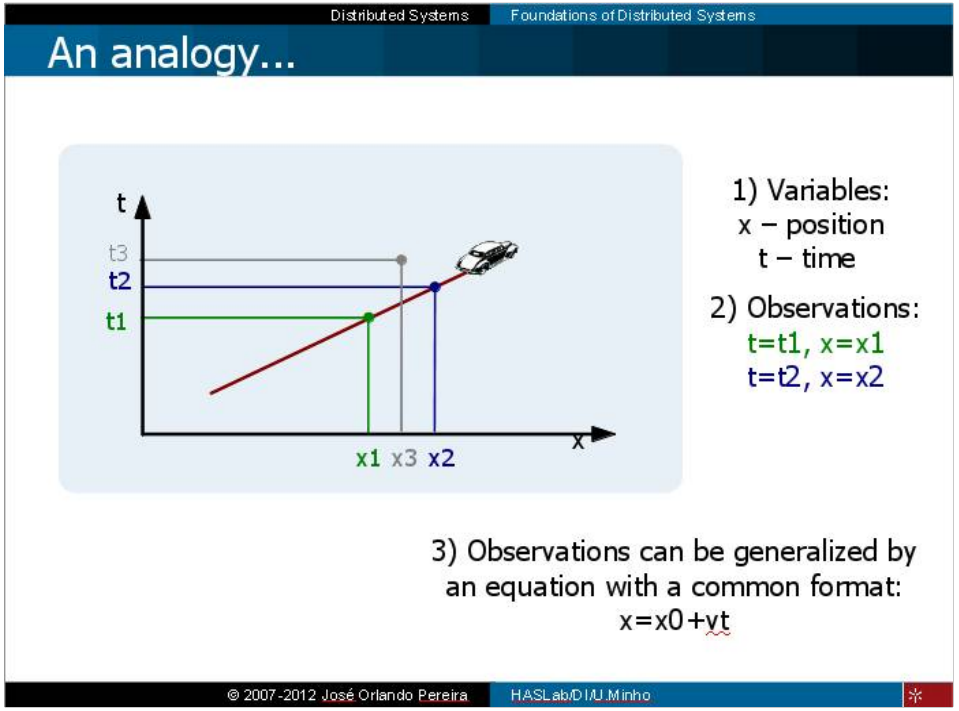
Case study: Trading system

- Input: offer to buy or sell something
- No output until a suitable buyer/seller pair is found
  - The trading system never holds stock
- Output: confirmation

Case study: Trading system

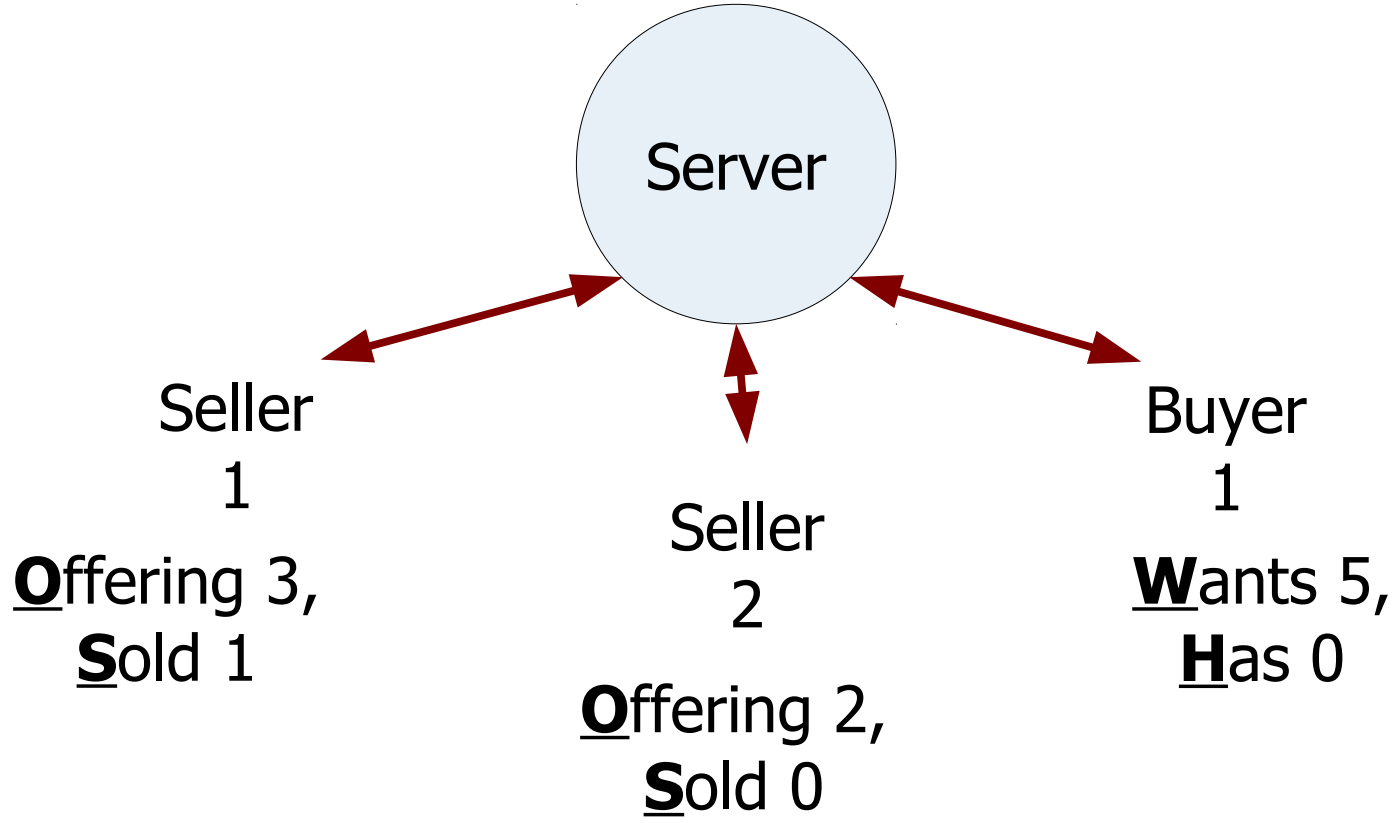
- What matters:
  - We don't buy/sell more than what has been offered/requested
  - If there are sellers and buyers for at least *k* items, eventually *k* items are sold and bought
  - If multiple buyers/sellers are competing, make sure no one is left behind
- What doesn't matter:
  - How the server is implemented
  - If there is a server at all...
  - ...

Step 1



- Select variables, observe and record...
- Major pitfall:
  - Programming variables vs. Model variables

Distributed state



Variable	O1	S1	O2	S2	W1	H1
Value	...	...	...	...	...	...
Value	3	1	2	0	5	0
Value	...	...	...	...	...	...

Distributed trace

- Sequence of states:

O1	S1	O2	S2	W1	H1
3	1	2	0	5	0
3	1	2	**1**	5	0
3	1	2	1	5	**1**
3	1	2	1	5	**2**
3	**2**	2	1	5	2
4	2	2	1	5	2
...	...	...	...	...	...

- Sequence of actions:

s2 sells 1

b1 buys 1

b1 buys 1

s1 sells 1

s1 offers 1

...

Distributed trace

- What actions are atomic?

O1	S1	O2	S2	W1	H1
3	1	2	0	5	0
3	1	2	1	5	0
4	1	2	1	5	**1**
4	1	2	1	5	2
…	…	…	…	…	…

Distributed trace

- How are alternative actions ordered?

O1	S1	O2	S2	W1	H1
3	1	2	0	5	0
3	1	2	**1**	5	0
3	1	2	1	5	**1**
3	1	2	1	5	**2**
3	**2**	2	1	5	2
4	2	2	1	5	2
...	...	...	...	...	...

O1	S1	O2	S2	W1	H1
3	1	2	0	5	0
3	1	2	**1**	5	0
3	1	2	1	5	**1**
3	**2**	2	1	5	1
3	2	2	1	5	**2**
4	2	2	1	5	2
...	...	...	...	...	...

Conclusion

- Our specifications are sets of possible traces
- Assuming a distributed system means that:
  - All possible orderings of events have to be considered
  - Large sets are required even for simple specifications
- Not necessarily a problem:
  - Recall: x = x0 + vt describes infinite "traces"

Step 2

- What can be observed?

An invalid trace

O1	S1	O2	S2	W1	H1
3	1	2	0	5	0
3	1	2	1	5	0
3	1	2	1	5	1
3	1	2	0	5	1
3	2	2	1	5	2
4	2	2	1	5	2
...	...	...	...	...	...

An invalid trace

- No action can explain this step:

O1	S1	O2	S2	W1	H1
3	1	2	0	5	0
3	1	2	1	5	0
3	1	2	**1**	5	1
3	1	2	**0**	5	1
3	2	2	1	5	2
4	2	2	1	5	2
…	…	…	…	…	…

An invalid trace

O1	S1	O2	S2	W1	H1
3	1	4	0	5	0
3	1	4	4	5	0
3	1	4	4	5	1
3	1	4	4	5	1
3	2	4	4	5	2
4	2	4	4	5	2
...	...	...	...	...	...

An invalid trace

- No previous sequence of actions can explain this state:

O1	S1	O2	S2	W1	H1
3	1	4	0	5	0
3	1	4	4	5	0
3	1	4	4	5	1
3	1	4	4	5	1
3	**2**	**4**	**4**	**5**	**2**
4	2	4	4	5	2
…	…	…	…	…	…

Safety property

- "Nothing bad ever happens..."
- Can be identified on a finite prefix of the trace

An invalid trace

O1	S1	O2	S2	W1	H1
1	0	1	0	1	0
1	1	1	0	1	0
1	1	1	0	1	1
1	1	1	0	2	1
1	1	1	0	2	1
1	1	1	0	2	1
...	...	...	...	...	...

An invalid trace

- Deadlock? Must see full trace...

O1	S1	O2	S2	W1	H1
1	0	1	0	1	0
1	1	1	0	1	0
1	1	1	0	1	1
1	1	1	0	2	1
1	1	1	0	2	1
1	1	1	0	2	1
...	...	...	...	...	...

An invalid trace

O1	S1	O2	S2	W1	H1
1	0	1	0	1	0
1	0	1	0	1	1
1	1	1	0	1	1
2	1	1	0	1	1
2	1	1	0	2	1
2	1	1	0	2	2
2	2	1	0	2	2
3	2	1	0	2	2
3	2	1	0	3	2
3	2	1	0	3	3
3	3	1	0	3	3
...	...	...	...	...	...

An invalid trace

- Does Seller 2 get a fair chance?
- Must see full trace..

O1	S1	O2	S2	W1	H1
1	0	1	0	1	0
1	0	1	0	1	1
1	1	1	0	1	1
2	1	1	0	1	1
2	1	1	0	2	1
2	1	1	0	2	2
2	2	1	0	2	2
3	2	1	0	2	2
3	2	1	0	3	2
3	2	1	0	3	3
3	3	1	0	3	3
...	...	...	...	...	...

Liveness

- "Something good eventually happens..."
- Cannot be identified on a finite prefix of the trace

Specification = Safety + Liveness

- Our specifications can always be decomposed in:
  - Safety
    +
  - Liveness
- How to write them down?
- How to derive them?

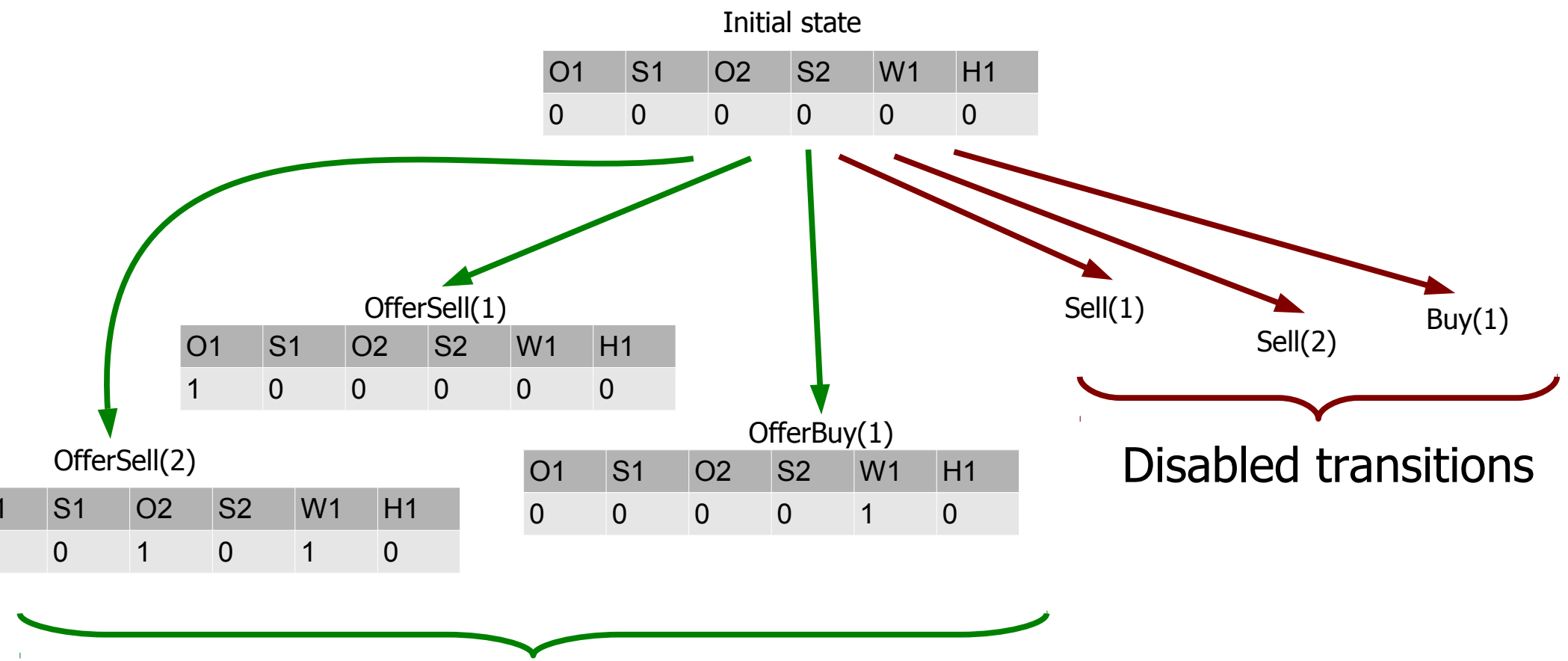
Roadmap

- Recall:
  - A specification is a set of traces
- How to write a compact expression that generates such set?
- How to determine if such specification (in compact format) satisfies safety and liveness properties?

State machine

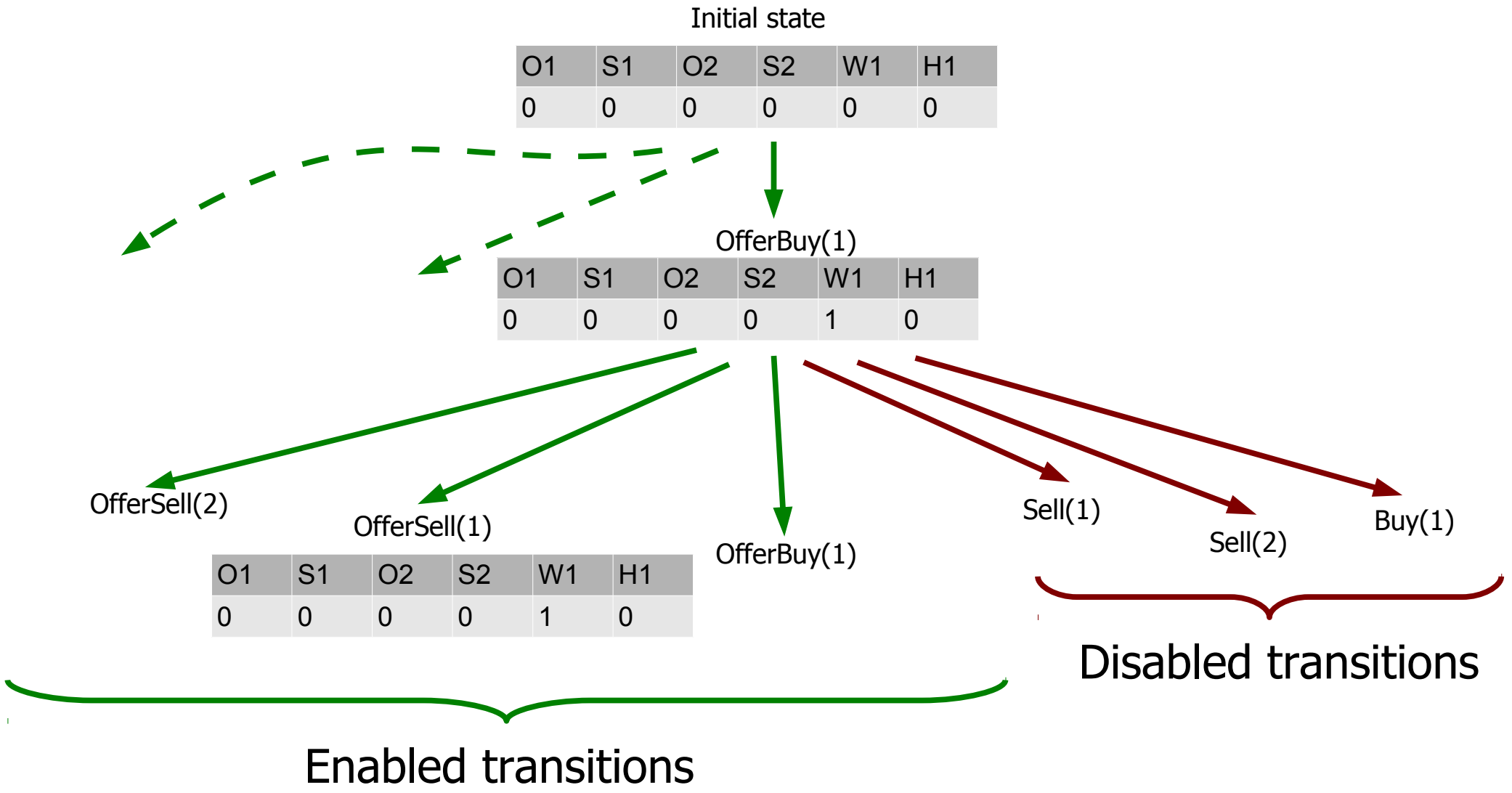
- BuyOffer(i):
  - Pre-condition:
    - True
  - Effect:
    - $O_i := O_i+1$
- Buy(i):
  - Pre-condition:
    - $\Sigma H<\min(\Sigma O,\Sigma W)$
    - $H_i<W_i$
  - Effect:
    - $H_i := H_i+1$
- SellOffer(i):
  - Pre-condition:
    - True
  - Effect:
    - $W_i := W_i+1$
- Sell(i):
  - Pre-condition:
    - $\Sigma S<\min(\Sigma O,\Sigma W)$
    - $S_i<O_i$
  - Effect:
    - $S_i := S_i+1$

Safe traces

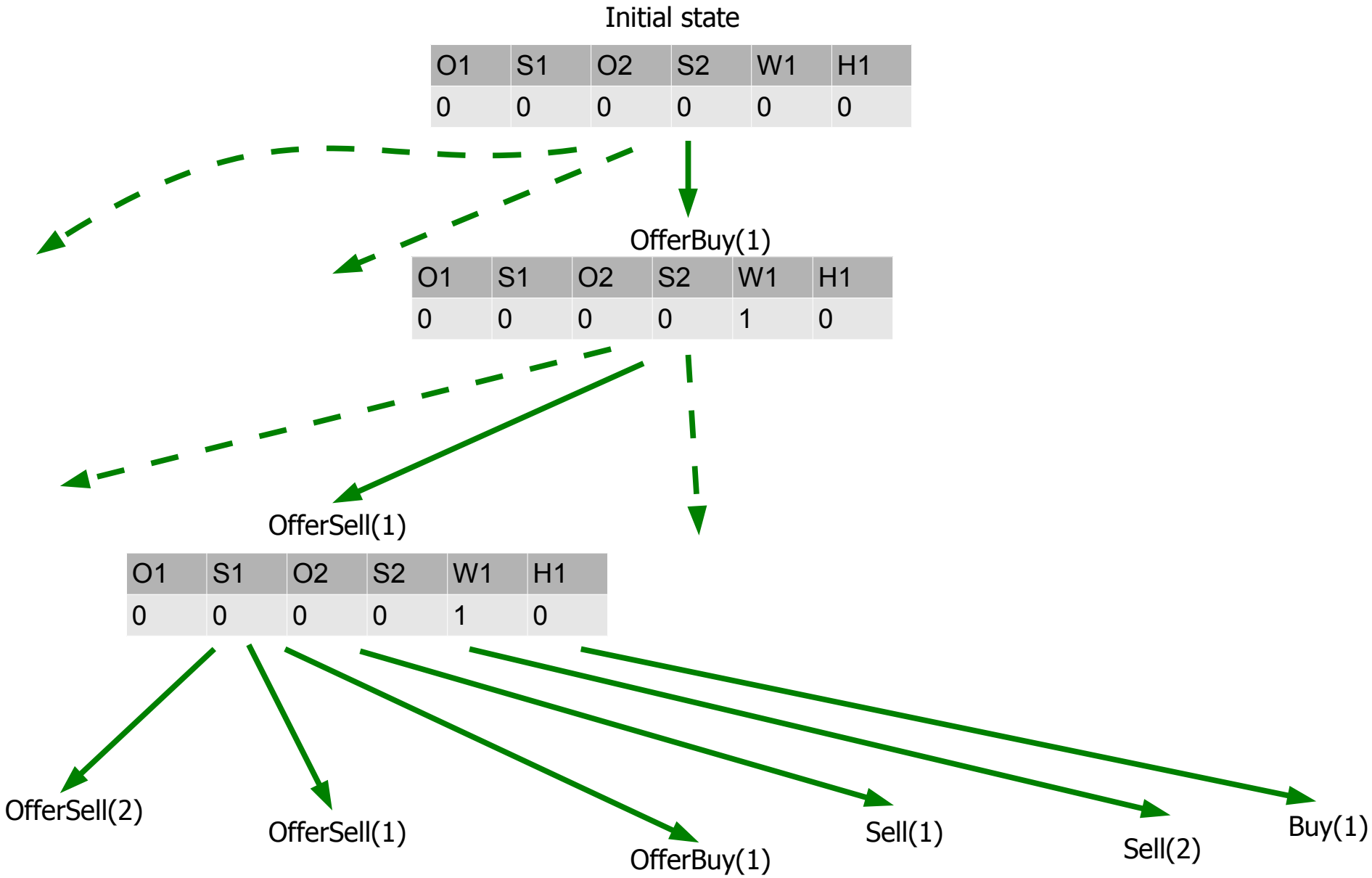


Three new safe traces found!

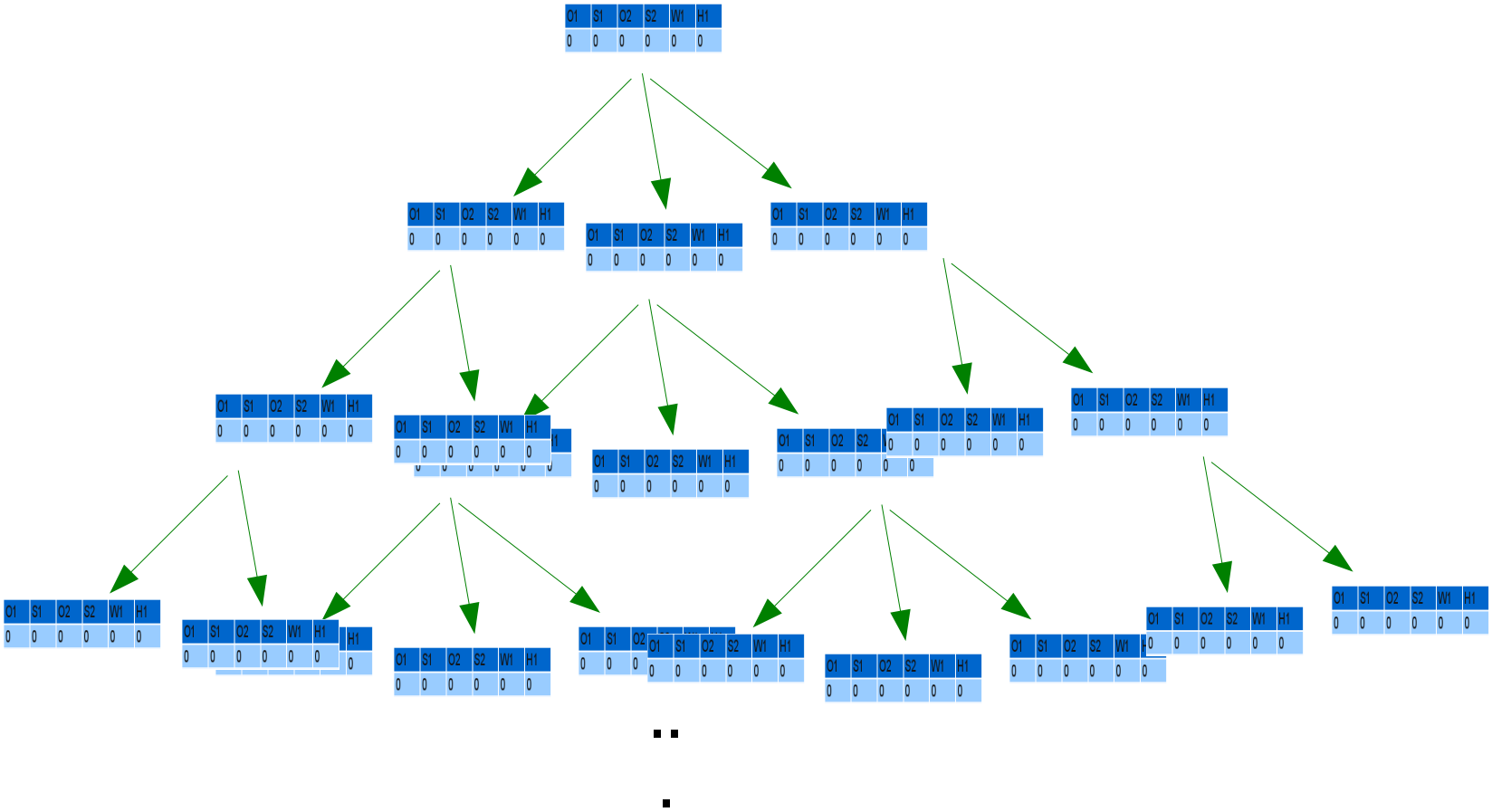
Safe traces


Initial state
O1 S1 O2 S2 W1 H1
0 0 0 0 0 0
OfferBuy(1)
O1 S1 O2 S2 W1 H1
0 0 0 0 1 0
OfferSell(2)
OfferSell(1)
O1 S1 O2 S2 W1 H1
0 0 0 0 1 0
OfferBuy(1)
Sell(1)
Sell(2)
Buy(1)
Disabled transitions
Enabled transitions

Safe traces


Initial state
O1 S1 O2 S2 W1 H1
0 0 0 0 0 0
OfferBuy(1)
O1 S1 O2 S2 W1 H1
0 0 0 0 1 0
OfferSell(1)
O1 S1 O2 S2 W1 H1
0 0 0 0 1 0
OfferSell(2)
OfferSell(1)
OfferBuy(1)
Sell(1)
Sell(2)
Buy(1)

Safe traces



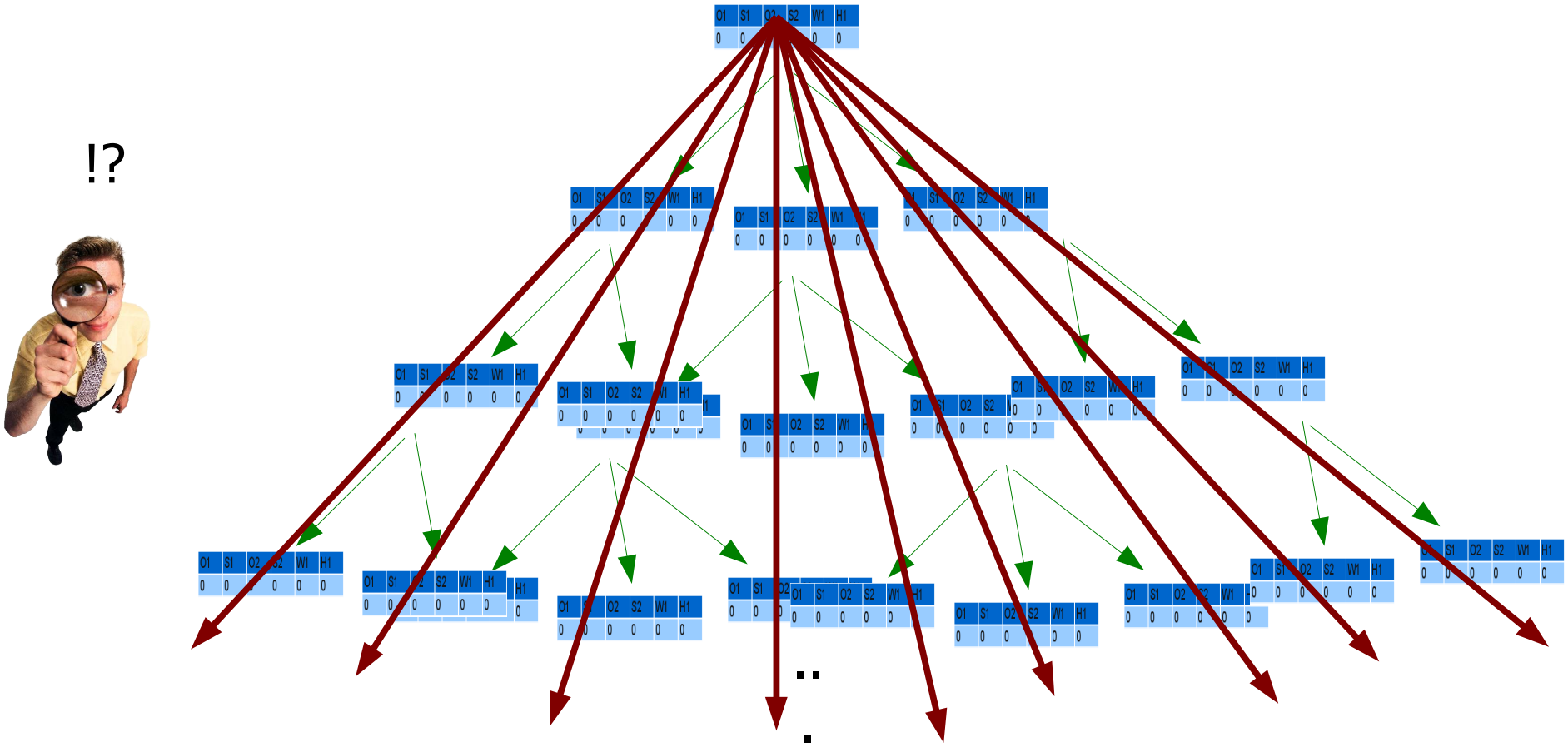
Safe traces

- Start with one trace for each initial state
- Repeat:
  - For any known safe trace
  - For any enabled transition
  - Append resulting state
  - Add new trace to set
- Most likely an infinite set...

Safety properties

- Does this specification meet the desired properties?
- Safety property:
  - We don't buy more that what has been offered (i.e. $\Sigma H \leq \Sigma O$)
- Observation:
  - This is a state invariant

Safety properties



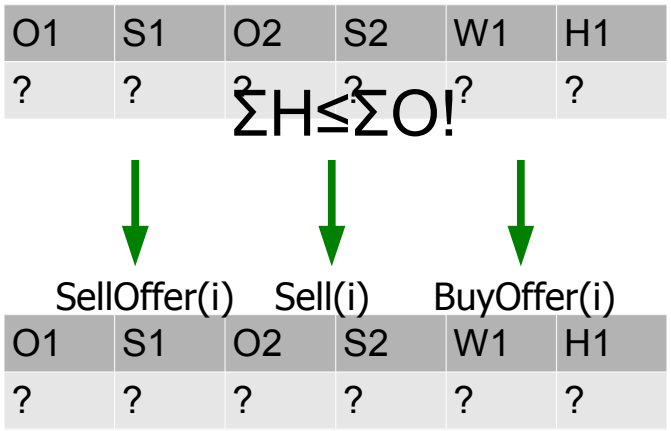
- Cannot evaluate all possible traces, but...

Base step

O1	S1	O2	S2	W1	H1
0	0	0	0	0	0

$\Sigma H \leq \Sigma O$? Yes!

Induction step



ΣH≤ΣO? Yes!

(all H unchanged, O never decreases)

Induction step

O1	S1	O2	S2	W1	H1
?	?	?	?	?	?

$\Sigma H \leq \Sigma O$!

Buy(i)

O1	S1	O2	S2	W1	H1
?	?	?	?	?	?+1

$\Sigma H \leq \Sigma O$?

Induction step

- We know:
  - Induction
    - $\Sigma H \leq \Sigma O$
  - Pre-condition
    - $\Sigma H < \min(\Sigma O, \Sigma W)$
      - $\Sigma H < \Sigma O$
      - $\Sigma H < \Sigma W$
    - $H_i < W_i$
  - Effect
    - $H_i := H_i + 1$
    - Everything else unchanged
- We want to know if:
  - **$\Sigma H \leq \Sigma O$**

Safety properties

O1	S1	O2	S2	W1	H1
0	0	0	0	0	0

O1	S1	O2	S2	W1	H1
?	?	?	?	?	?

$\sum H \leq \sum O !$

SellOffer(i) Sell(i) BuyOffer(i)

O1	S1	O2	S2	W1	H1
?	?	?	?	?	?

O1	S1	O2	S2	W1	H1
?	?	?	?	?	?

$\sum H \leq \sum O !$

Buy(i)

O1	S1	O2	S2	W1	H1
?	?	?	?	?	?+1

- Can easily inspect:
  - All initial states
  - All transitions (many are trivial)

Safety properties

- A state invariant can proved by induction
- Other safety properties can be translated into state invariants by:
  - Strengthening the property
  - Recording past trace as state