

Ask me anything

3 questions
1 upvotes

What transitions are possible in the 2x2 composition of MAKER and USER?

(0,0)->make->(1,0)-
>ready->(0,1)->use->
(0,0)

make -> ready ->
use(ready is
synchronised)

make->ready-
>usemake->ready-
>make->use

MAKER = (make->ready->MAKER) .
USER = (ready->use->USER) .

|| MAKER_USER = (MAKER || USER) .

What happens if you change ready in MAKER to ready1 and in USER to ready2?

The result will be two independent actions, no synchronization

Completely independent, so it is the "unrestricted" cross product

Then we will have two independent processes

They are not synchronized anymore and completely independent.

We will have independent actions. Cross product of all actions.

What transitions are possible in the 3x3 composition of MAKER2 and USER2?

```
MAKERv2 = (make -> ready -> used -> MAKERv2).  
USERv2  = (ready -> use -> used -> USERv2).  
|| MAKER_USERv2 = (MAKERv2 || USERv2).
```

(make -> ready -> use
-> used)*

(make -> ready -> use
-> used) ad infinitum

START = (make ->
ready -> use -> used ->
START)

The maker waits
until the user used.
We are eliminating
one path.

What can happen if we leave +VarAlpha away from the Inc process?

```
const N = 3
range T = 0..N
set VarAlpha = {read[T],write[T]}

Var      = Var[0],
Var[u:T] = ( read[u]  ->Var[u]
            | write[v:T]->Var[v]).

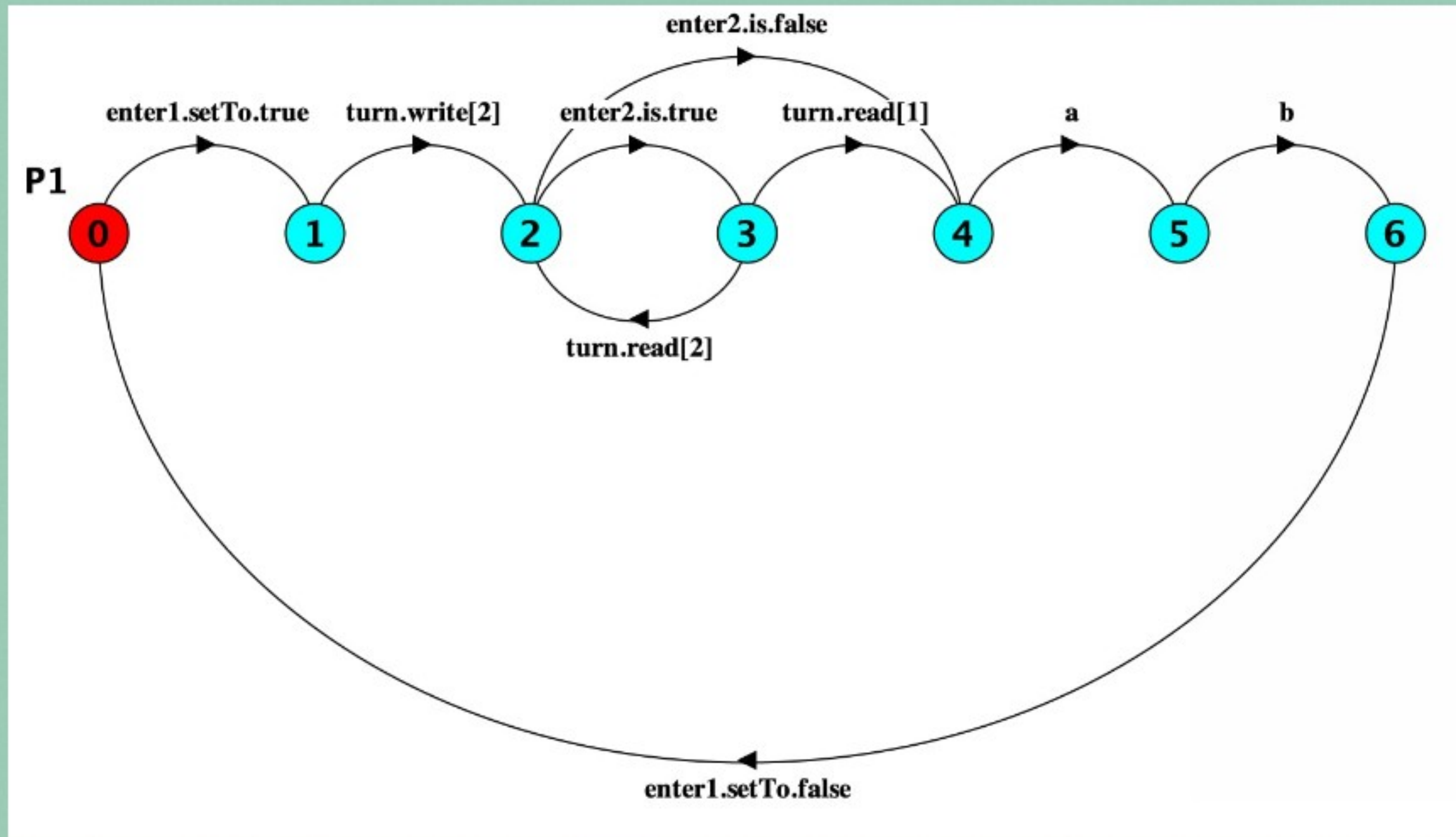
Inc = (read[v:0..N-1] -> write[v+1] -> STOP)+VarAlpha.
||SeqInc = (Var || Inc).
```

Then, we can read every possible state of the variable at any time

If we eliminate the VarAlpha, the increment might perform random read and writes without performing a sing-step incrementation

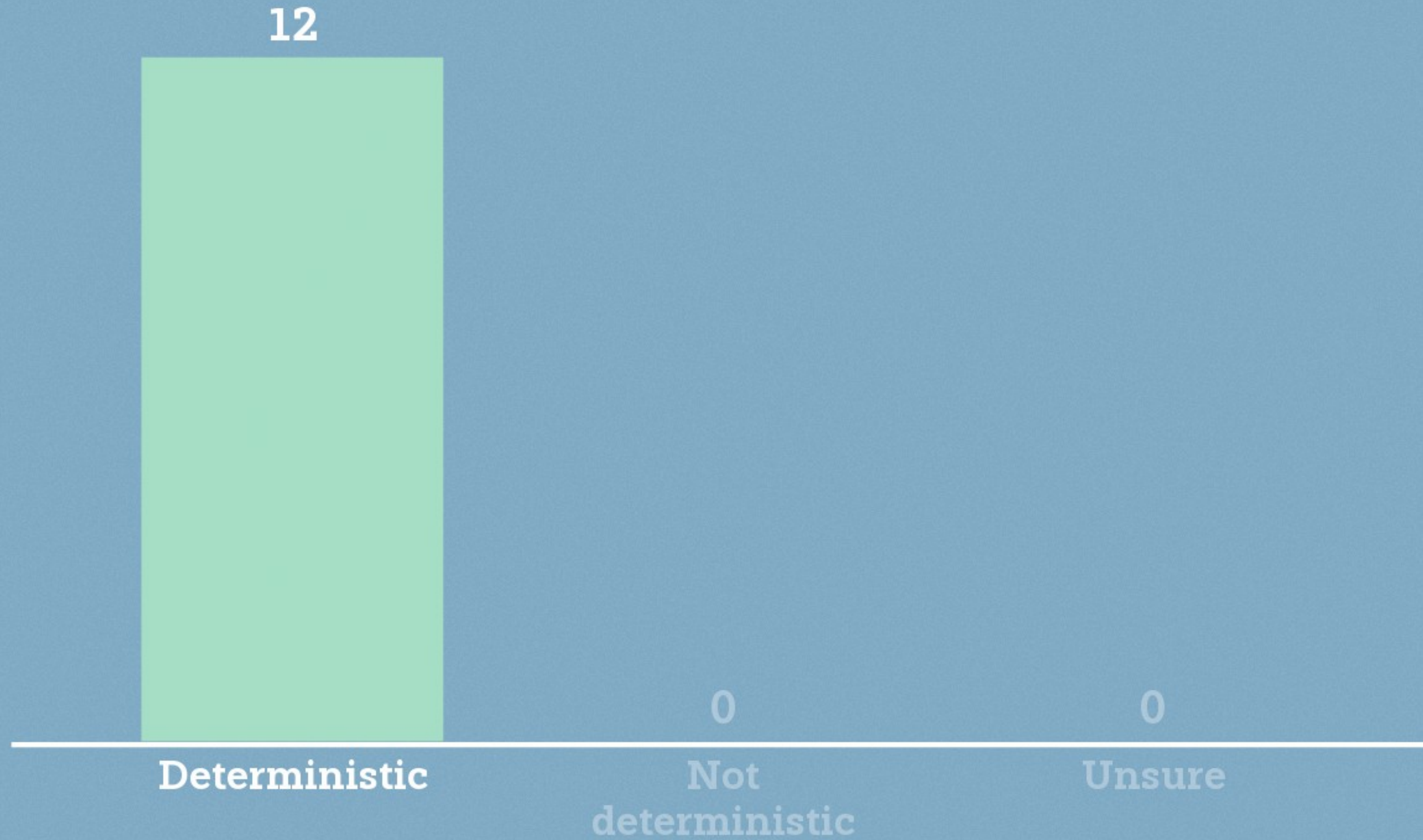
write[3] and read[0] become independent

If we leave out +VarAlpha there will be no synchronization.

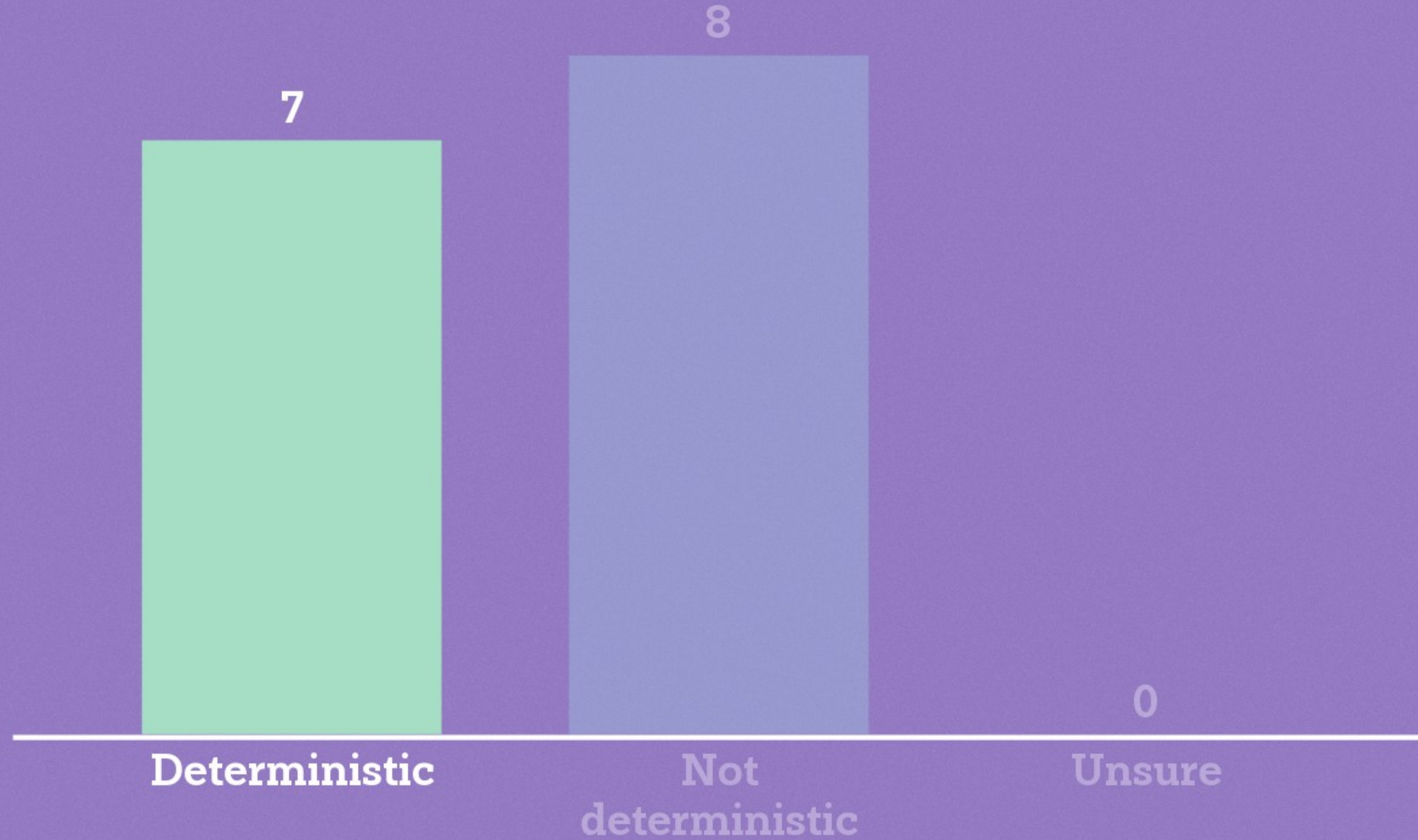


Demo: breaking the busy-wait protocol

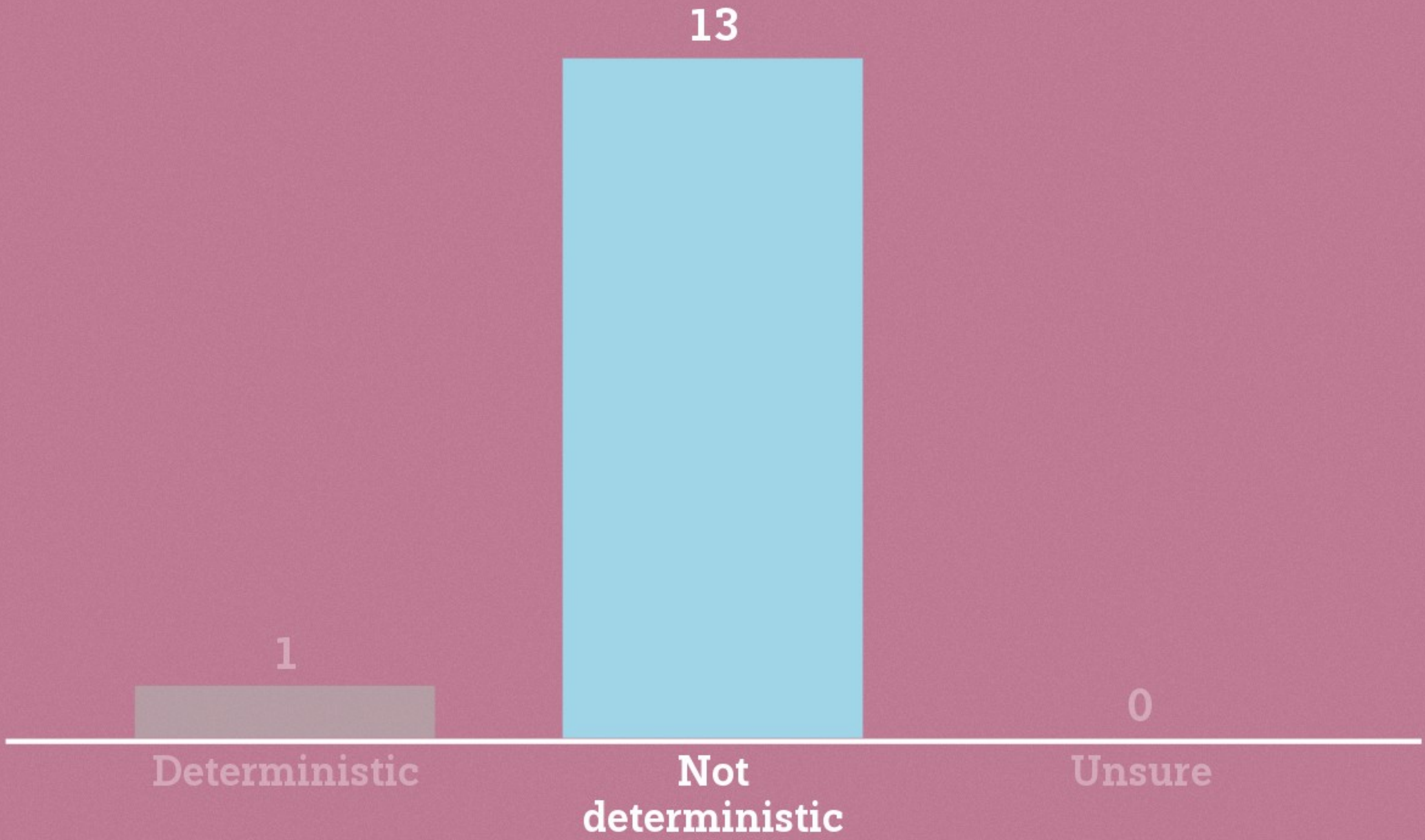
property $P = (a \rightarrow b \rightarrow P)$.



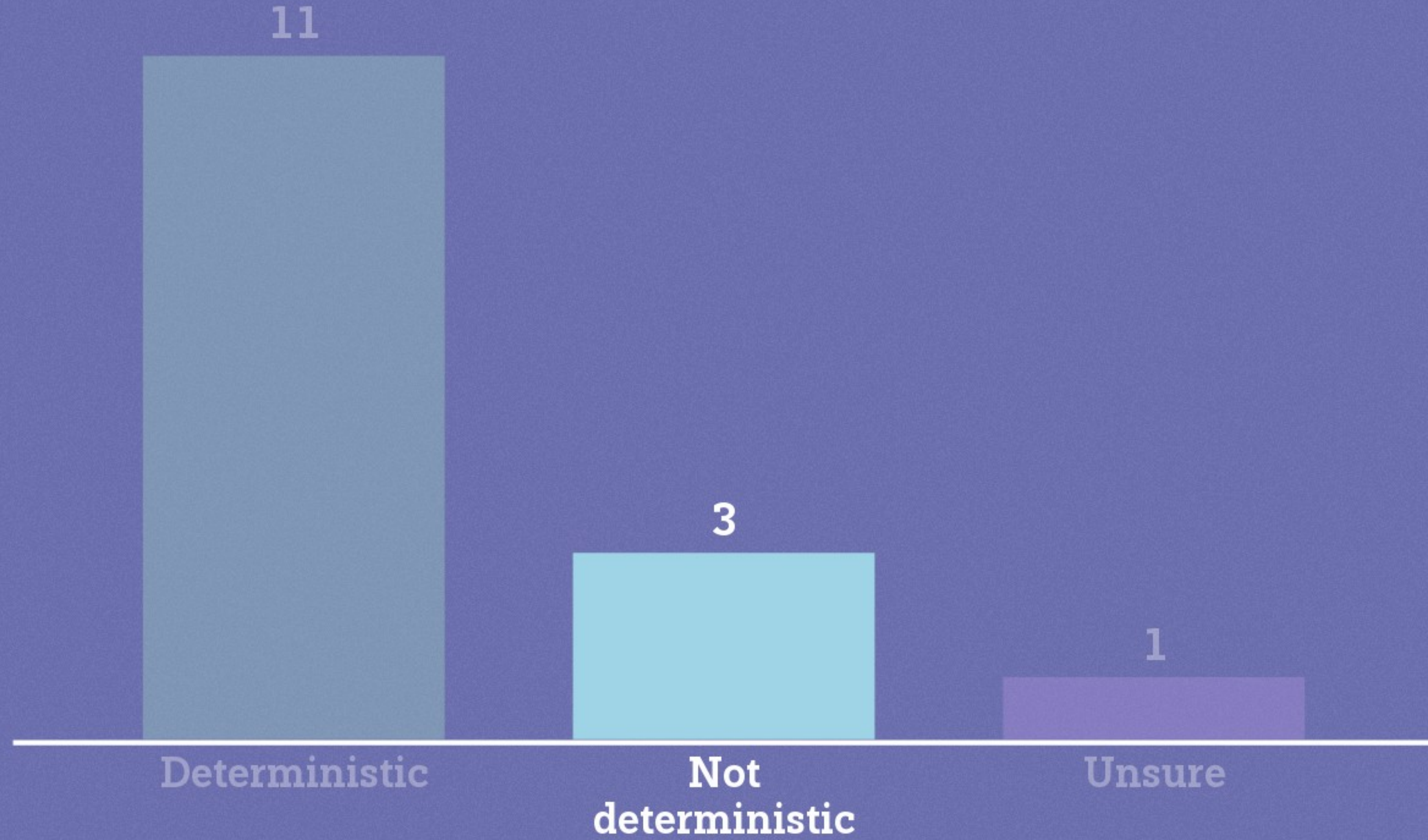
property P = (a → b → P | b → c → P).



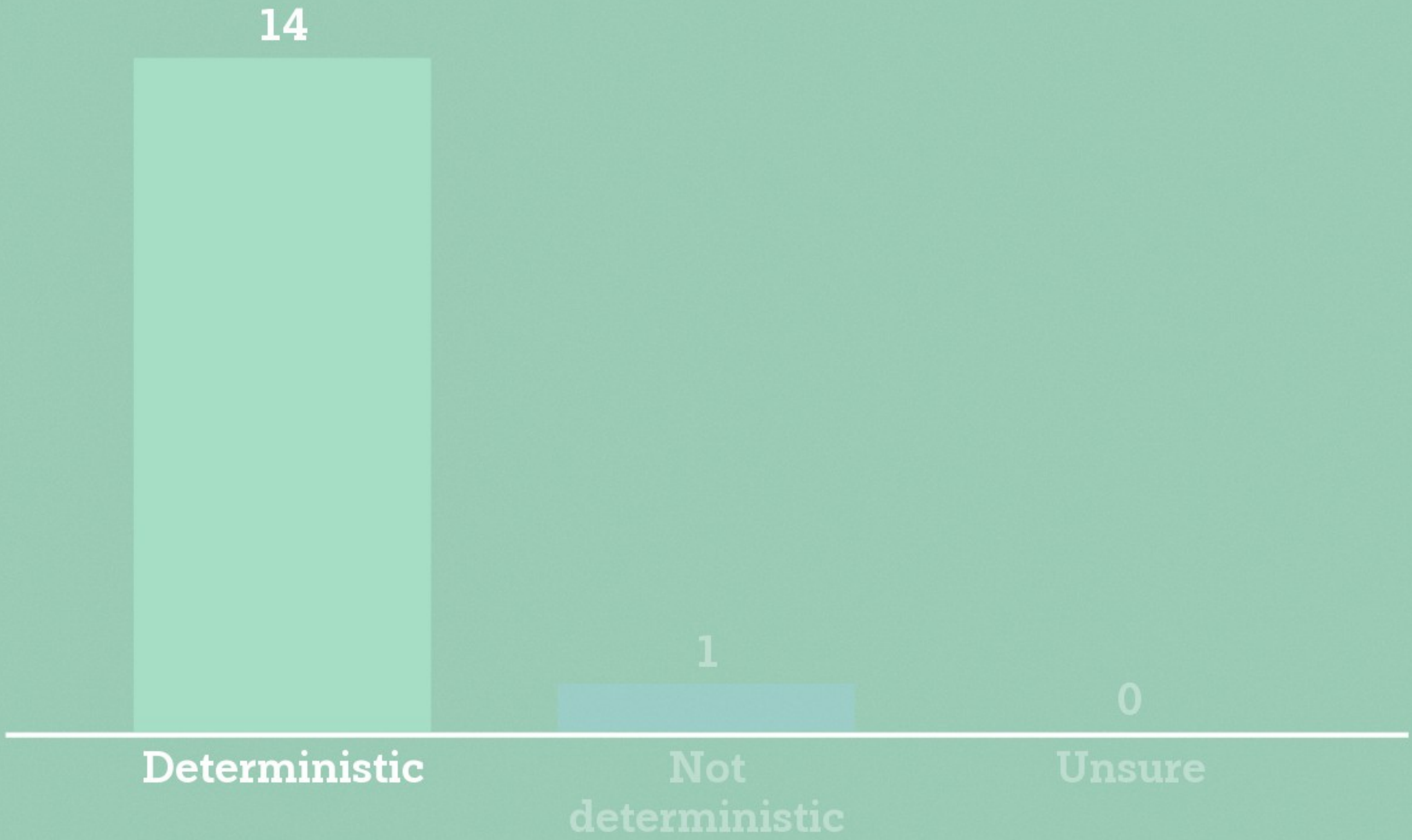
property P = (a -> b -> P | a -> c -> P).



property P = (a -> b -> P | a -> b -> P).



property P = (a → a → P | b → b → P).



Last chance for questions