checks all rules each time.

When it finds a matching rule, that rule runs.

Only one rule is allowed per state and surroundings.

Picobot

Rules

```
#  
# Hashtag lines are optional comments  
#  
0 x*** -> N 0  
0 N*** -> X 1  
1 ***x -> S 1  
1 ***S -> X 0
```

Enter rules for Picobot

Be sure to hit "Enter rules" after making changes.

Messages

No rule for state 1 and surr Nxxx
Stopping.

Go Stop Step Reset <-- MAP -->

1 Nxxx 509
State Surroundings Cells to go

none none

RECORDED Previous Rule

Next Rule

SCREENCAST MATEIASort Robot -

North South

Issues

state	surroundings		direction	new state
0	x***	->	N	0
0	N***	->	X	1
0	N*W*	->	S	0
1	N***	->	S	1
1	x**S	->	S	0

What are the problems with this program?

Issues

state	surroundings		direction	new state
0	x***	->	N	0
0	N***	->	X	1
0	N*W*	->	S	0
1	N***	->	S	1
1	x**S	->	S	0
1	x***x not defined			
0	N*W* defined twice			

Go south when it is blocked

Rules

```
#  
# Hashtag lines are optional comments  
#  
0 x*** -> N 0  
0 N*** -> X 1  
0 N**W -> S 0  
1 N*** -> S 1  
1 x**S -> S 0
```

Enter rules for Picobot

Be sure to hit "Enter rules" after making changes.

Messages

OK

Go Stop Step Reset <-- MAP -->

0

xxxx

528

State

Surroundings

Cells to go

1 X**S -> S 0

1 X**S -> S 0

Previous Rule

Next Rule

West

East

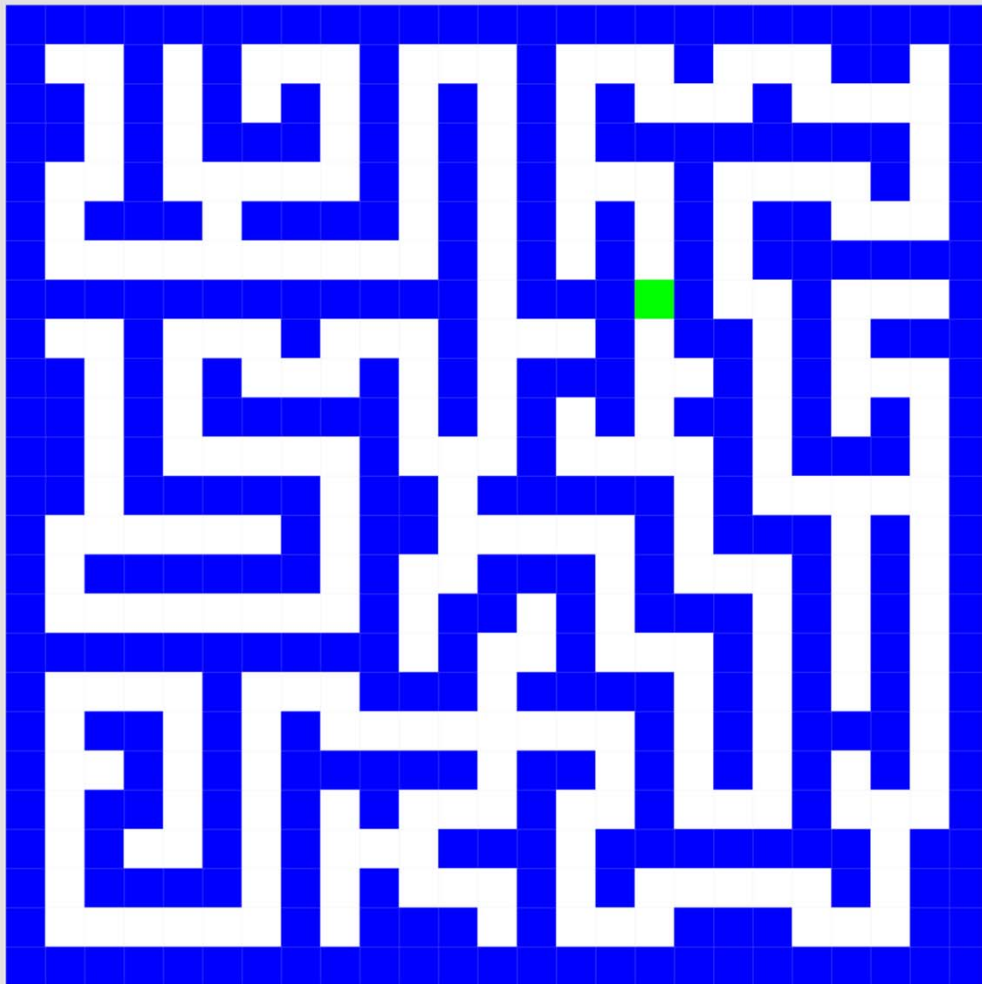
- Teleport Robot -

North

South

RECORDED WITH

SCREENCAST  MATIC



Picobot

Rules

```
#  
# Hashtag lines are optional comments  
#  
# state 0 with nothing N: go one step N  
0 x*** -> N 0  
  
# state 0 with something to the N: go W + into st 1  
# ** This will crash if picobot has a wall to the W! **  
0 N*** -> W 1  
  
# state 1 with nothing to the S: go one step S  
1 ***x -> S 1  
  
# state 1 with something to the S: stay put + into state 0  
1 ***S -> X 0
```

Enter rules for Picobot

Be sure to hit "Enter rules" after making changes.

Messages

OK

Go Stop Step Reset <-- MAP -->

0

xEWx

279

State

Surroundings

Cells to go

Previous Rule

Next Rule

West

East

- Teleport Robot -

North

South



**WEEKS OF PROGRAMMING
CAN SAVE YOU
HOURS OF PLANNING**