A Petri net-based notation for normative modeling: *evaluation on deontic paradoxes*

16 June 2017, MIREL workshop @ ICAIL

Giovanni Sileno[2,1] gsileno@enst.fr
Alexander Boer[1], **Tom van Engers**[1]

[1] *Leibniz Center for Law, University of Amsterdam*
[2] *Télécom ParisTech, Université Paris-Saclay, Paris*

General Research Problem

- Aligning Law and Action

General Research Problem

- Aligning Law and Action
  - law concerns a **system of norms**, that in abstract, in a fixed point in time, may be approached atemporally.

General Research Problem

- Aligning Law and Action
  - law concerns a **system of norms**, that in abstract, in a fixed point in time, may be approached atemporally.
  - when applied, law deals with a **continuous flow of events**.

General Research Problem

- Aligning Law and Action
  - law concerns a **system of norms**, that in abstract, in a fixed point in time, may be approached atemporally.
  - when applied, law deals with a **continuous flow of events**.
- Prototypical encounter: legal cases.

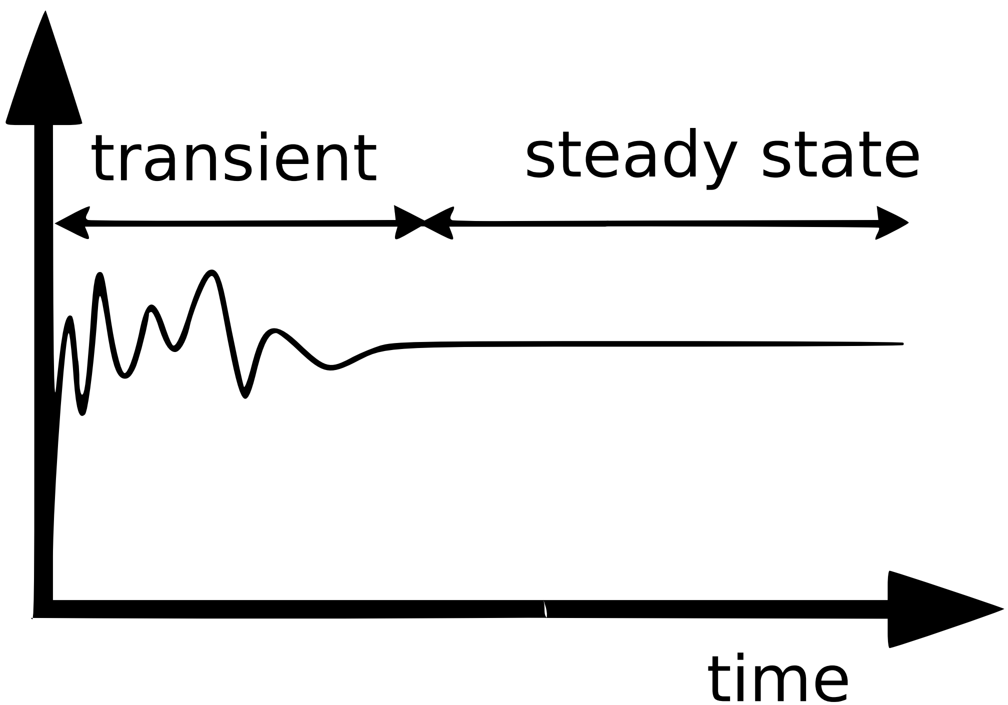
General Research Problem

- Aligning Law and Action
  - law concerns a **system of norms**, that in abstract, in a fixed point in time, may be approached atemporally.
  - when applied, law deals with a **continuous flow of events**.
- Prototypical encounter: legal cases.
- More general but similar problem: *narrative interpretation.*

Looking for a notation

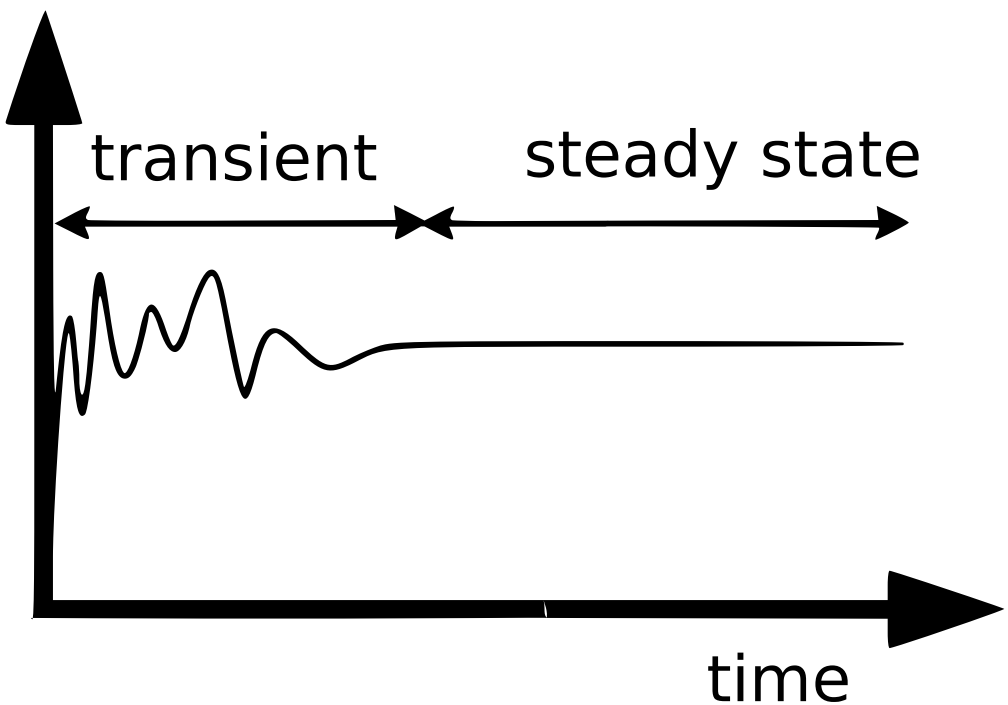
Steady states and transients

- Physical systems can be approached from steady state (equilibrium) or transient (non-equilibrium, dynamic) perspectives



Steady states and transients

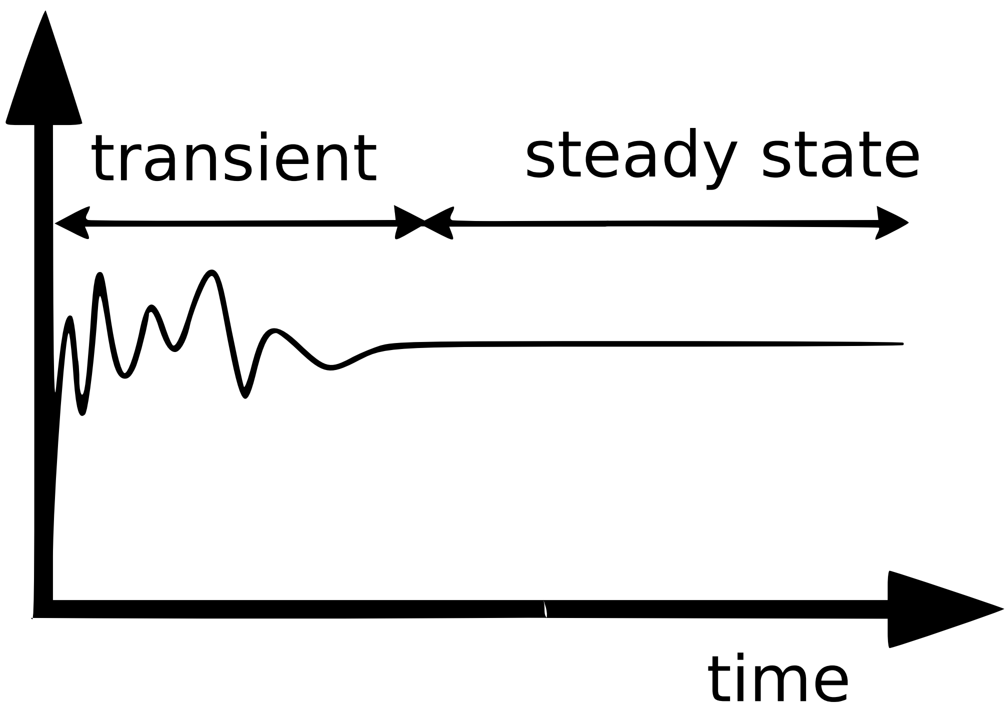
- Physical systems can be approached from steady state (equilibrium) or transient (non-equilibrium, dynamic) perspectives



- Steady states descriptions **omit** transient characteristics

Steady states and transients

- Physical systems can be approached from steady state (equilibrium) or transient (non-equilibrium, dynamic) perspectives



- Steady states descriptions **omit** transient characteristics

  ex. **Ohm's Law**
  V = R * I

Specifying transients and steady states

- Possible analogies:
  - *steady state* approach with
    - Logic
    - Declarative programming

focus on
What

Specifying transients and steady states

- Possible analogies:
  - *steady state* approach with
    - Logic
    - Declarative programming
  - *transient* approach
    - Process modeling
    - Procedural programming

focus on **What**

focus on **How**

Specifying transients and steady states

- Possible analogies:
  - *steady state* approach with
    - Logic
    - Declarative programming
  - *transient* approach
    - Process modeling
    - Procedural programming

focus on
What

focus on
How

Petri Nets!

Specifying transients and steady states

- Possible analogies:
  - *steady state* appro[ach]
    - Logic
    - Declarative programming
  - *transient* approach
    - Process modeling
    - Procedural programming

Answer Set Programming

focus on **What**

focus on **How**

Petri Nets!

Specifying transients and steady states

- Possible analogies:
  - *steady state* approach
    - Logic
    - Declarative programming
  - *transient* approach
    - Process modeling
    - Procedural programming

Answer Set Programming

focus on **What**

focus on **How**

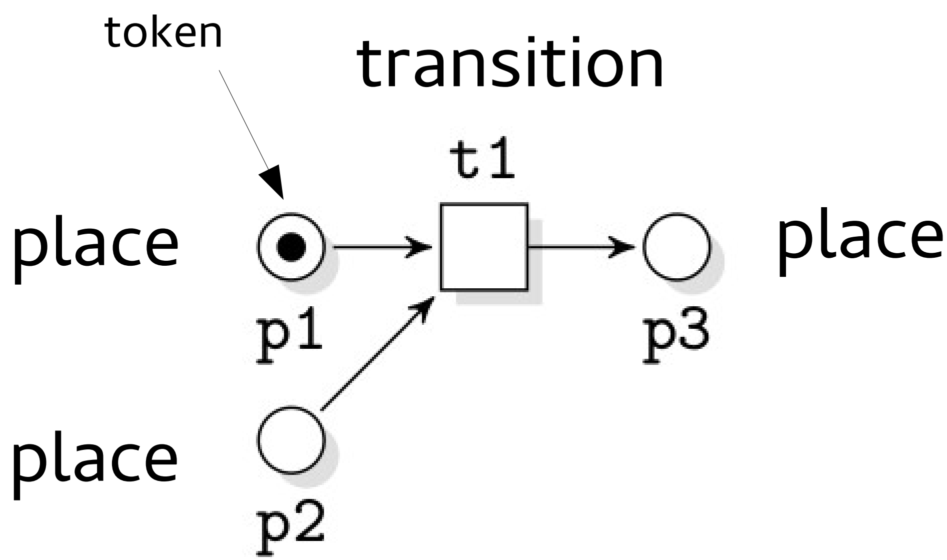
logic programming petri nets

→ LPPN

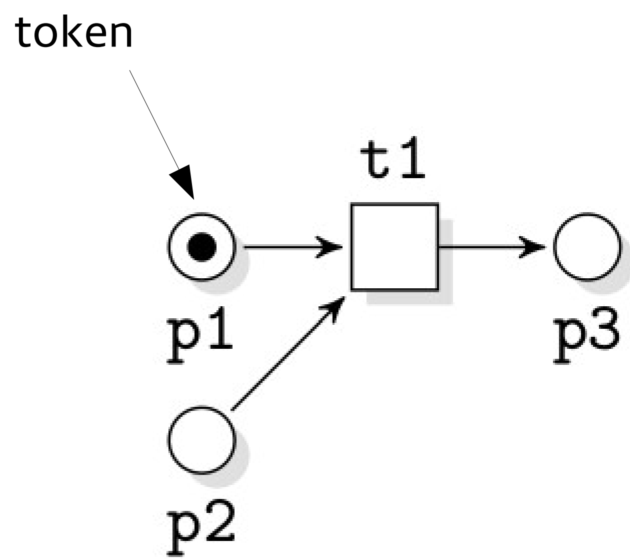
Petri Nets!

Logic Programming Petri Nets

procedural LPPN

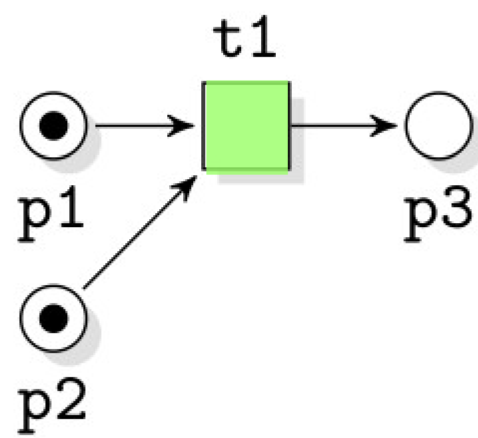


procedural LPPN



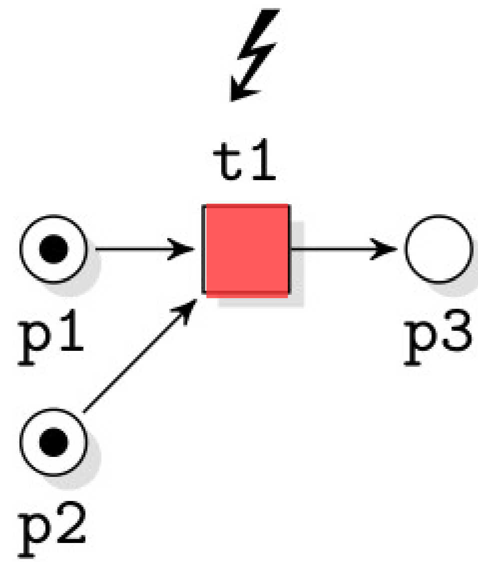
- *token game*: transition disabled. It cannot fire.

procedural LPPN



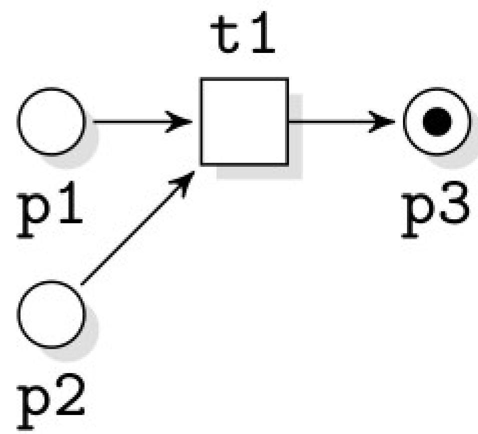
- *token game*: transition enabled. It can fire.

procedural LPPN



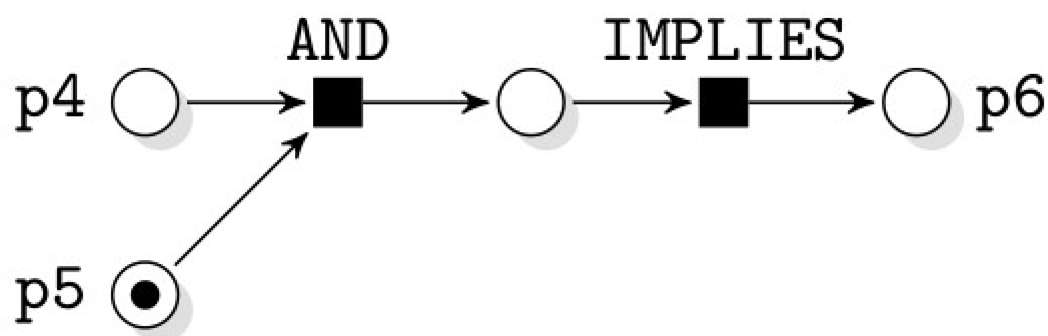
- *token game*: transition fires: it will consume tokens from the input places.

procedural LPPN



- *token game*: ...and produce tokens in the outputs.

declarative LPPN for places

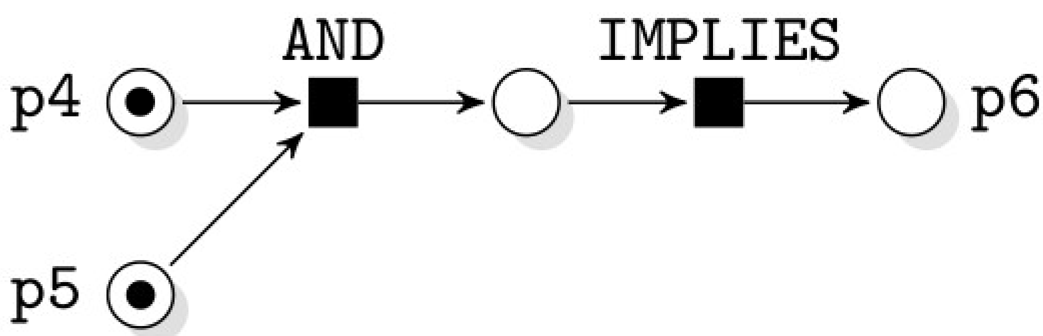


- Equivalent to

  ASP/Prolog: `p6 :- p4, p5. p5.`

  FOL: $(p4 \wedge p5 \rightarrow p6) \wedge p5$

declarative LPPN for places

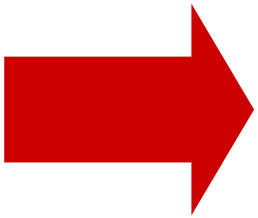


- Equivalent to

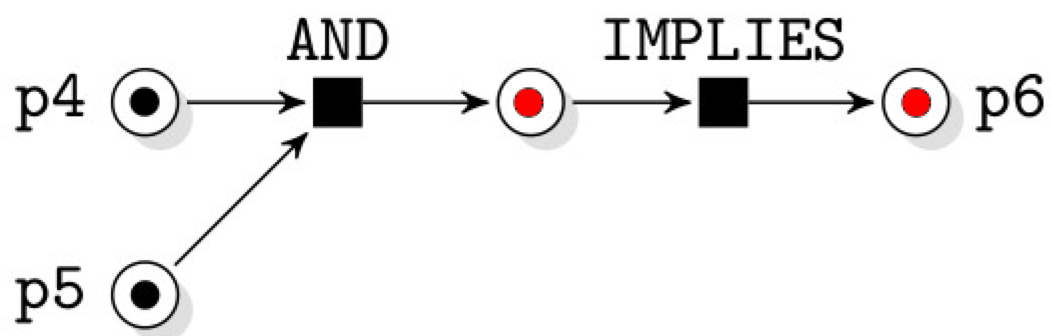
  ASP/Prolog: `p6 :- p4, p5. p5. p4.`

  FOL: $(p4 \land p5 \rightarrow p6) \land p5 \land p4$

entails



declarative LPPN for places



- Equivalent to

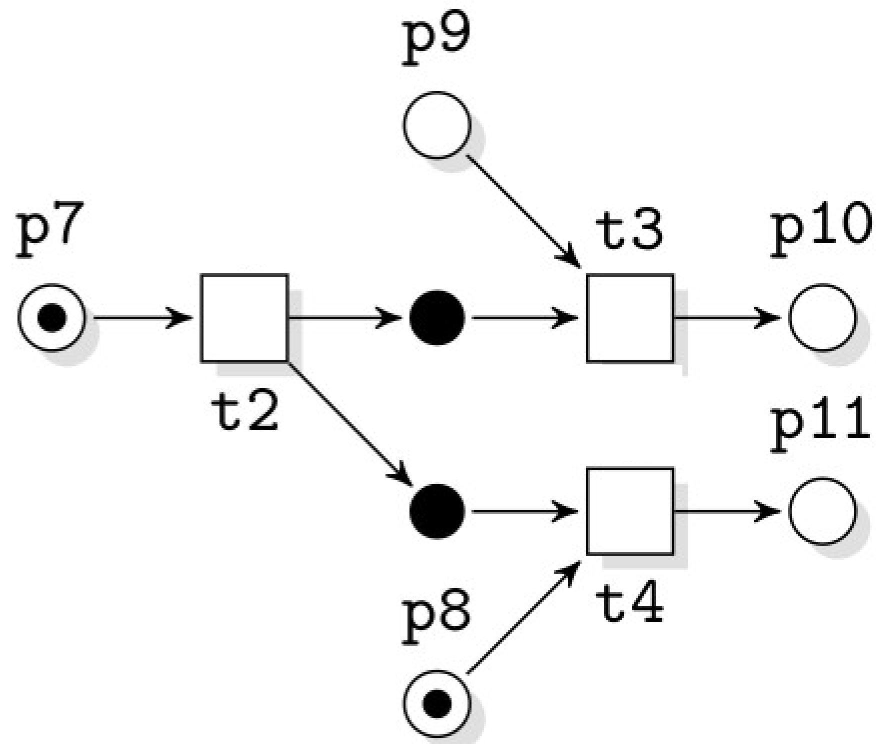
  ASP/Prolog: `p6 :- p4, p5. p5. p4.`

  FOL: $(p4 \wedge p5 \rightarrow p6) \wedge p5 \wedge p4$

entails

p6

declarative LPPN for transitions

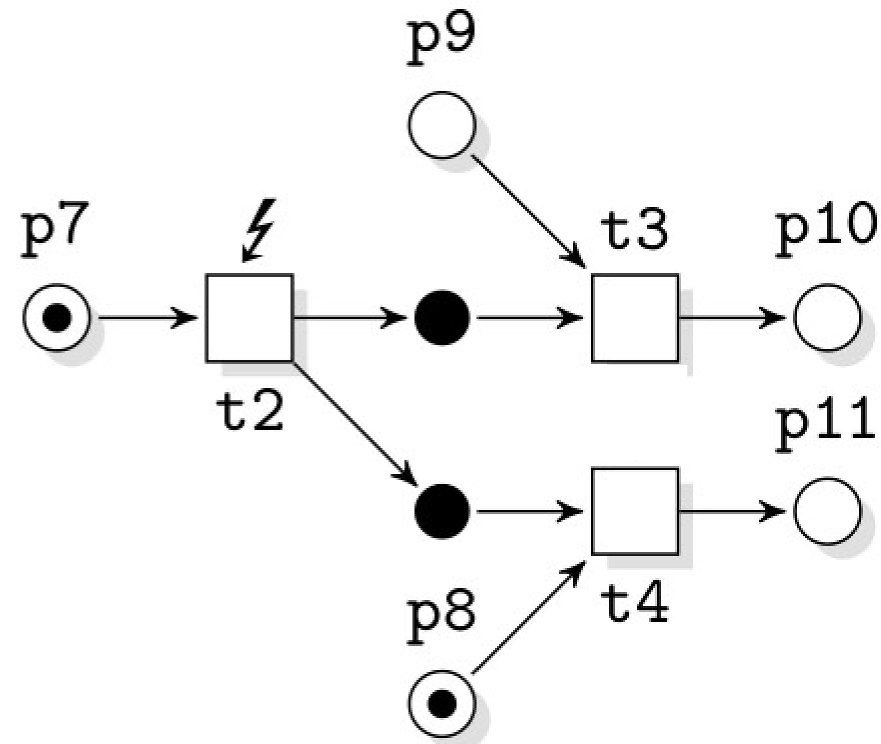


- Equivalent to

  ASP/Prolog: `t4 :- t2, p8. t3 :- t2, p9.`

  FOL: `(t2 ∧ p8 → t4) ∧ (t2 ∧ p9 → t4)`

declarative LPPN for transitions



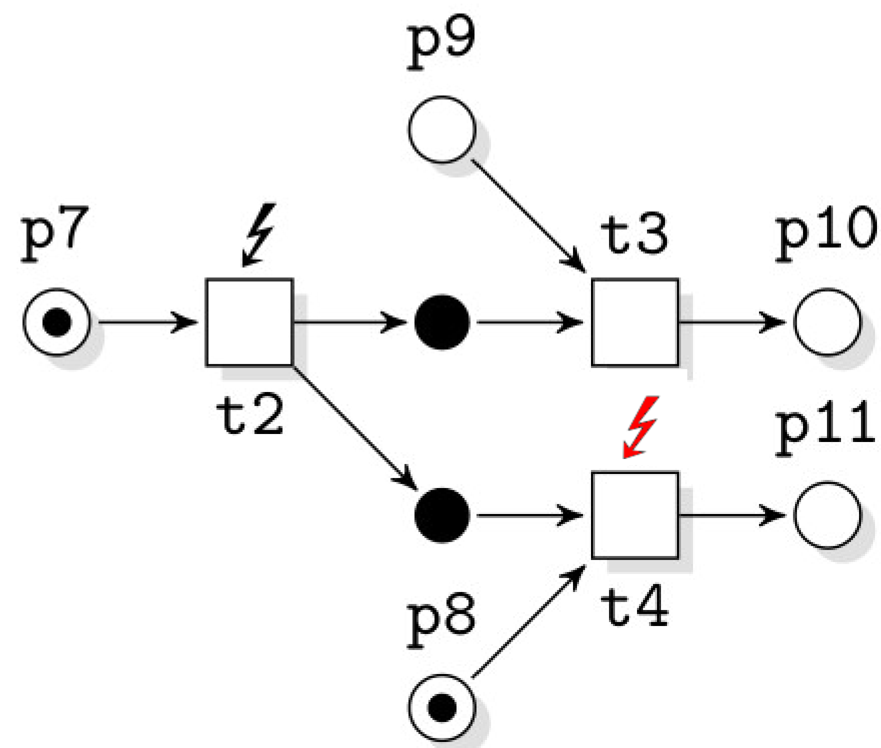
entails

- Equivalent to

  ASP/Prolog: `t4 :- t2, p8. t3 :- t2, p9. t2.`

  FOL: `(t2 ∧ p8 → t4) ∧ (t2 ∧ p9 → t4) ∧ t2`

declarative LPPN for transitions



t4

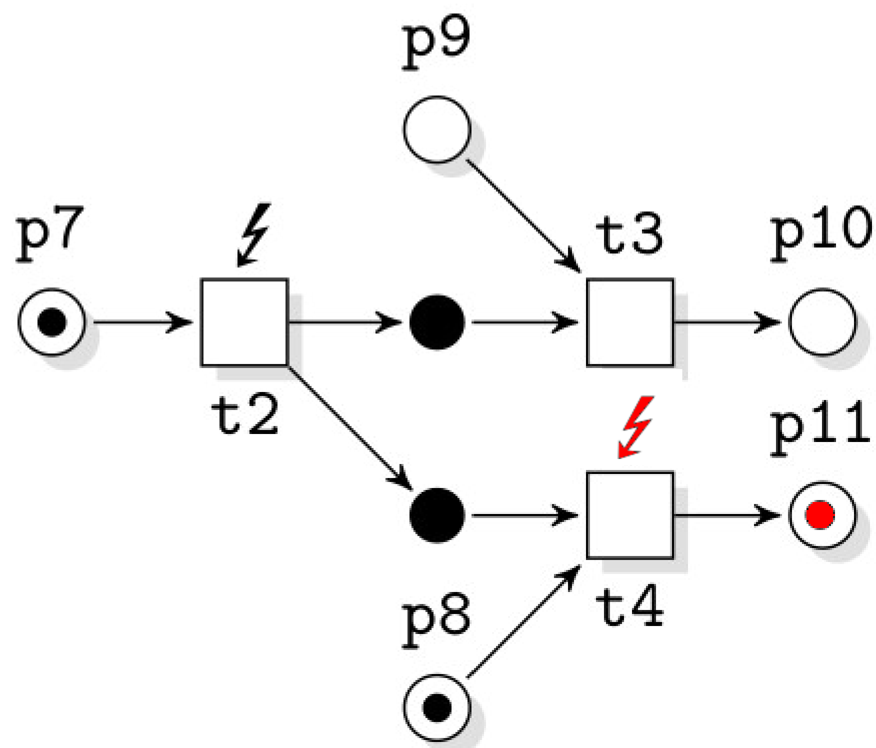
entails

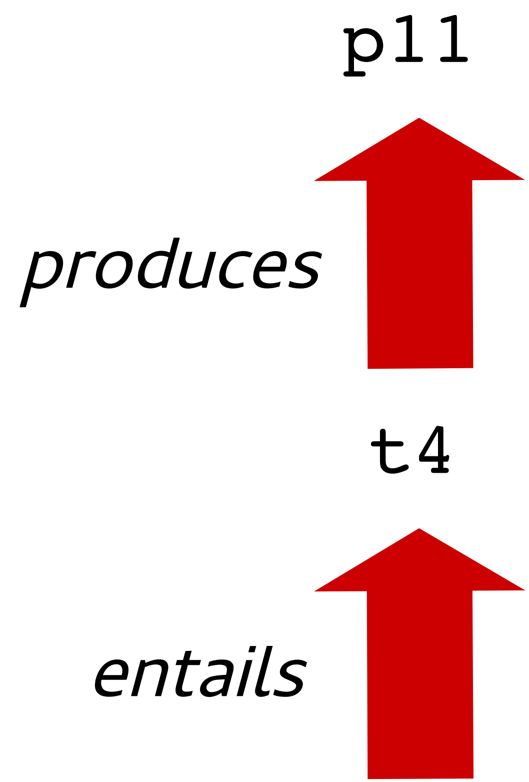
- Equivalent to

  ASP/Prolog: `t4 :- t2, p8. t3 :- t2, p9. t2.`

  FOL: `(t2 ∧ p8 → t4) ∧ (t2 ∧ p9 → t4) ∧ t2`

declarative LPPN for transitions





- Equivalent to

  ASP/Prolog: `t4 :- t2, p8. t3 :- t2, p9. t2.`

  FOL: $(t2 \wedge p8 \rightarrow t4) \wedge (t2 \wedge p9 \rightarrow t4) \wedge t2$

Evaluating the LPPN notation

Evaluating the LPPN notation

- Normative modeling is one of the purposes of the notation. We used it to model well-known problems defined by the deontic logic community.

Evaluating the LPPN notation

- Normative modeling is one of the purposes of the notation. We used it to model well-known problems defined by the deontic logic community.
- Contrary To Duty (CTD) structures:

  situations in which a primary obligation exists, and with its violation, a secondary obligation comes into existence.

Evaluating the LPPN notation

- Normative modeling is one of the purposes of the notation. We used it to model well-known problems defined by the deontic logic community.
- Contrary To Duty (CTD) structures:

  situations in which a primary obligation exists, and with its violation, a secondary obligation comes into existence.

- Common in e.g. compensatory norms.

A simple deontic paradox

- You are forbidden to cross the road.
- If you are crossing the road,
  (you have to) cross the road!

A simple deontic paradox

- You are forbidden to cross the road.
- If you are crossing the road,
  (you have to) cross the road!
- You are crossing.

A simple deontic paradox

- You are forbidden to cross the road.
- If you are crossing the road,
  (you have to) cross the road!
- You are crossing.

```
obl(not cross)
if cross then obl(cross)
cross
```

```
obl(not cross)
obl(cross)
```

contradiction

You mustn't cross the road.

If you are crossing the road,
cross the road!

- Usual solution: **preferences** amonst possible worlds.

You mustn't cross the road.

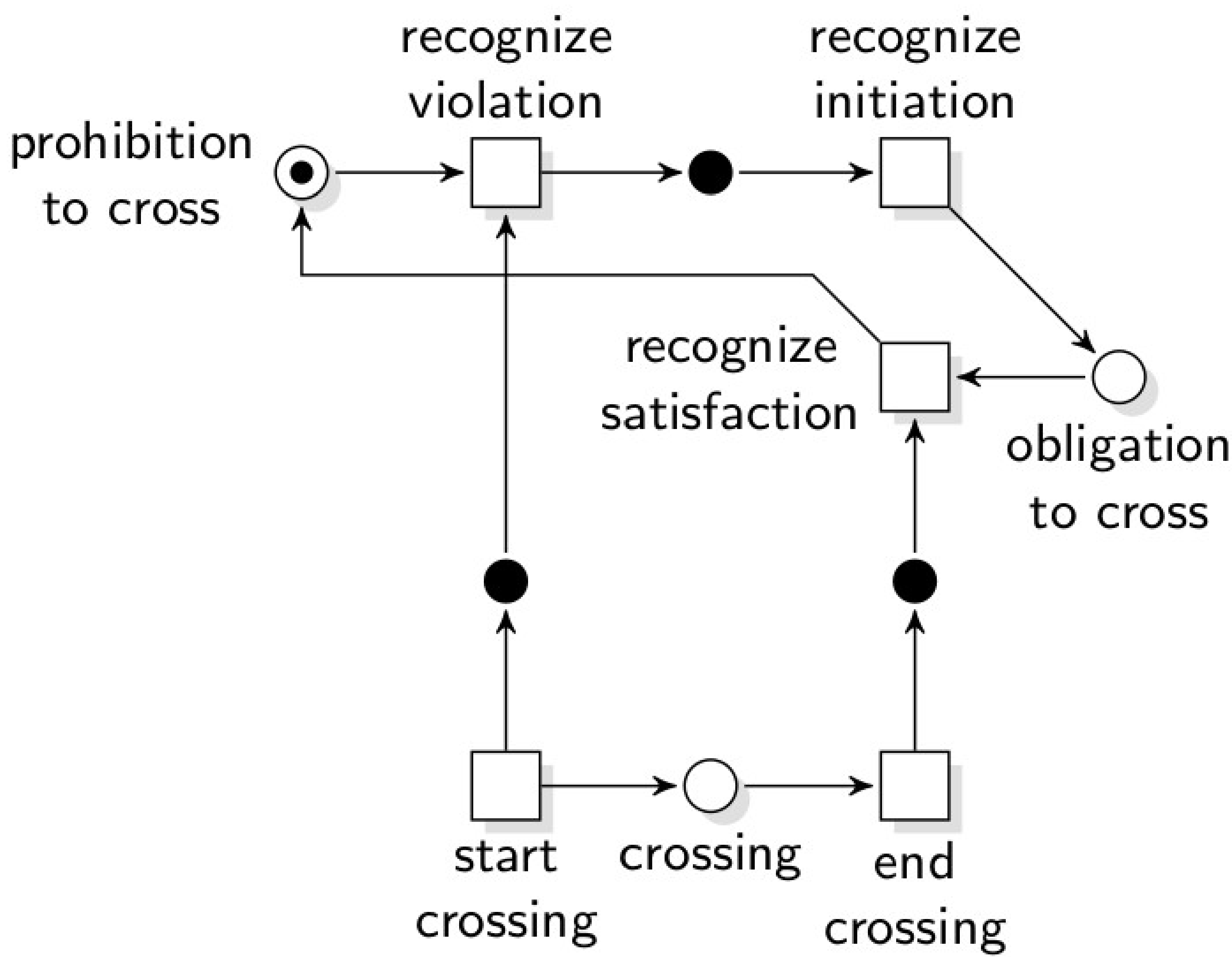
If you are crossing the road,
cross the road!

- Usual solution: **preferences** amonst possible worlds.
- Alternative solution:
  - implicit temporal meaning: *if you started crossing the road, finish to cross the road!*

→ Let us increase granularity of the model...

You mustn't cross the road. **Version 1**

If you are crossing the road,
cross the road!

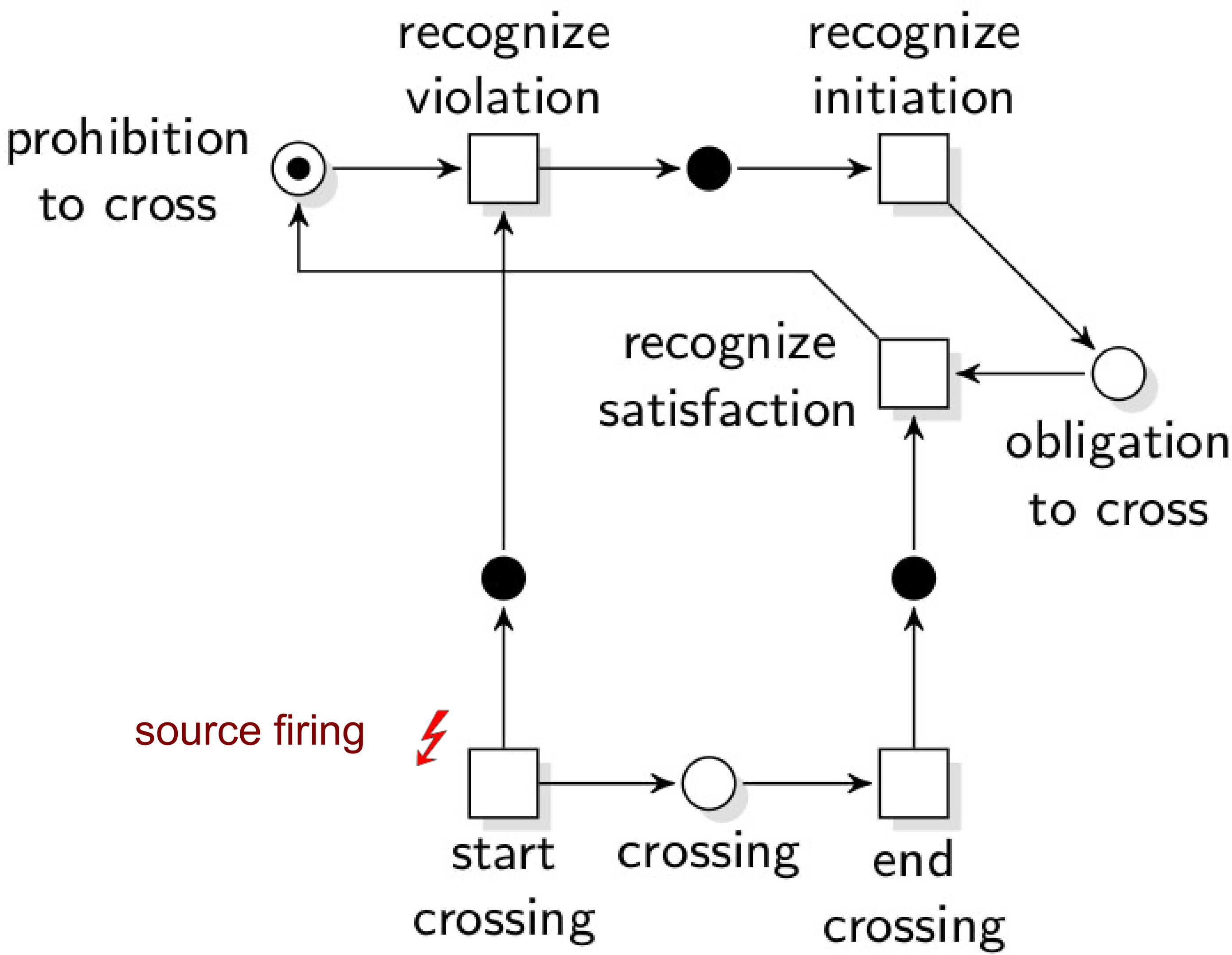


You mustn't cross the road.

If you are crossing the road, cross the road!

Version 1

- You start crossing.

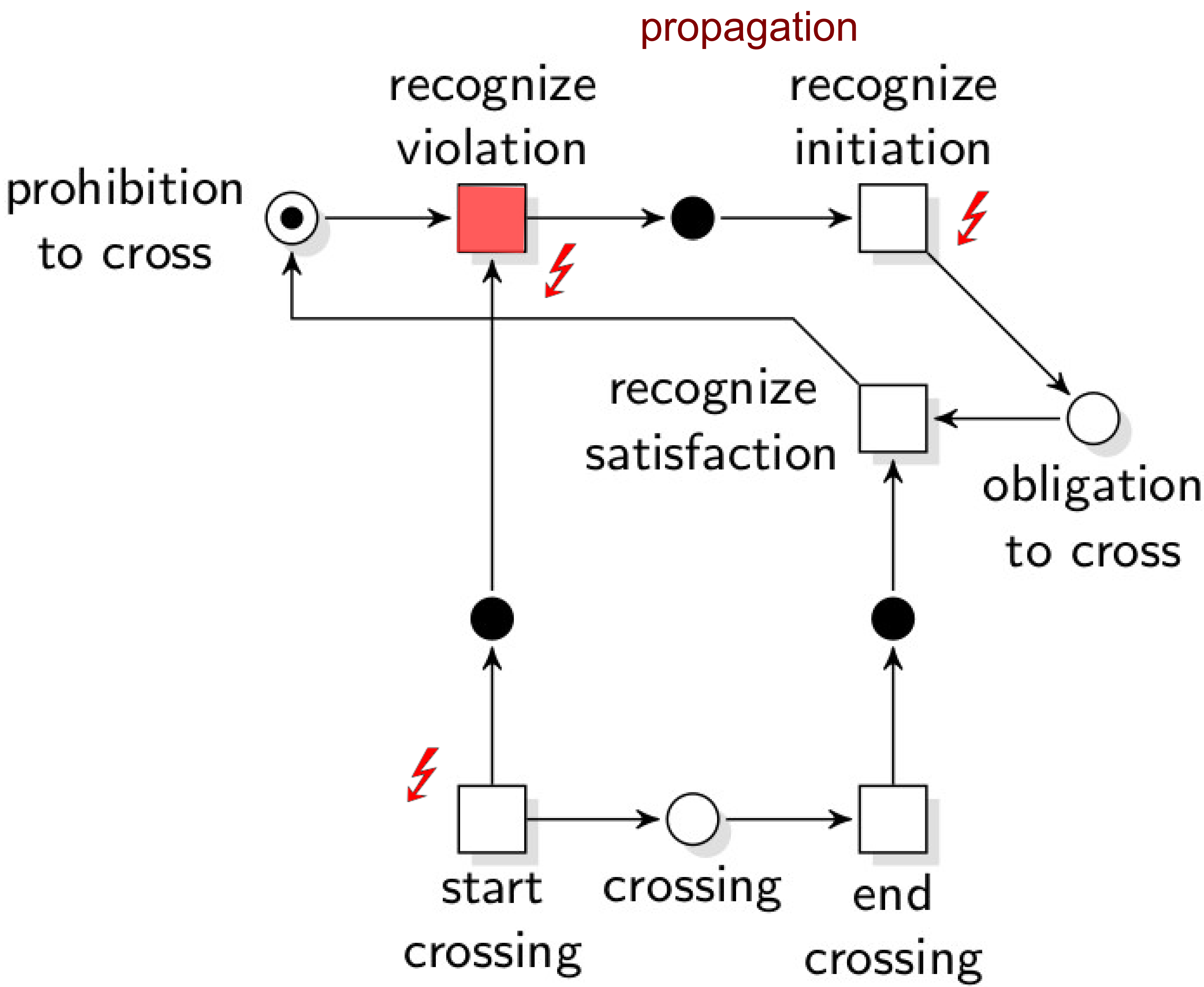


You mustn't cross the road.

If you are crossing the road, cross the road!

Version 1

- You start crossing.

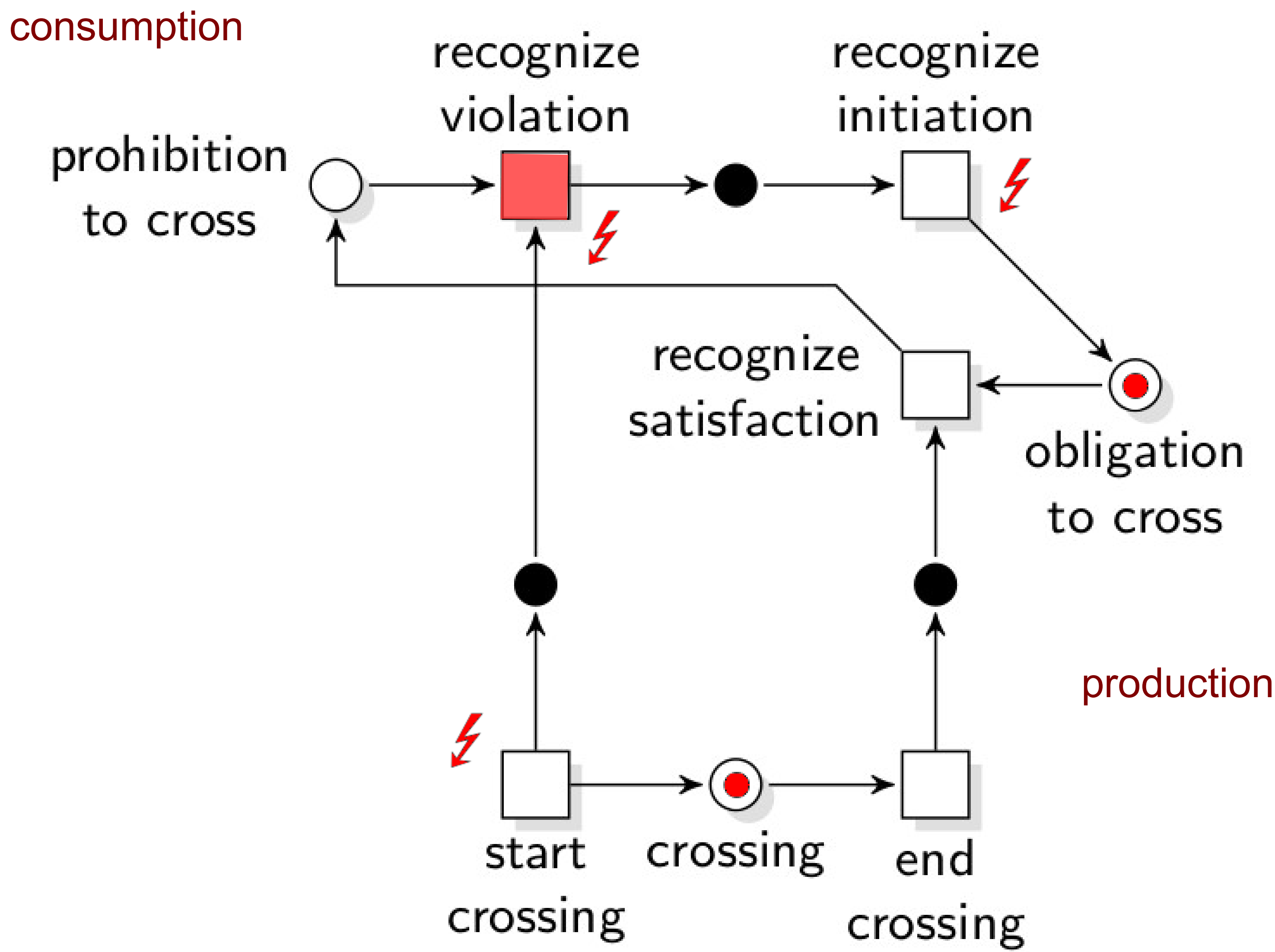


You mustn't cross the road.

If you are crossing the road, cross the road!

Version 1

- You start crossing.

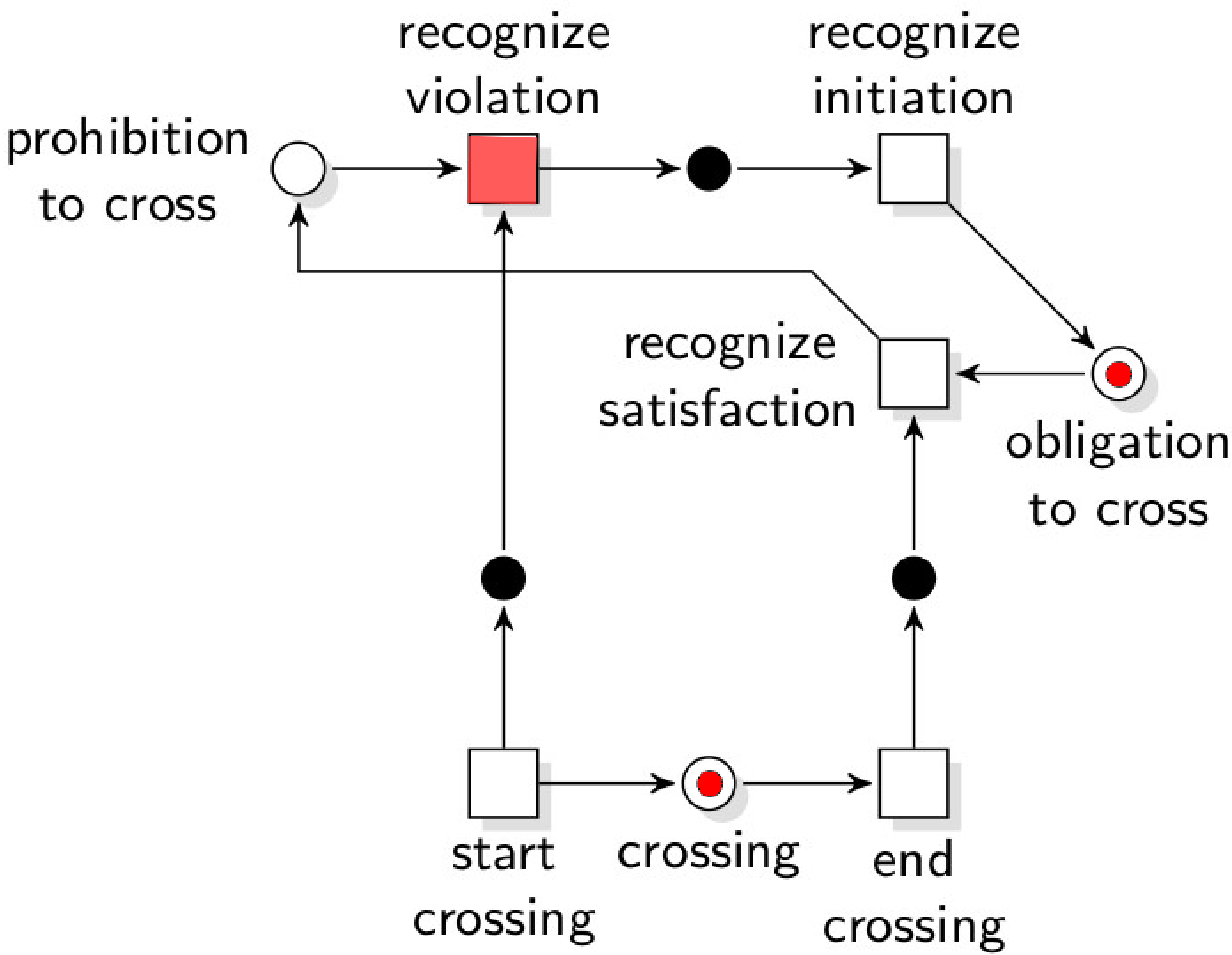


You mustn't cross the road.

Version 1

If you are crossing the road, cross the road!

- You *are* crossing.



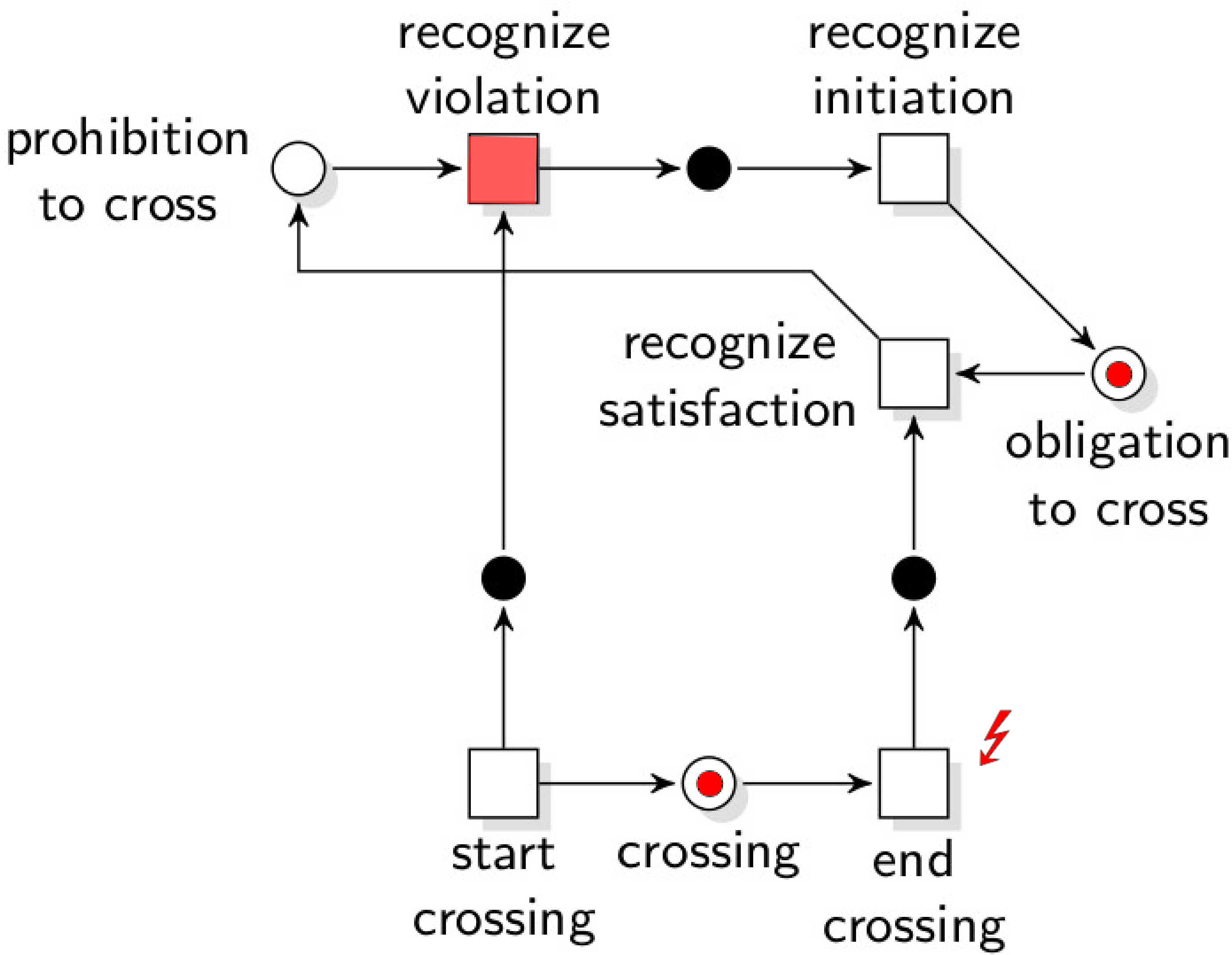
- Not so realistic: intuitively, we still have such prohibition.

You mustn't cross the road.

If you are crossing the road, cross the road!

Version 1

- You *finish* crossing.



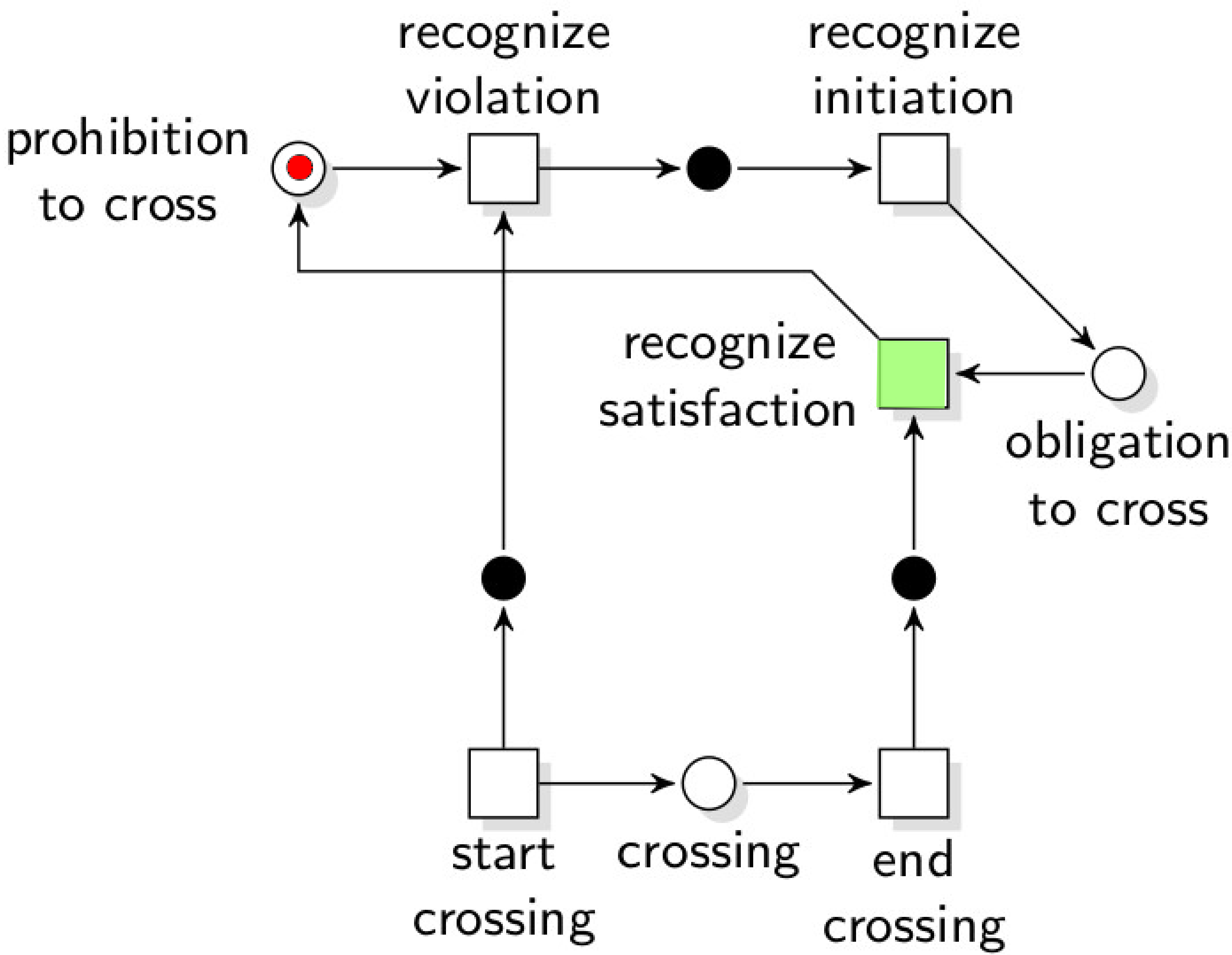
- Not so realistic: intuitively, we still have such prohibition.

You mustn't cross the road.

If you are crossing the road, cross the road!

Version 1

- You *finish* crossing.

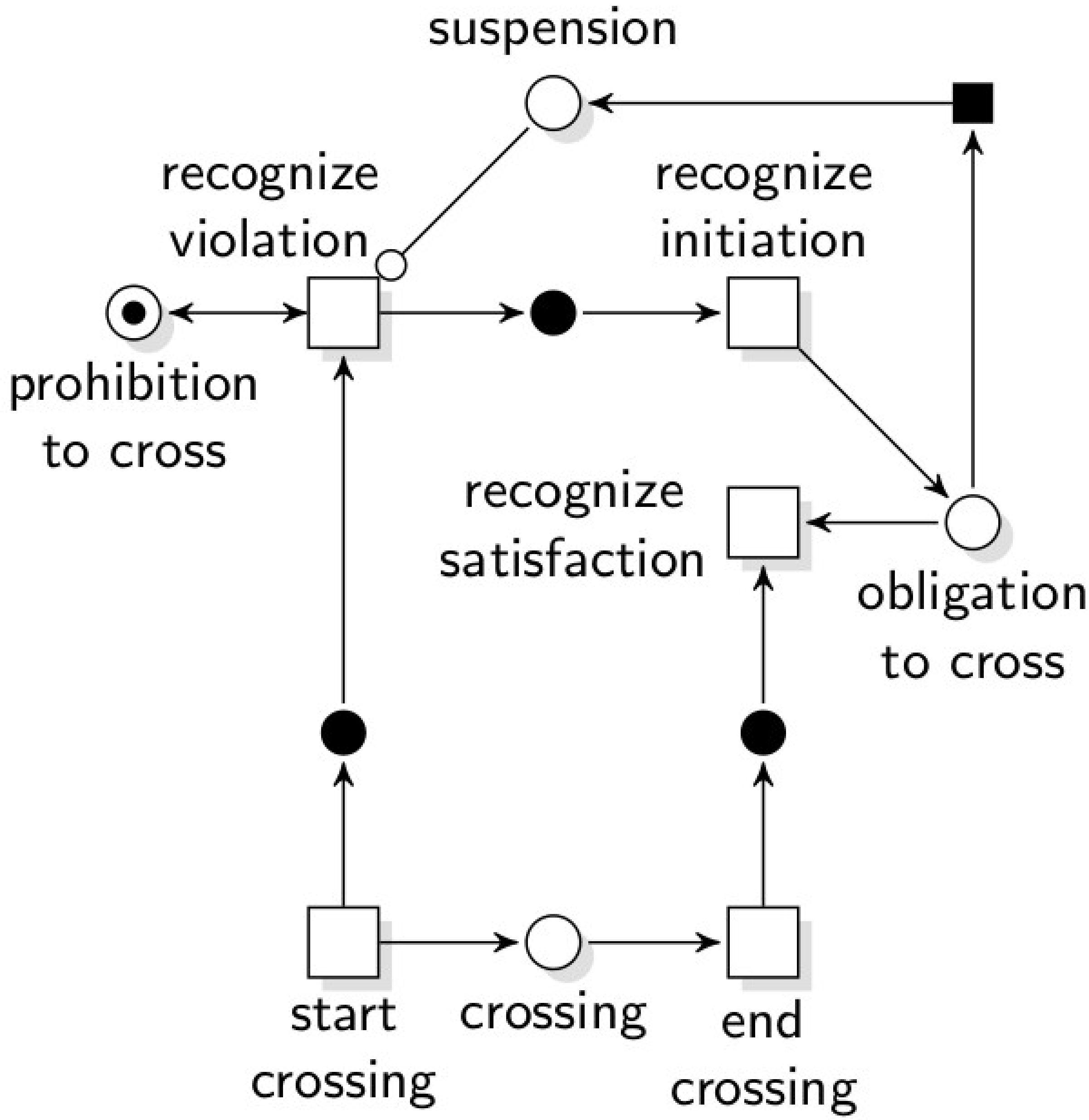


- Not so realistic: intuitively, we still have such prohibition.

You mustn't cross the road.

Version 2

If you are crossing the road,
cross the road!

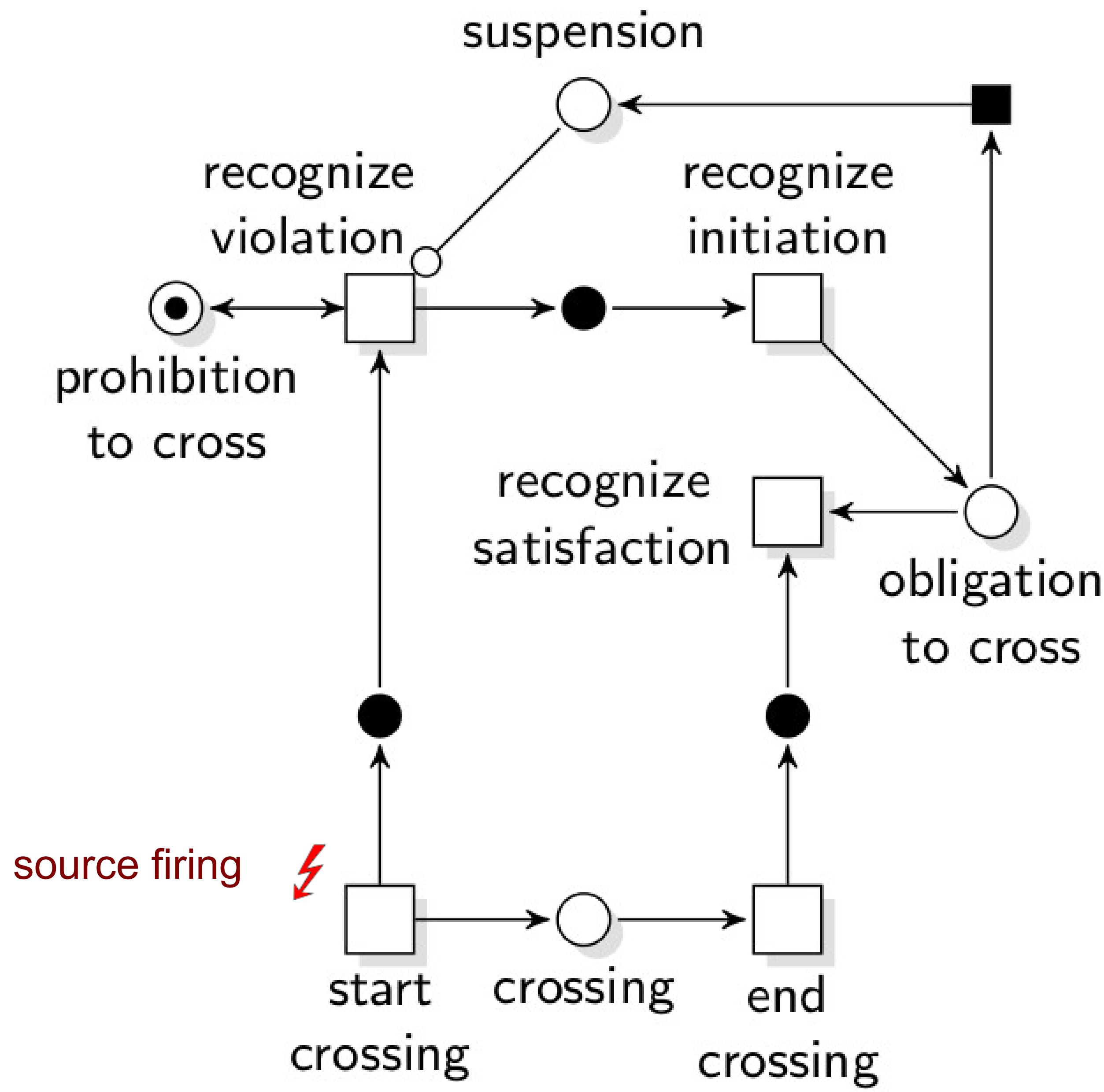


You mustn't cross the road.

If you are crossing the road, cross the road!

- You start crossing.

Version 2

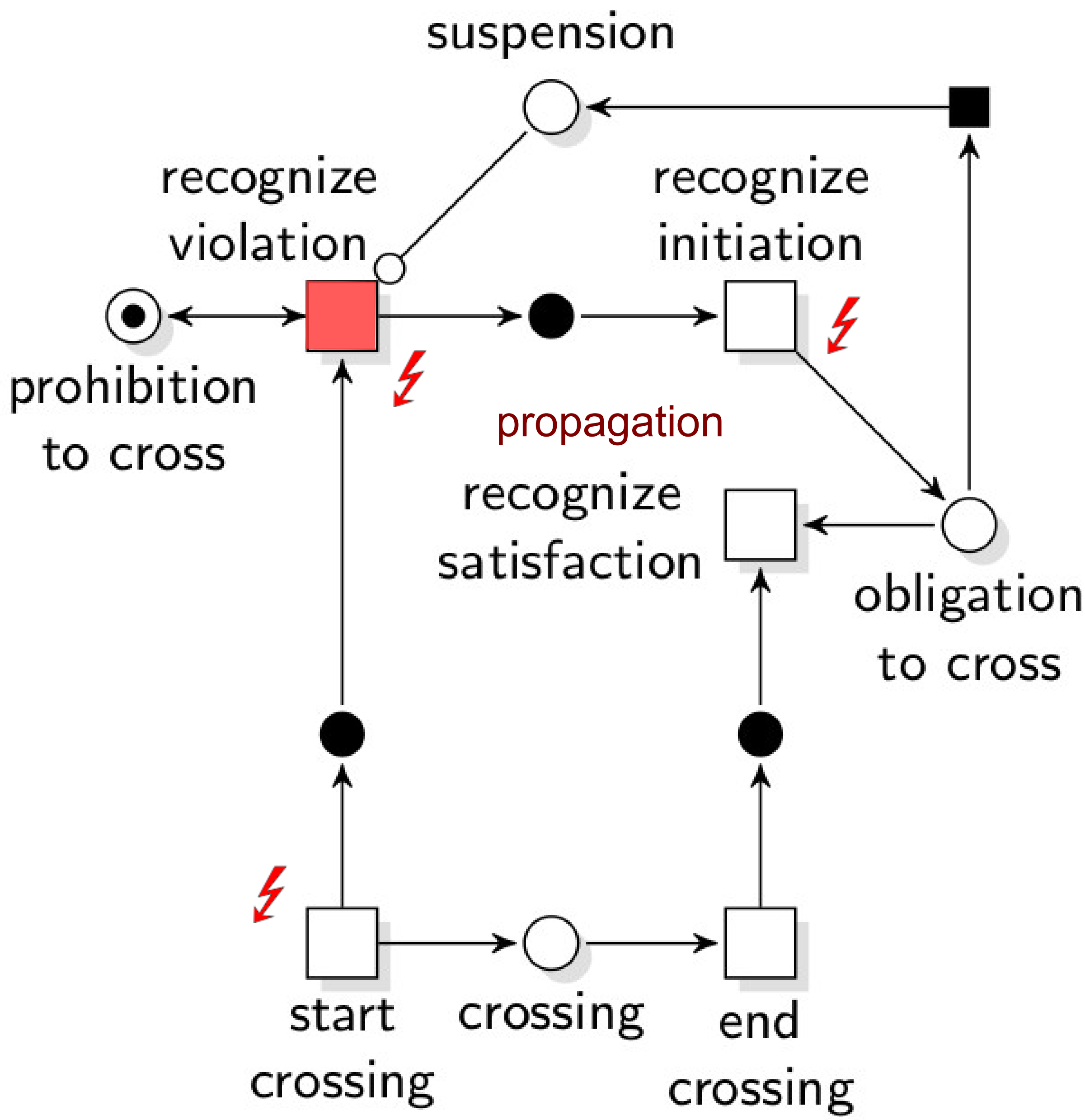


You mustn't cross the road.

If you are crossing the road, cross the road!

- You start crossing.

Version 2

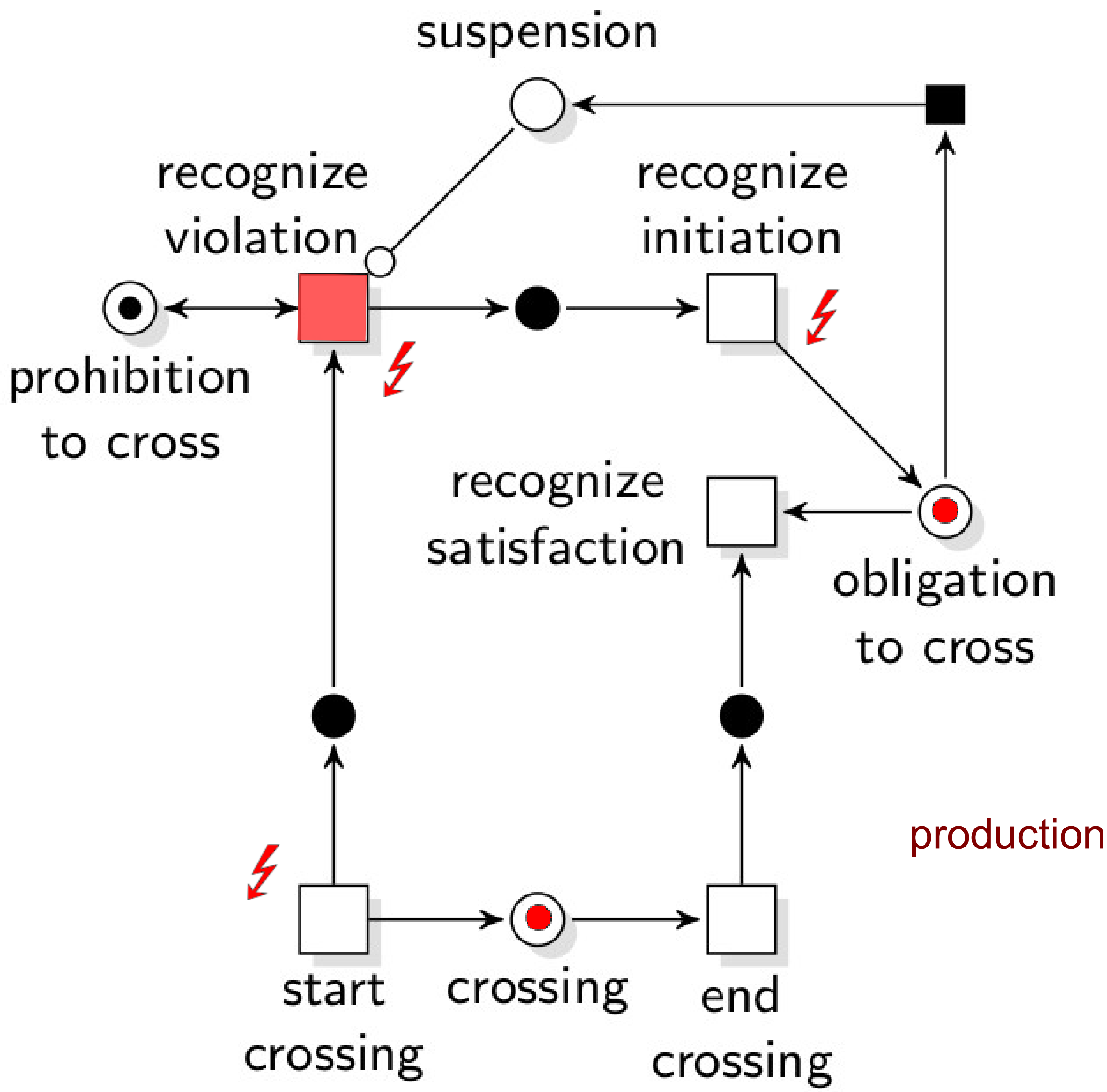


You mustn't cross the road.

If you are crossing the road, cross the road!

- You start crossing.

Version 2

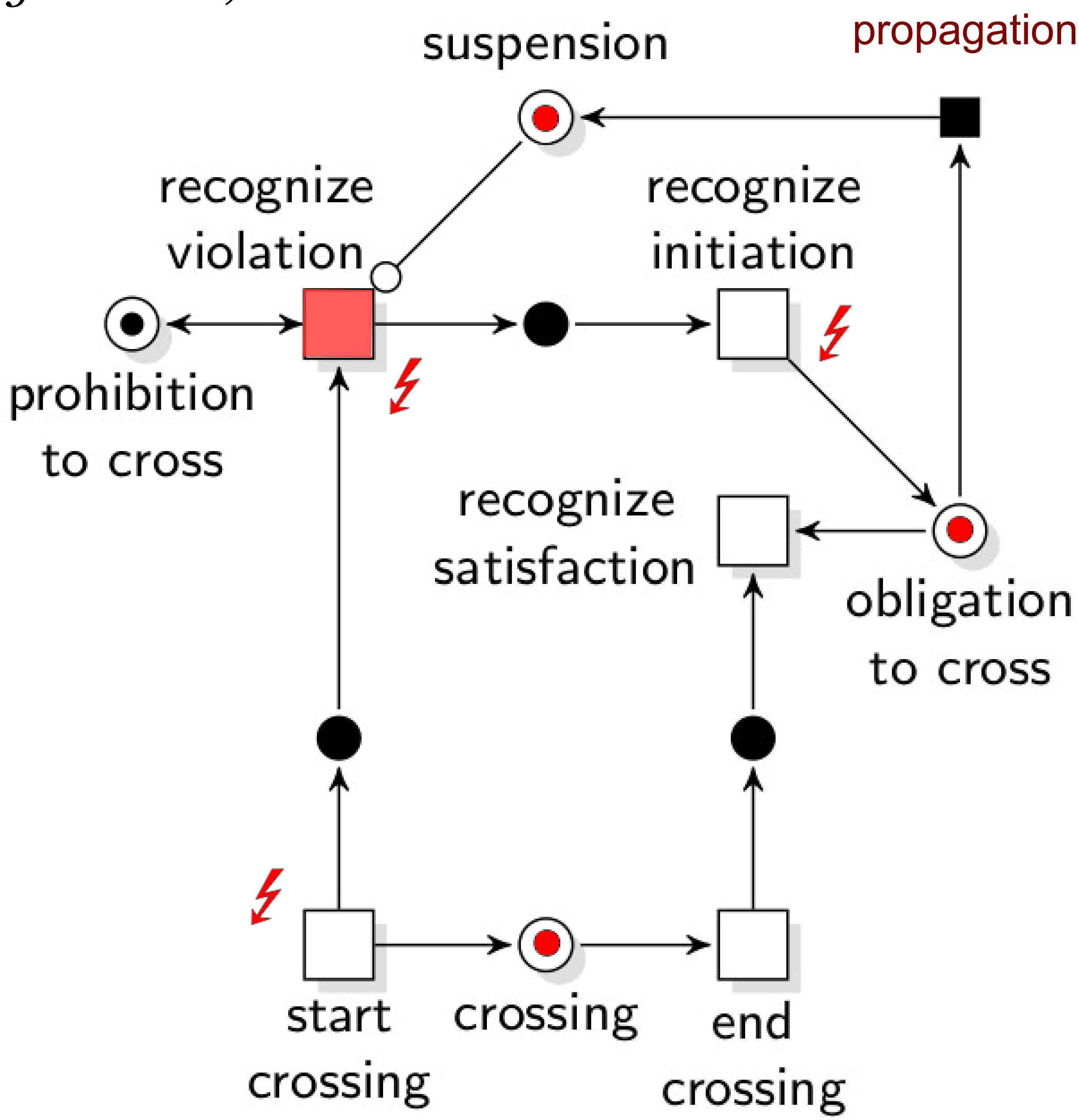


You mustn't cross the road.

If you are crossing the road, cross the road!

- You start crossing.

Version 2

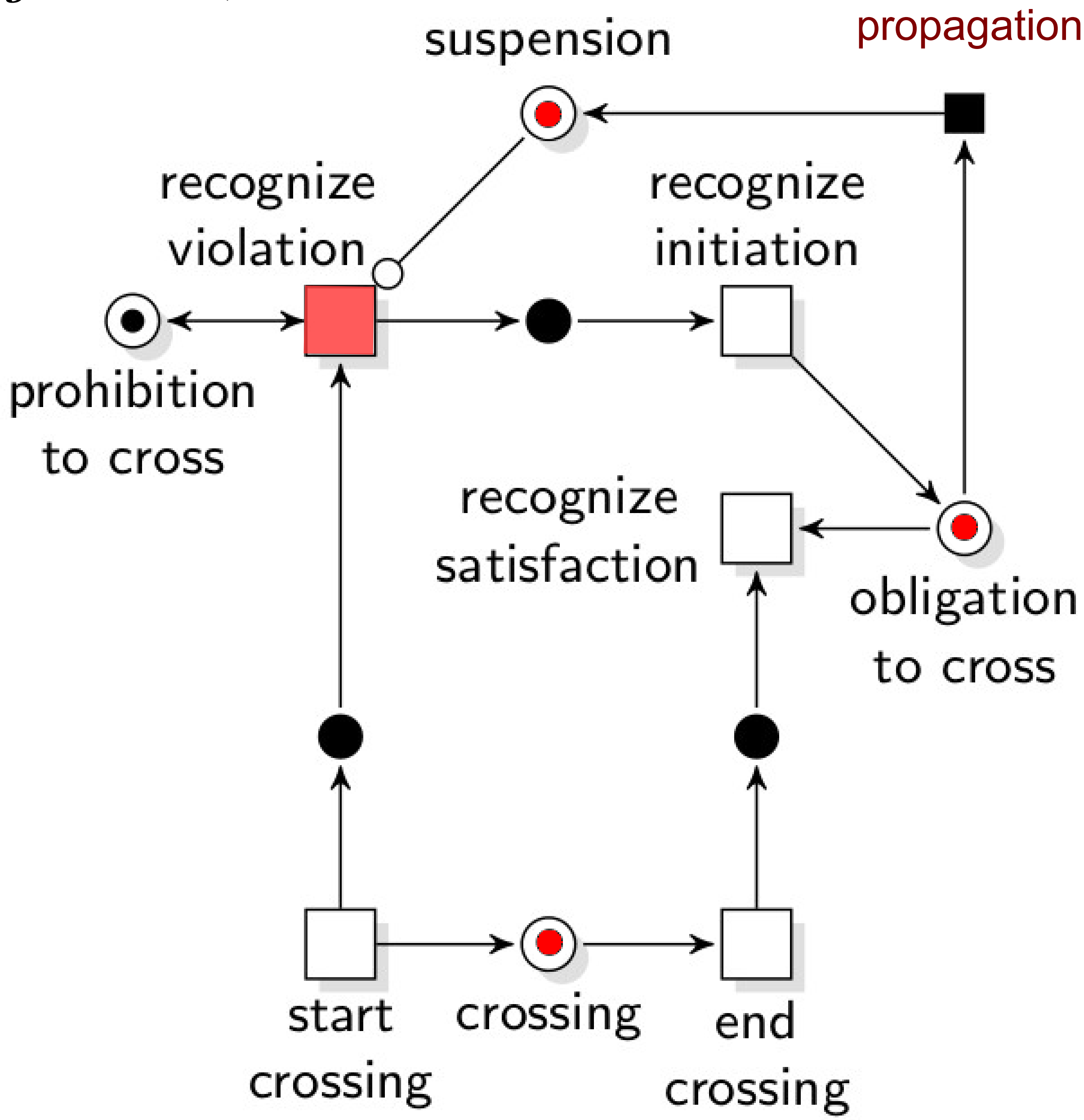


You mustn't cross the road.

If you are crossing the road, cross the road!

• You *are* crossing.

Version 2



Obligation has higher priority (topology captures *salience*).

Gentle murderer

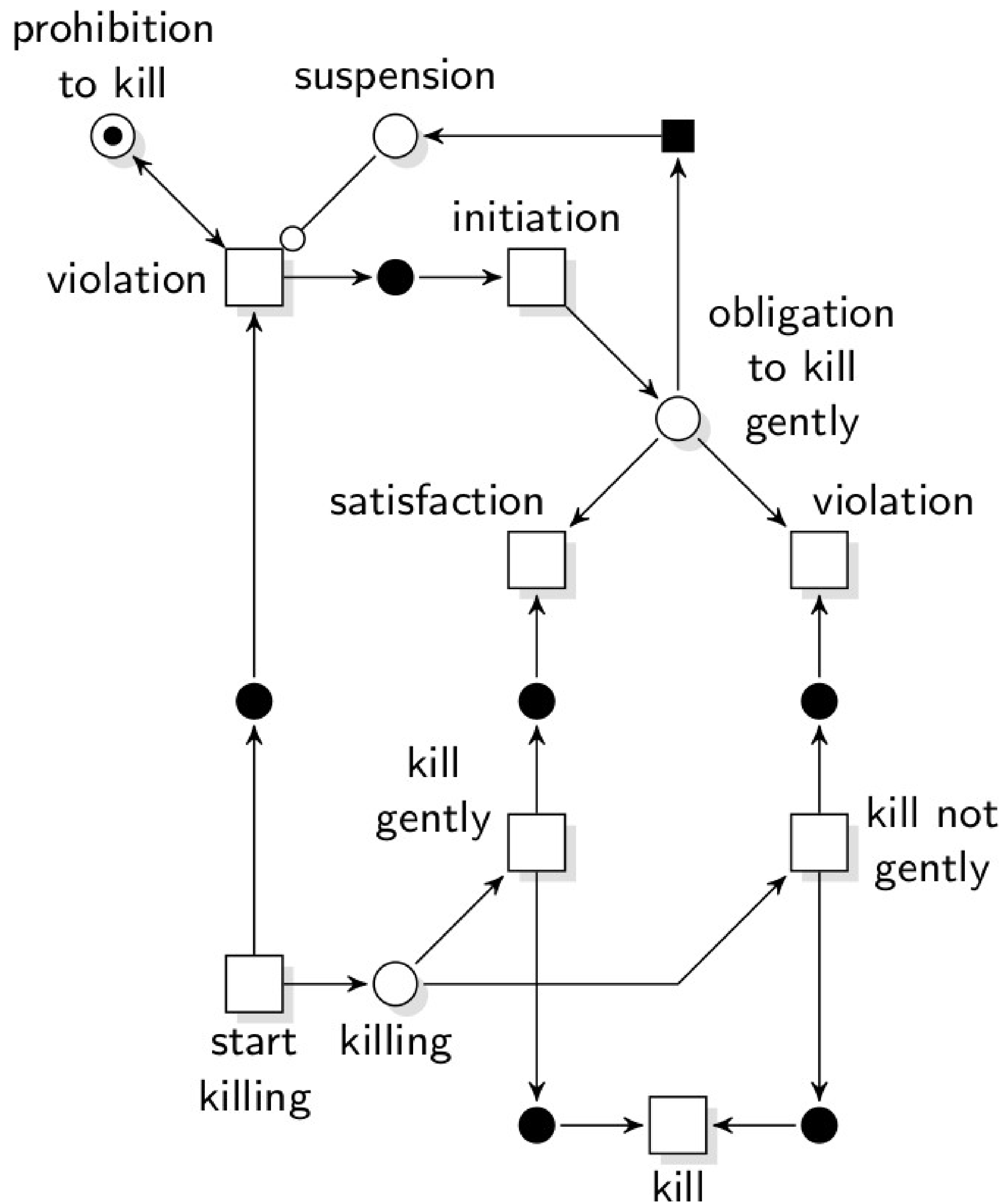
[Forrester, 1984]

It is forbidden to kill,
but if one kills,
one ought to kill gently.

Gentle murderer

[Forrester, 1984]

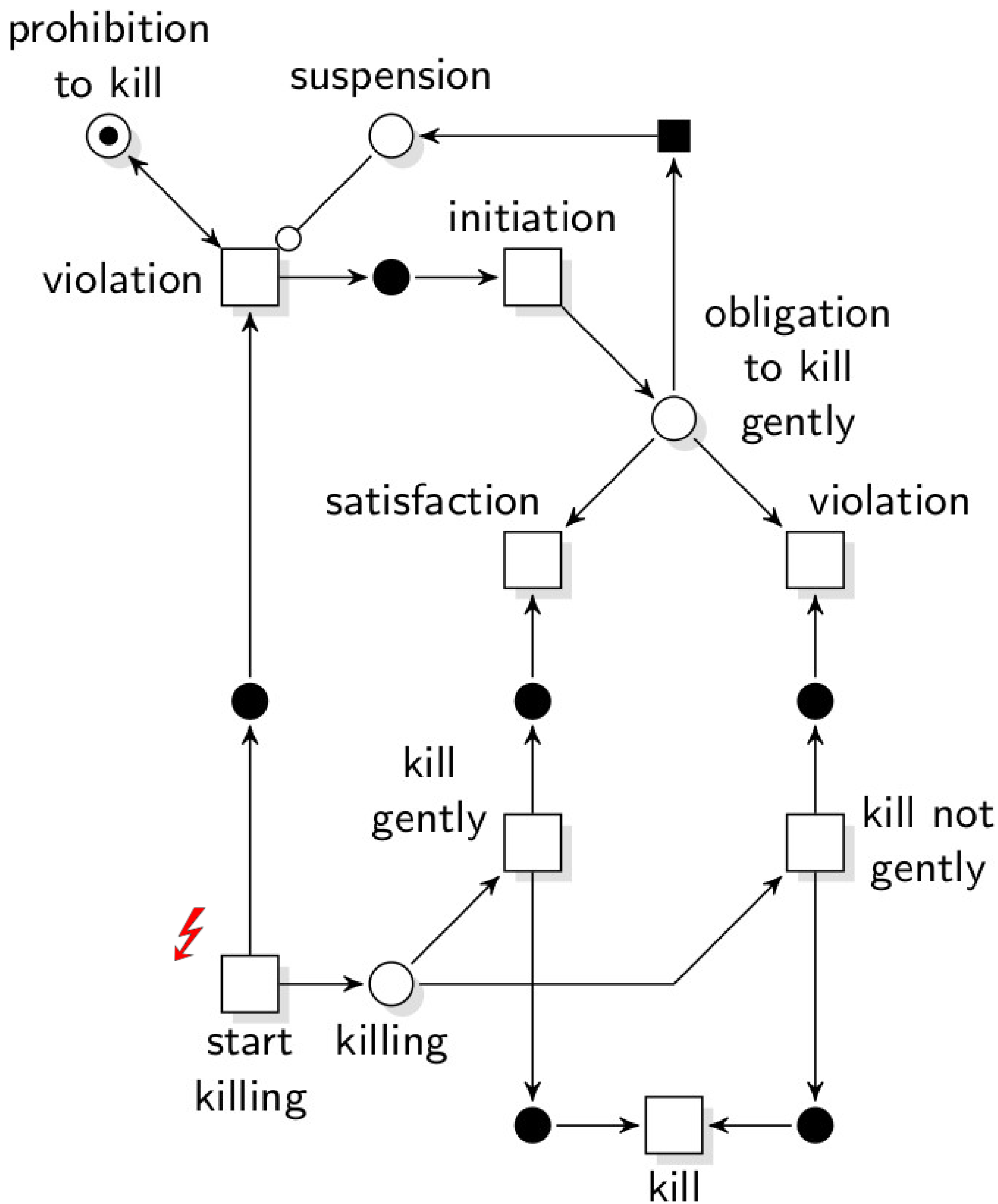
It is forbidden to kill,
but if one kills,
one ought to kill gently.



Gentle murderer

[Forrester, 1984]

It is forbidden to kill,
but if one kills,
one ought to kill gently.

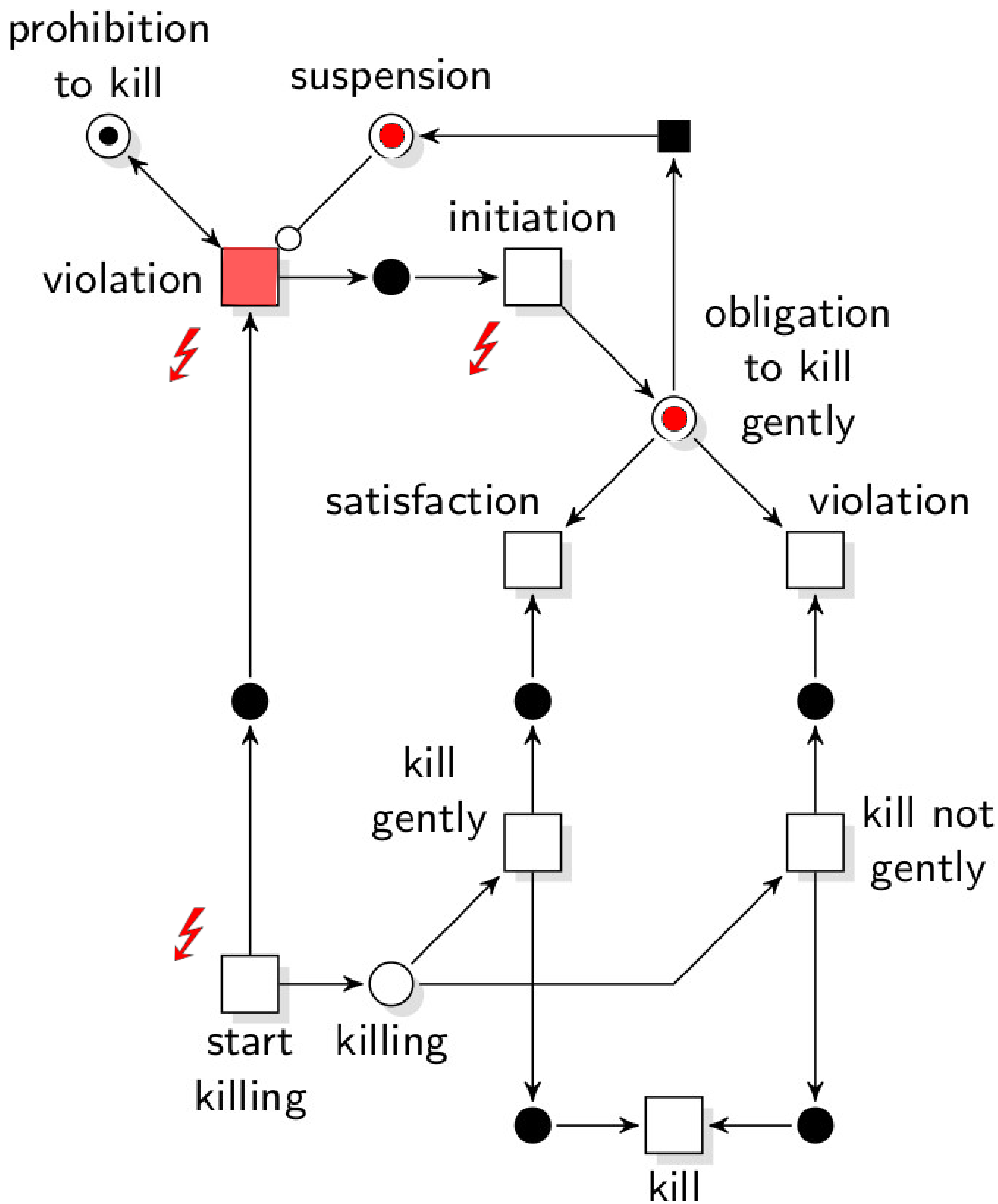


- He starts killing.

Gentle murderer

[Forrester, 1984]

It is forbidden to kill,
but if one kills,
one ought to kill gently.



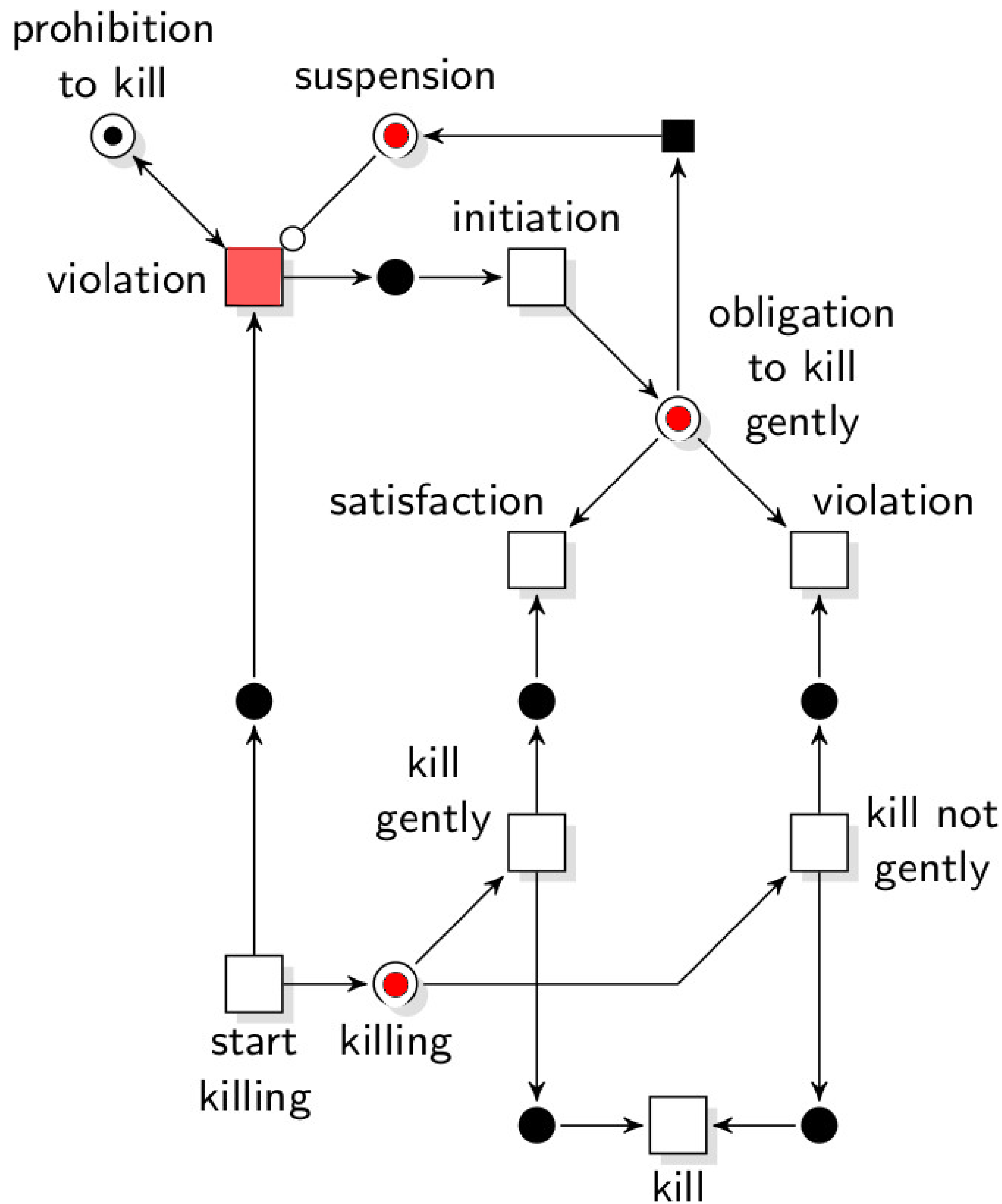
- He starts killing.

Gentle murderer

[Forrester, 1984]

It is forbidden to kill,
but if one kills,
one ought to kill gently.

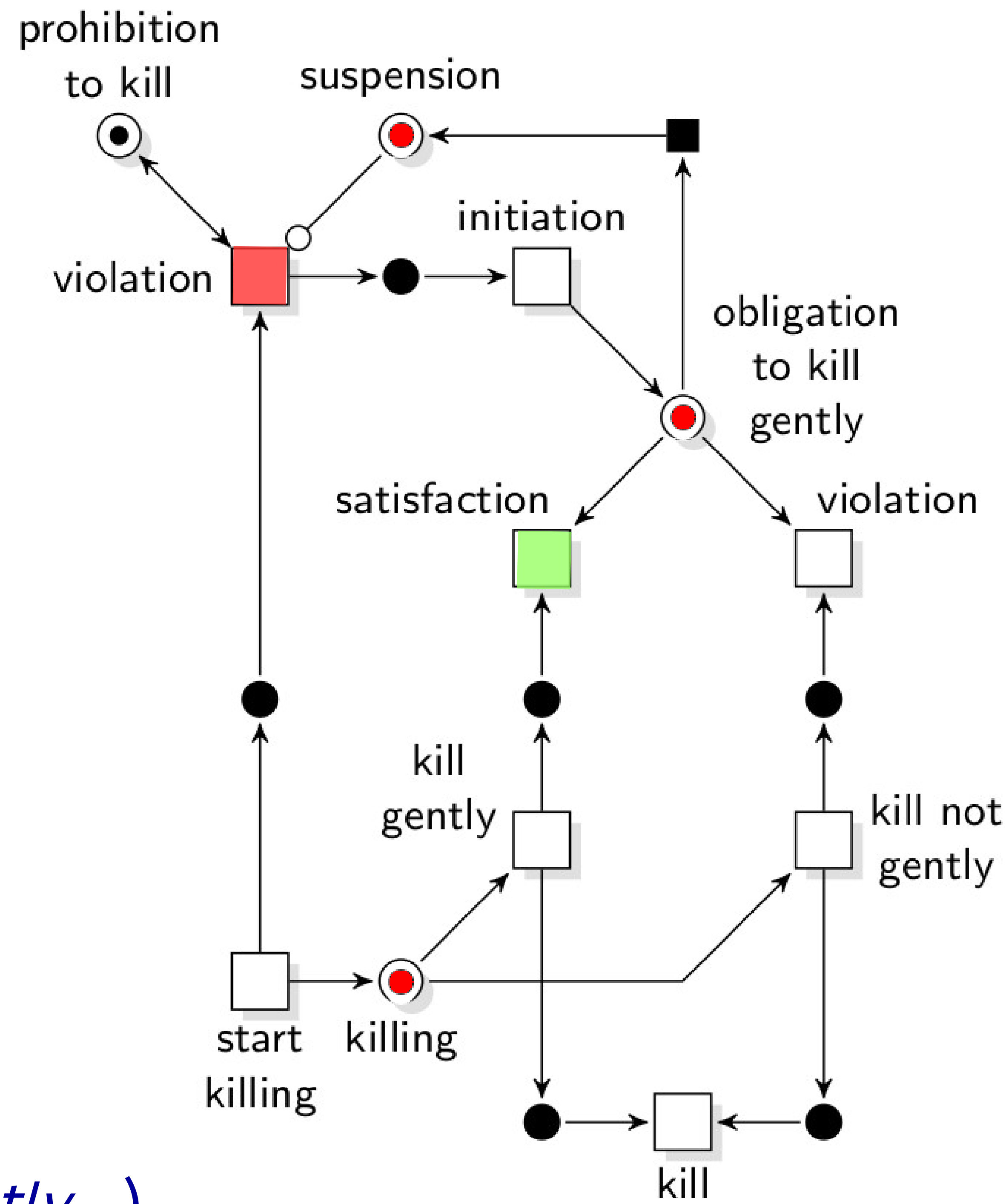
- He starts killing.



Gentle murderer

[Forrester, 1984]

It is forbidden to kill,
but if one kills,
one ought to kill gently.

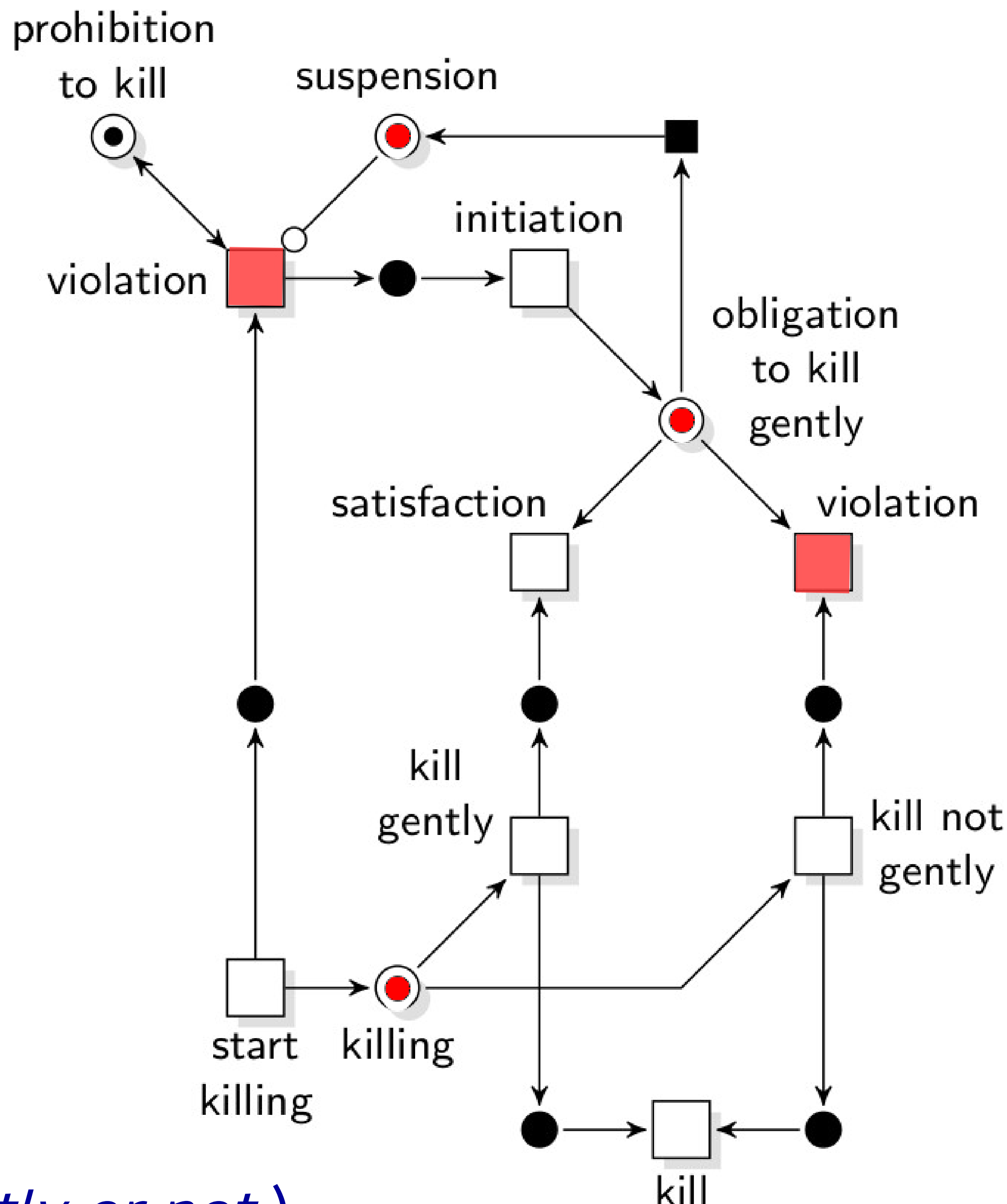


- He starts killing.
  (*either he kills gently...*)

Gentle murderer

[Forrester, 1984]

It is forbidden to kill,
but if one kills,
one ought to kill gently.



- He starts killing.
  (*either he kills gently or not.*)

White fence

[Prakken & Sergot, 1996]

There must be no fence.
If there is a fence, it must be a white fence.
If the cottage is by the sea, there may be a fence.

White fence

[Prakken & Sergot, 1996]

There must be no fence.

If there is a fence, it must be a white fence.

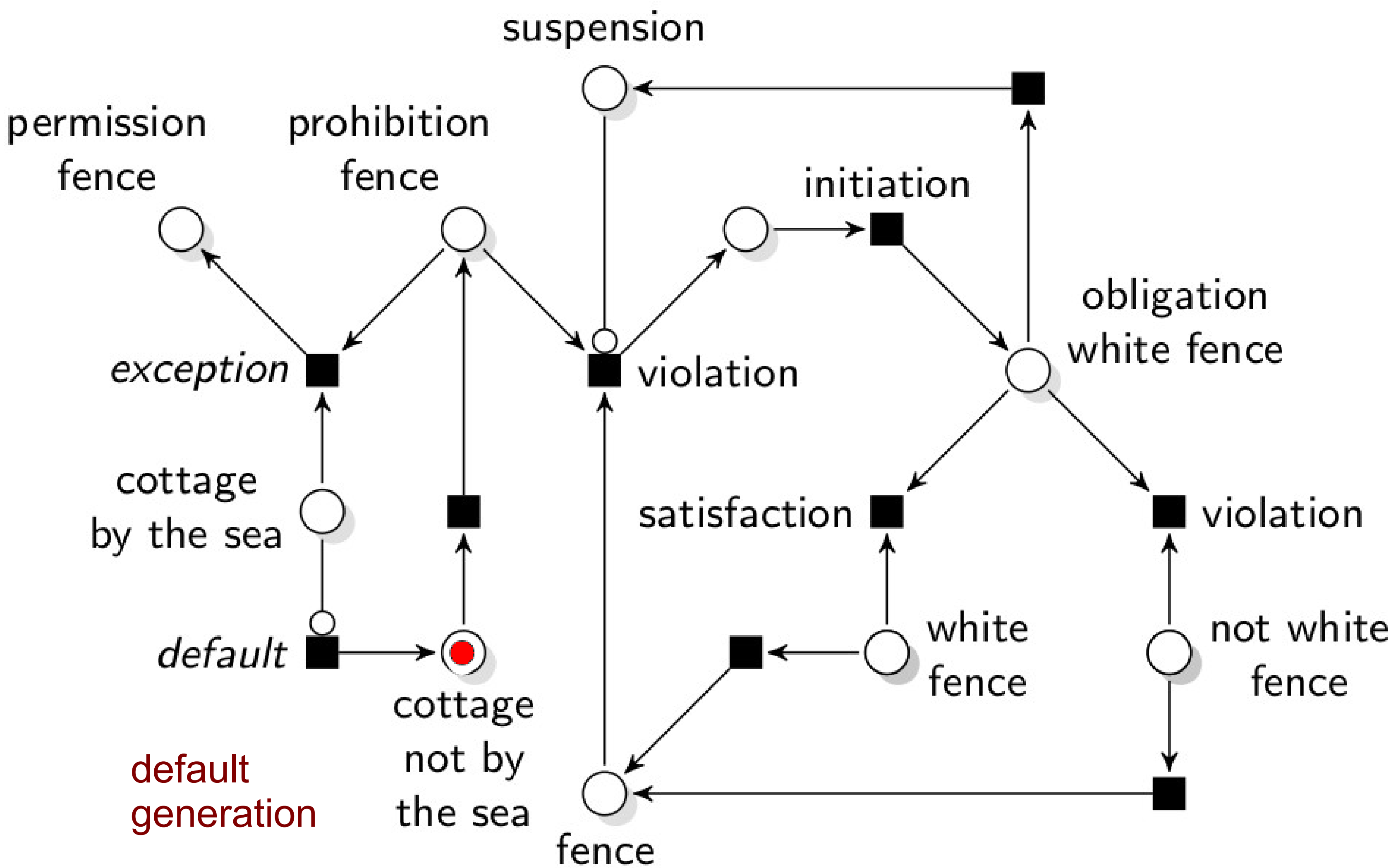
If the cottage is by the sea, there may be a fence.

exception, implicit default

White fence

[Prakken & Sergot, 1996]

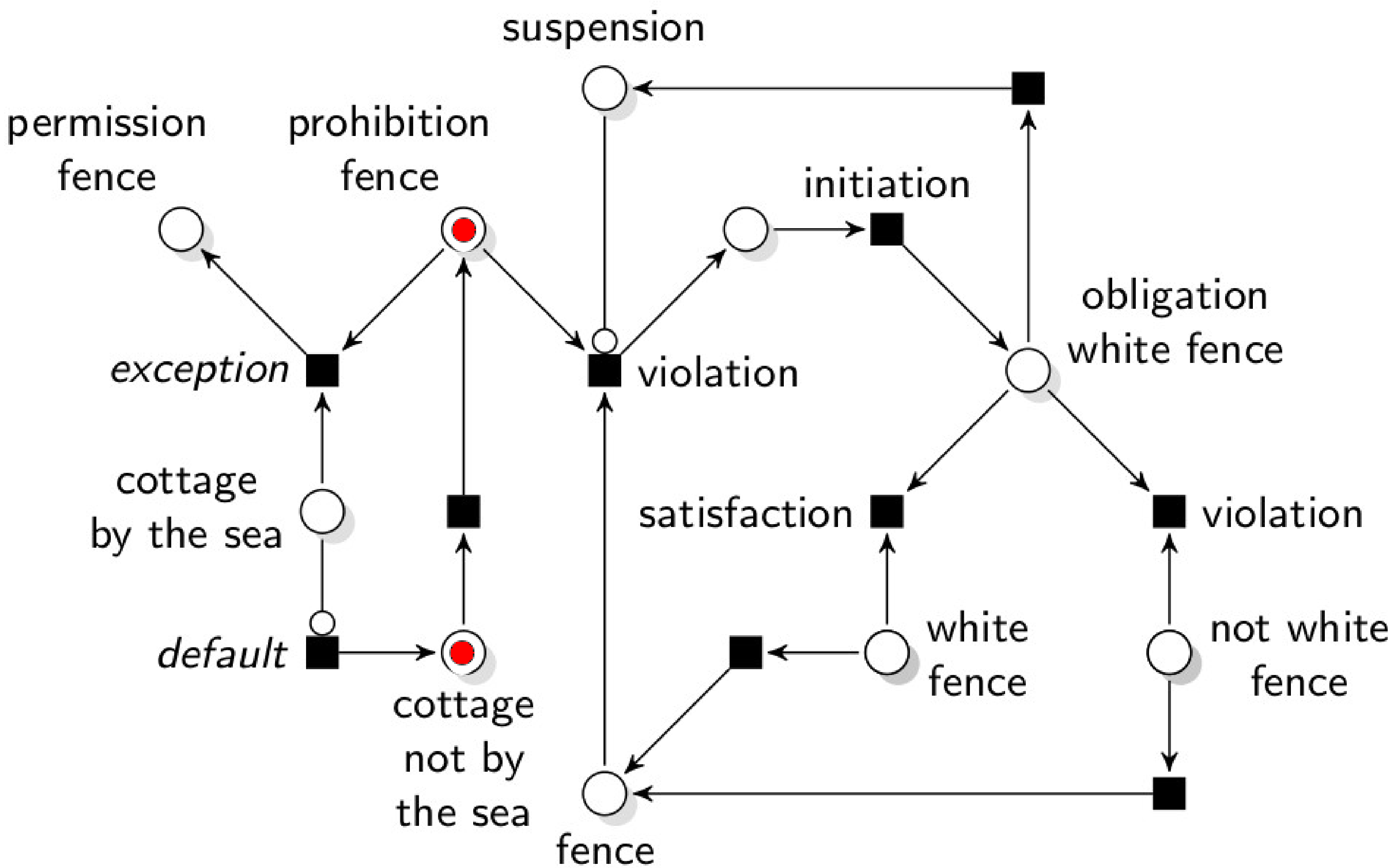
There must be no fence.
If there is a fence, it must be a white fence.
If the cottage is by the sea, there may be a fence.



White fence

[Prakken & Sergot, 1996]

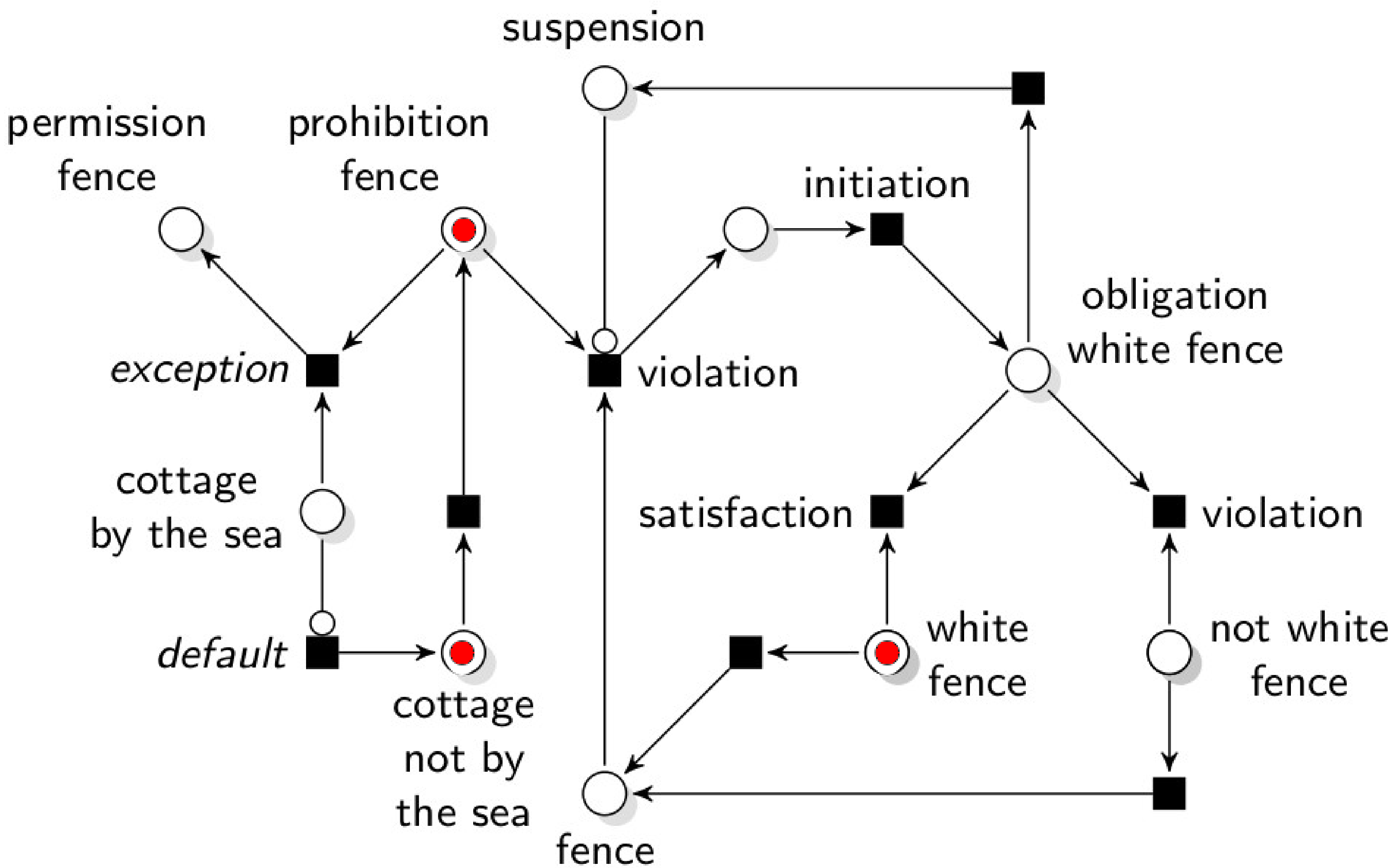
There must be no fence.
If there is a fence, it must be a white fence.
If the cottage is by the sea, there may be a fence.



White fence

[Prakken & Sergot, 1996]

There must be no fence.
If there is a fence, it must be a white fence.
If the cottage is by the sea, there may be a fence.

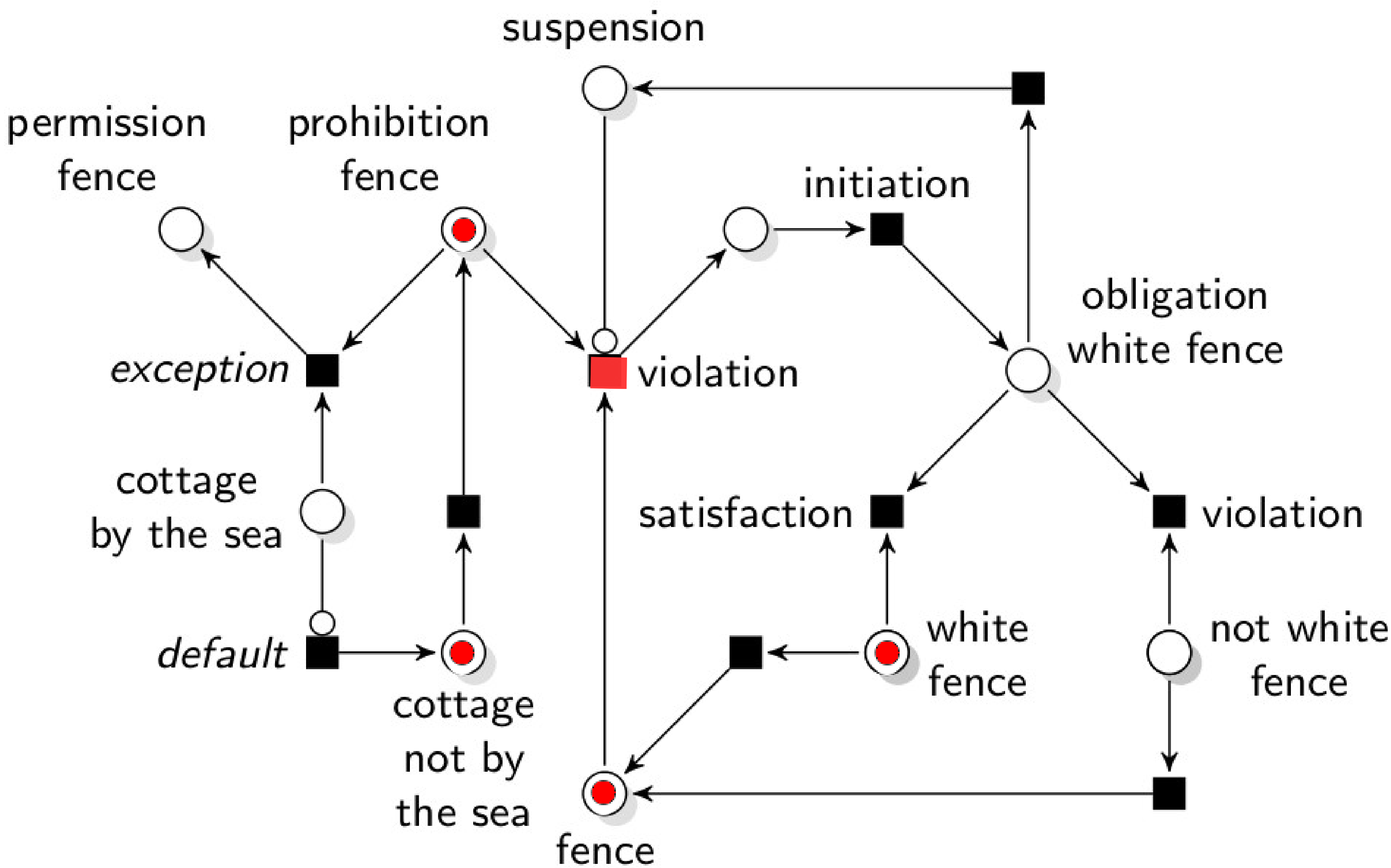


- There is a white fence.

White fence

[Prakken & Sergot, 1996]

There must be no fence.
If there is a fence, it must be a white fence.
If the cottage is by the sea, there may be a fence.

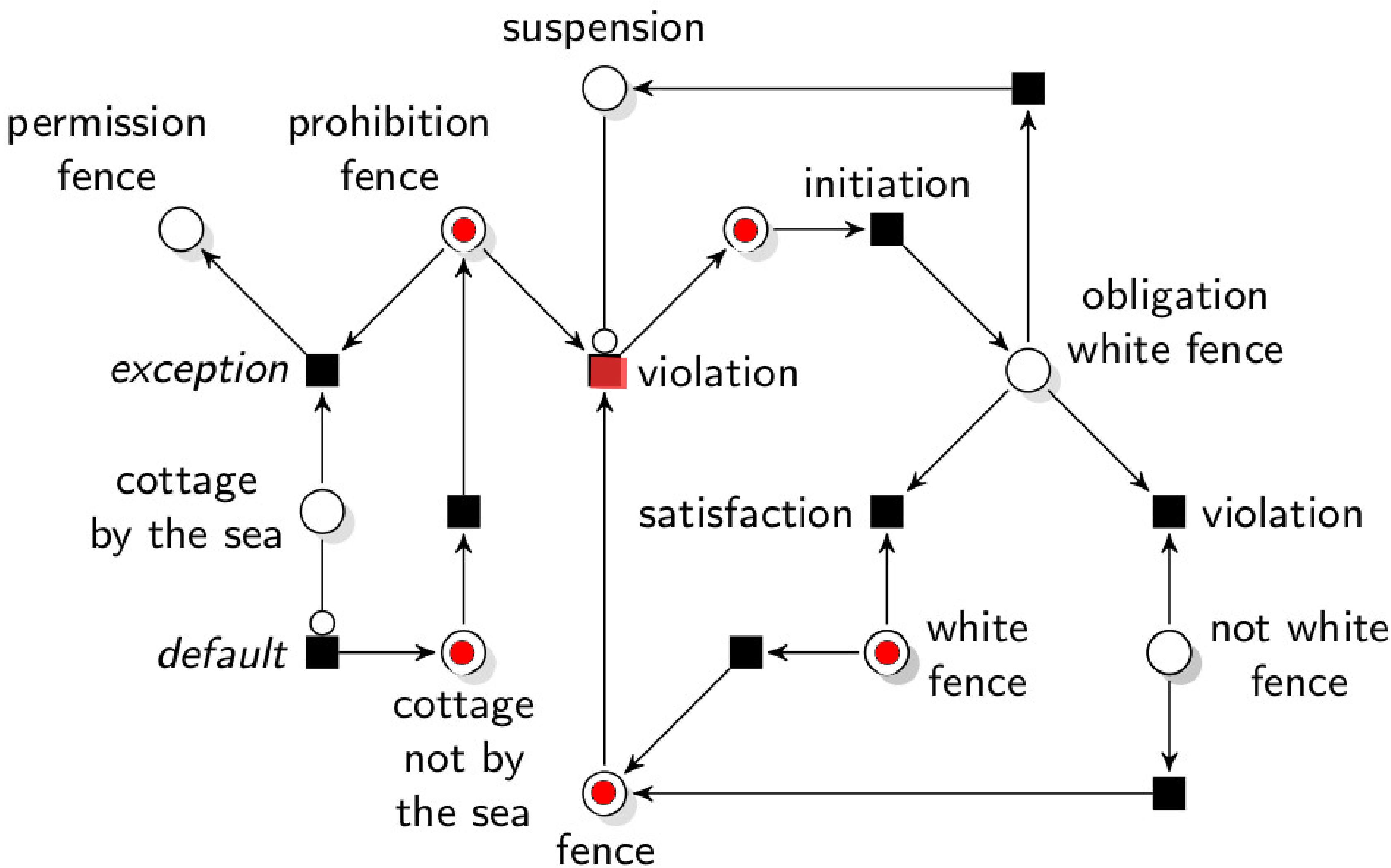


- There is a white fence.

White fence

[Prakken & Sergot, 1996]

There must be no fence.
If there is a fence, it must be a white fence.
If the cottage is by the sea, there may be a fence.

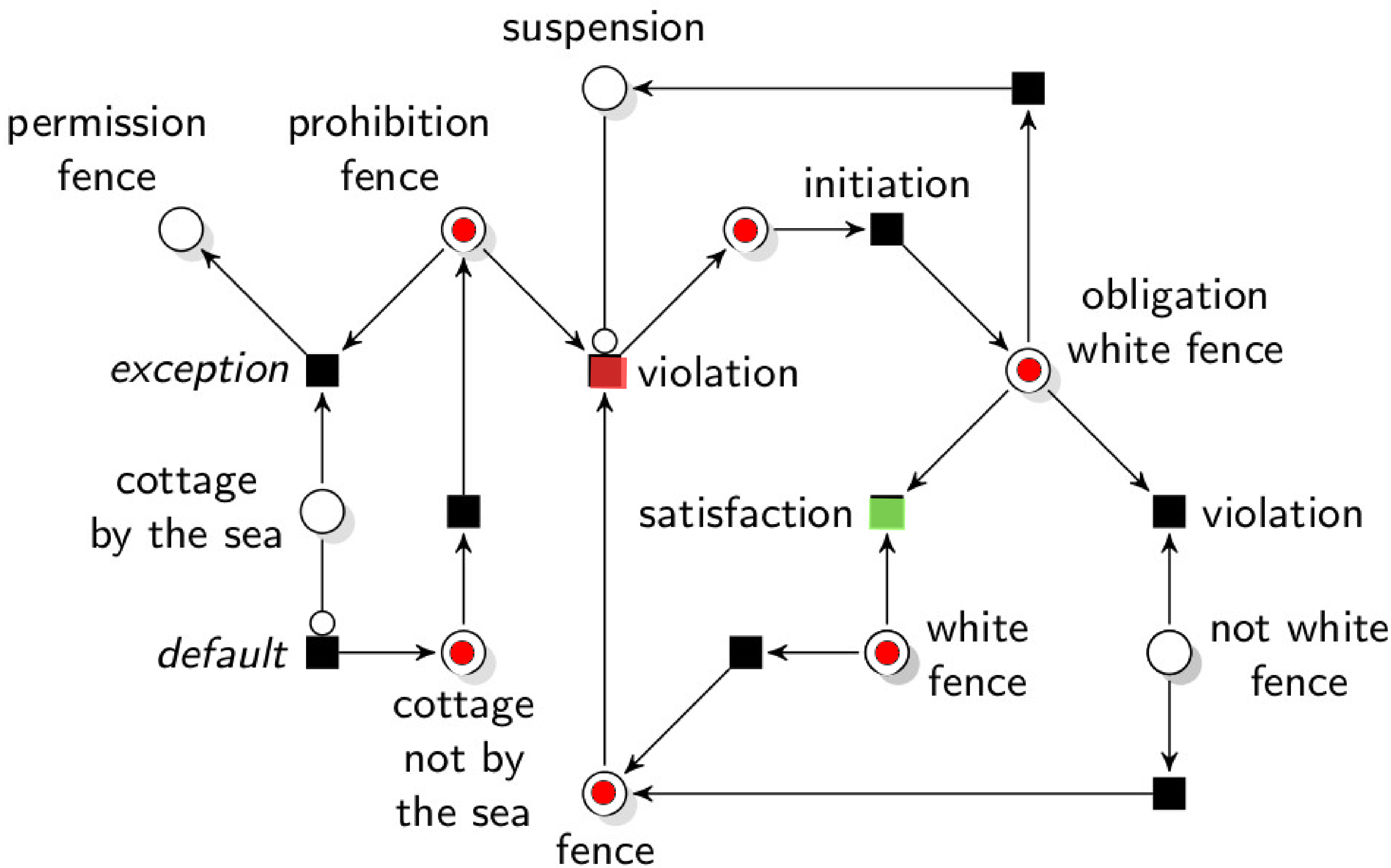


- There is a white fence.

White fence

[Prakken & Sergot, 1996]

There must be no fence.
If there is a fence, it must be a white fence.
If the cottage is by the sea, there may be a fence.



- There is a white fence.

Privacy Act

[Governatori, 2015]

- *The collection of personal information is forbidden, unless acting on a court order authorising it.*
- *The destruction of illegally collected personal information before accessing it is a defence against the illegal collection of the personal information.*
- *The collection of medical information is forbidden, unless the entity collecting the medical information is permitted to collect personal information.*

Privacy Act

[Governatori, 2015]

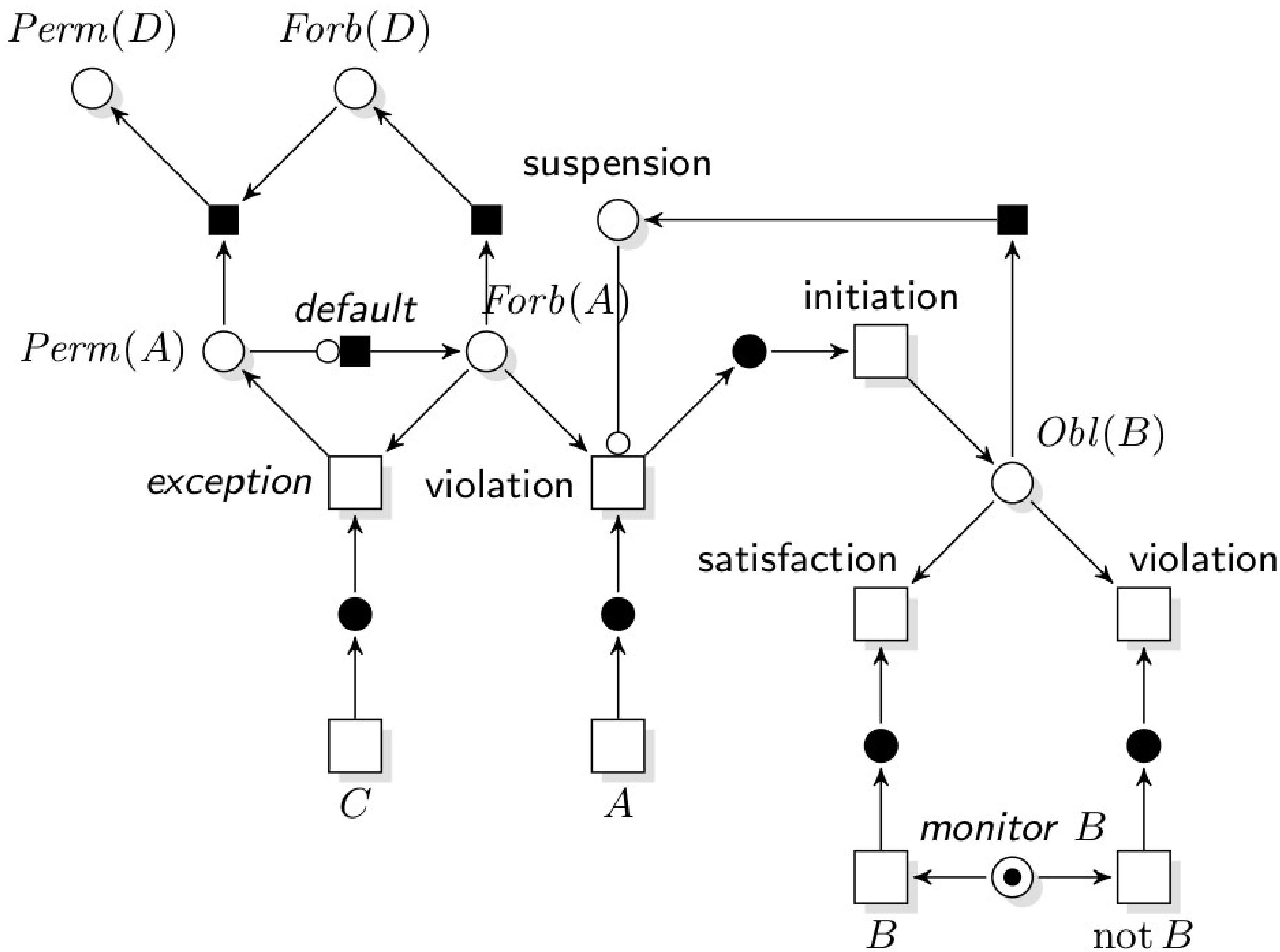
Forbidden A. If C, then Permitted A.
If Forbidden A and A, then Obligatory B.
Forbidden D. If Permitted A, then Permitted D.

- *The collection of personal information is forbidden, unless acting on a court order authorising it.*
- *The destruction of illegally collected personal information before accessing it is a defence against the illegal collection of the personal information.*
- *The collection of medical information is forbidden, unless the entity collecting the medical information is permitted to collect personal information.*

Privacy Act

[Governatori, 2015]

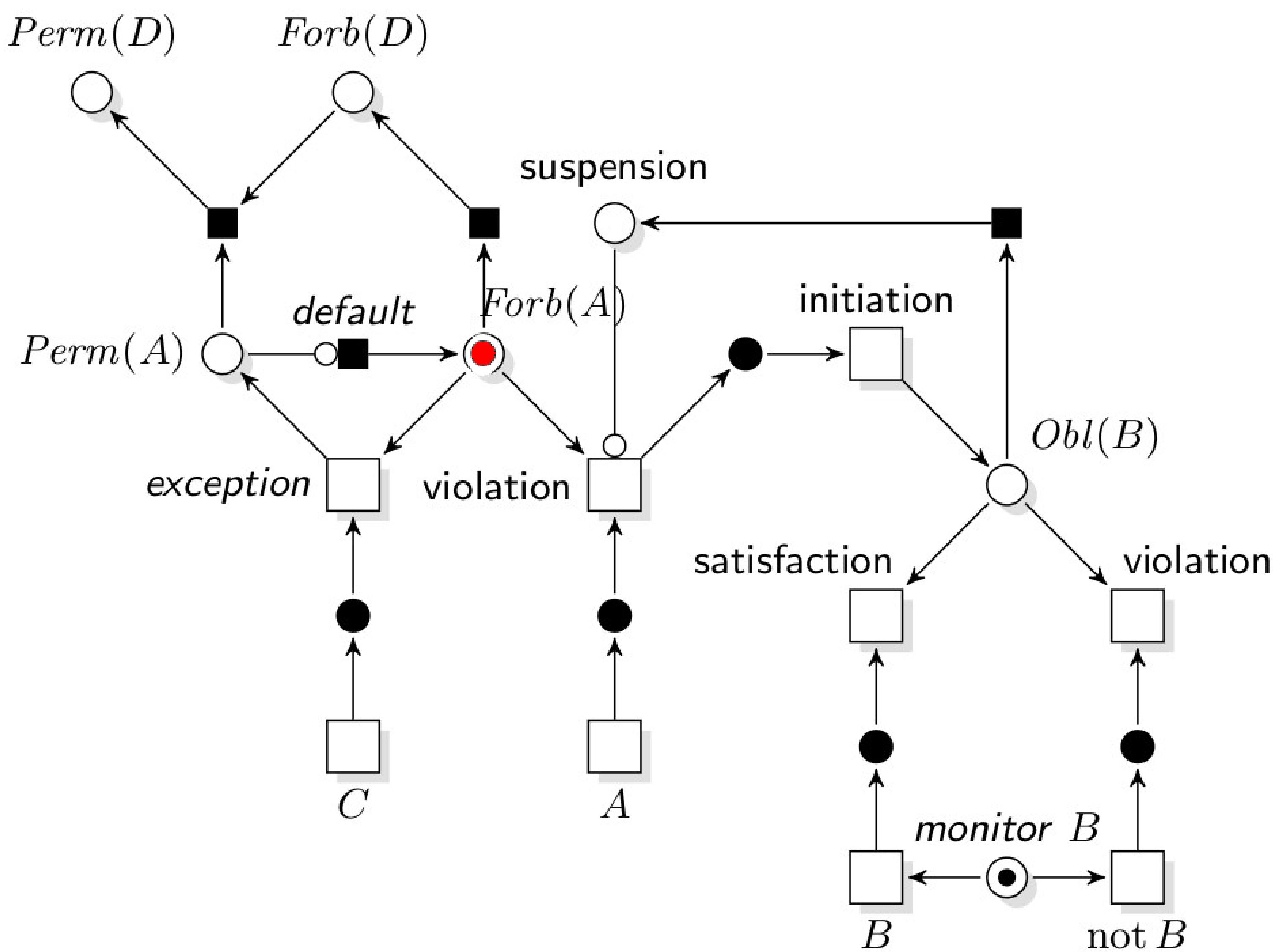
Forbidden A. If C, then Permitted A.
If Forbidden A and A, then Obligatory B.
Forbidden D. If Permitted A, then Permitted D.



Privacy Act

[Governatori, 2015]

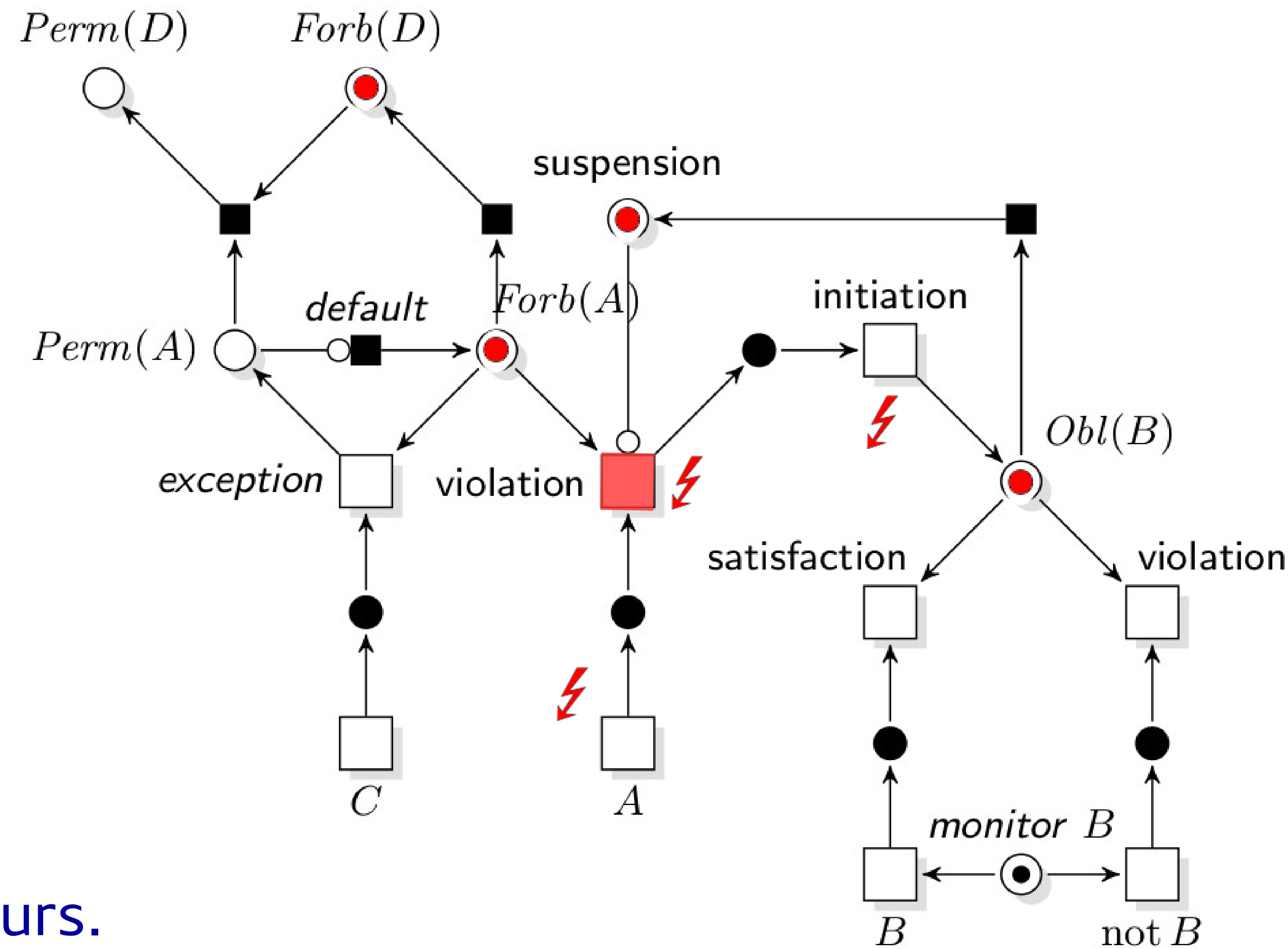
Forbidden A. If C, then Permitted A.
If Forbidden A and A, then Obligatory B.
Forbidden D. If Permitted A, then Permitted D.



Privacy Act

[Governatori, 2015]

Forbidden A. If C, then Permitted A.
If Forbidden A and A, then Obligatory B.
Forbidden D. If Permitted A, then Permitted D.

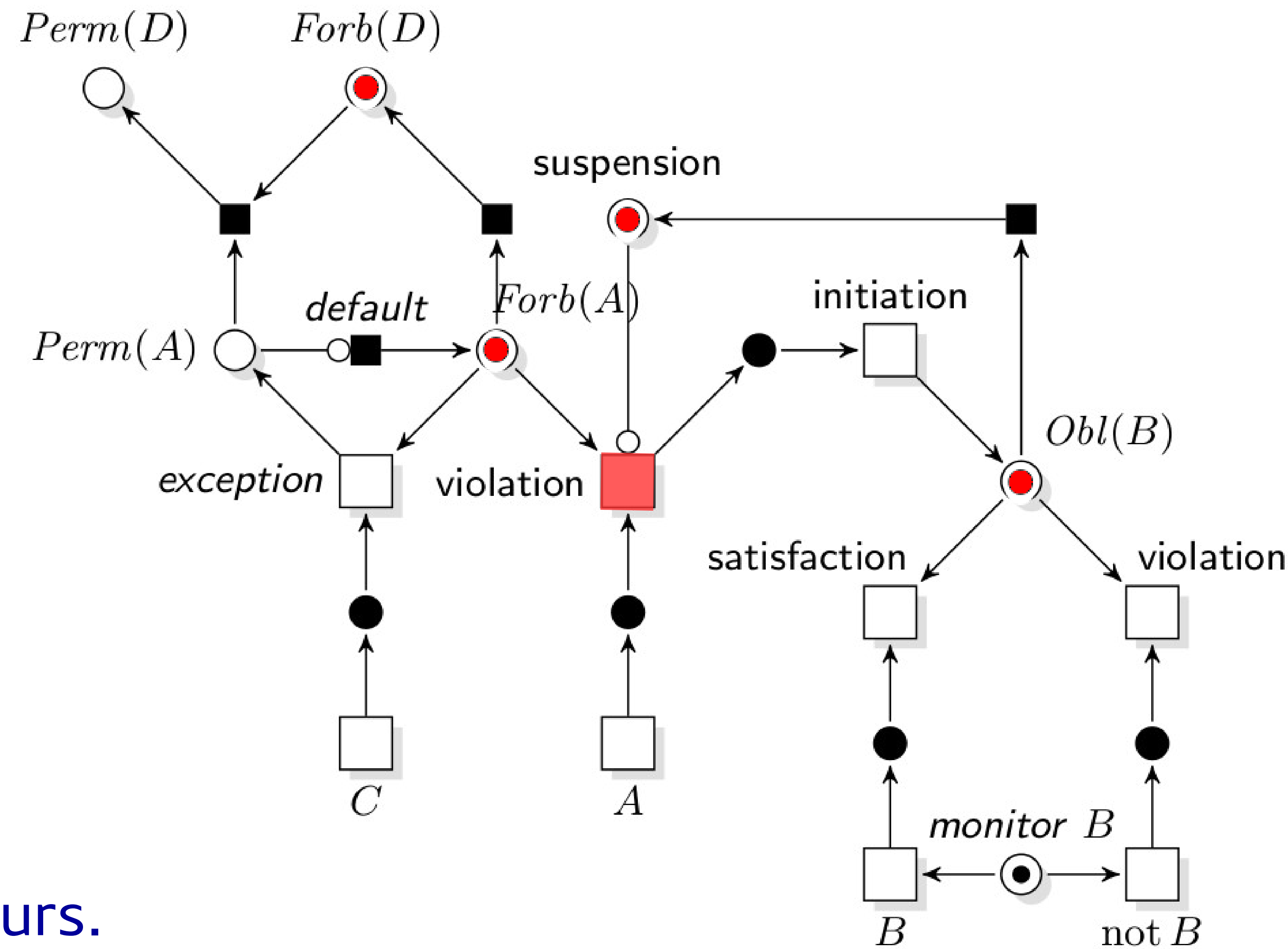


- *A* occurs.

Privacy Act

[Governatori, 2015]

Forbidden A. If C, then Permitted A.
If Forbidden A and A, then Obligatory B.
Forbidden D. If Permitted A, then Permitted D.

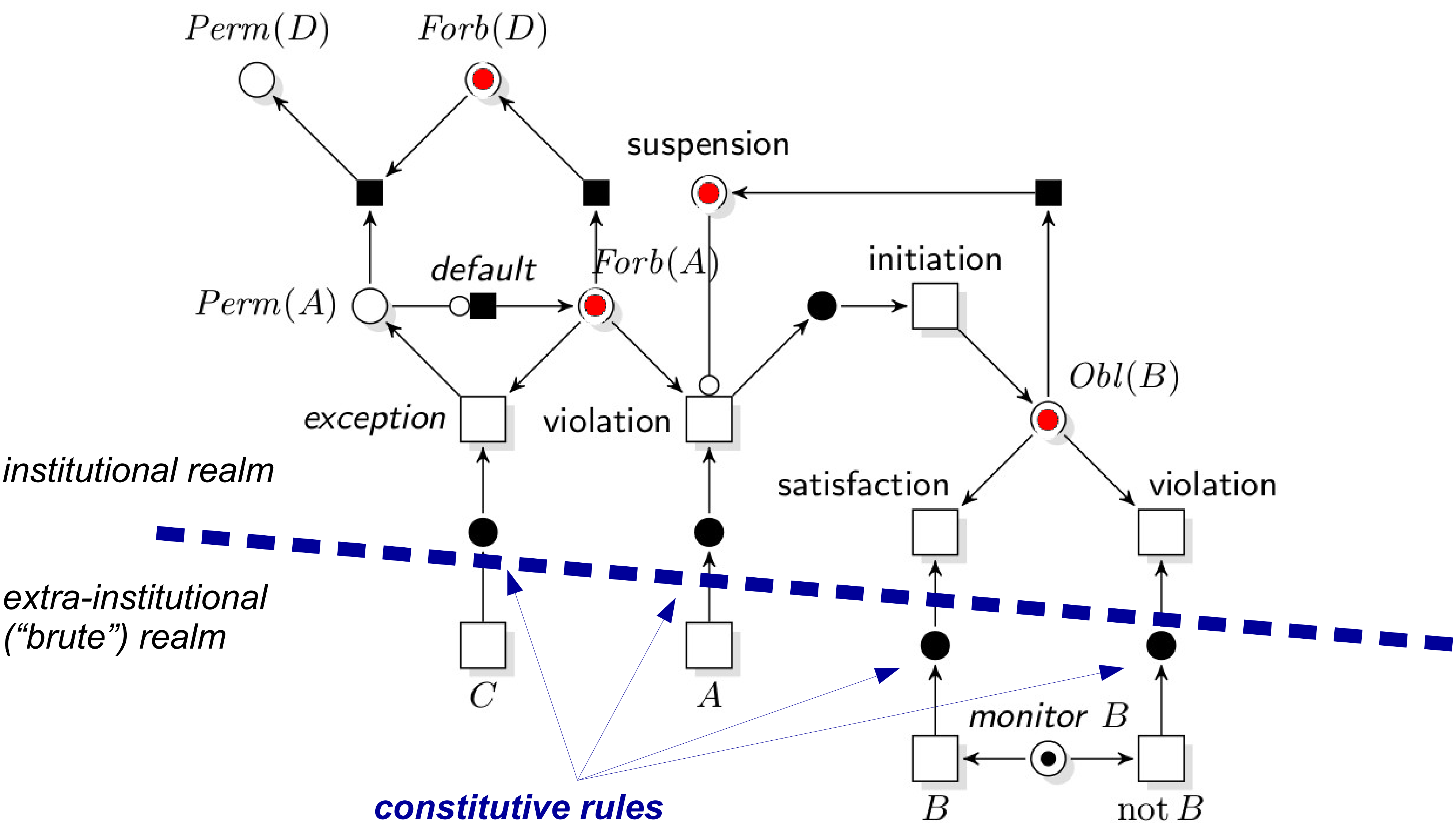


- *A* occurs.

Privacy Act

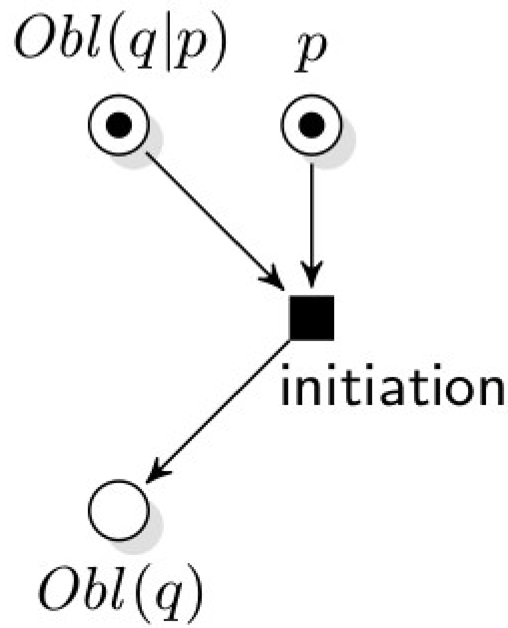
[Governatori, 2015]

Forbidden A. If C, then Permitted A.
If Forbidden A and A, then Obligatory B.
Forbidden D. If Permitted A, then Permitted D.



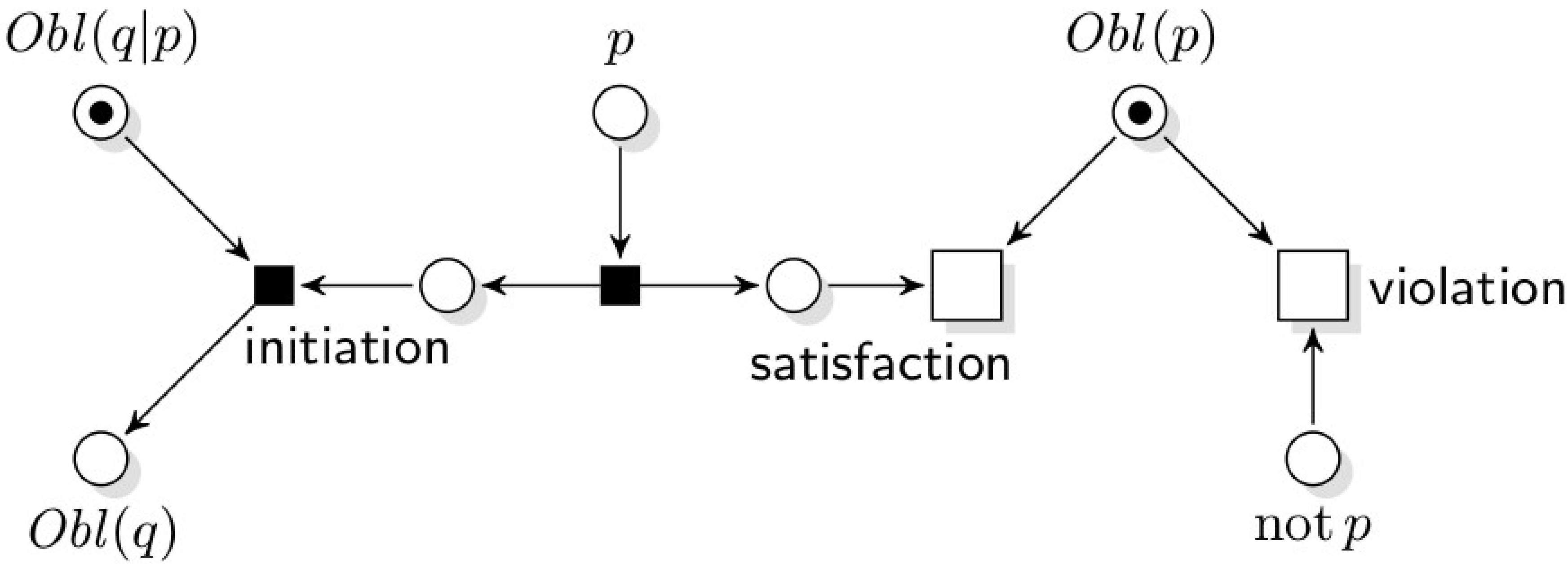
A bit further into deontic axioms

Factual detachment



$$p \wedge Obl\ (q|p) \rightarrow Obl\ (q)$$

Deontic detachment



$$Obl\ (p) \land Obl\ (q|p) \rightarrow Obl\ (q)$$

Not implied by LPPN! *It does not capture the **anticipation.***

Derived obligations

- *Bob's promise to meet you commits him to meeting you.*
- *It is obligatory that if Bob promises to meet you, he does so.*

Derived obligations

- *Bob's promise to meet you commits him to meeting you.*

  $p \rightarrow \mathrm{Obl}\ (m)$

- *It is obligatory that if Bob promises to meet you, he does so.*

  $\mathrm{Obl}\ (p \rightarrow m)$

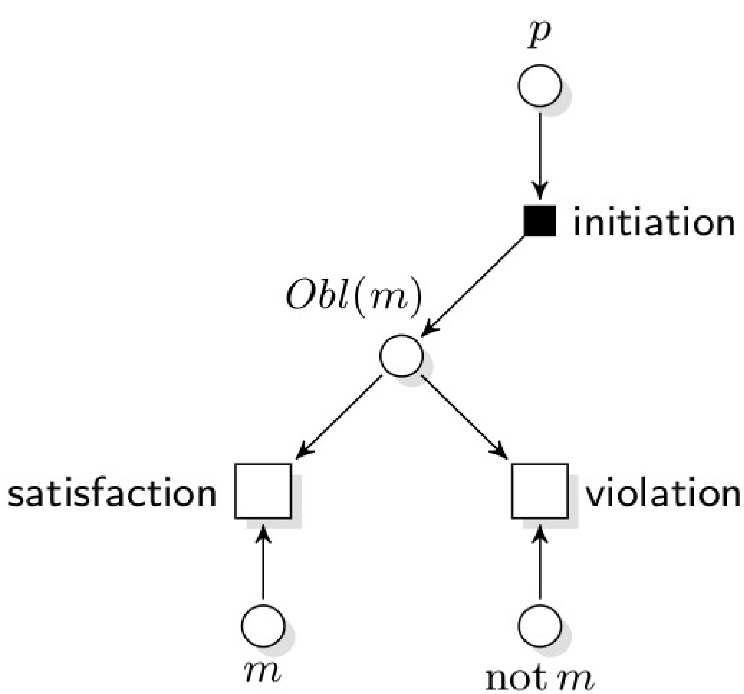
Derived obligations

- *Bob's promise to meet you commits him to meeting you.*

  `p → Obl (m)`

- *It is obligatory that if Bob promises to meet you, he does so.*

  `Obl (p → m)`



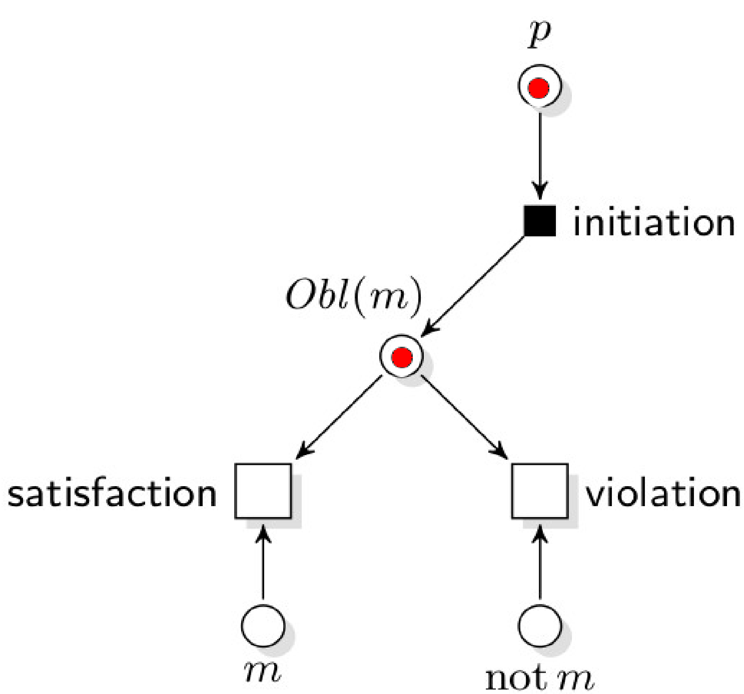
Derived obligations

- *Bob's promise to meet you commits him to meeting you.*

  `p → Obl (m)`

- *It is obligatory that if Bob promises to meet you, he does so.*

  `Obl (p → m)`



- p

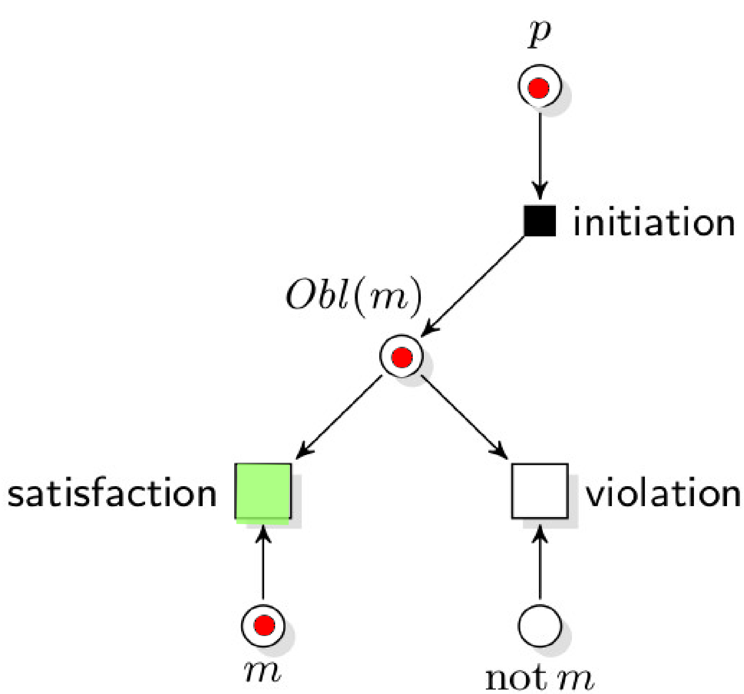
Derived obligations

- *Bob's promise to meet you commits him to meeting you.*

  `p → Obl (m)`

- *It is obligatory that if Bob promises to meet you, he does so.*

  `Obl (p → m)`



- p,
- either m..

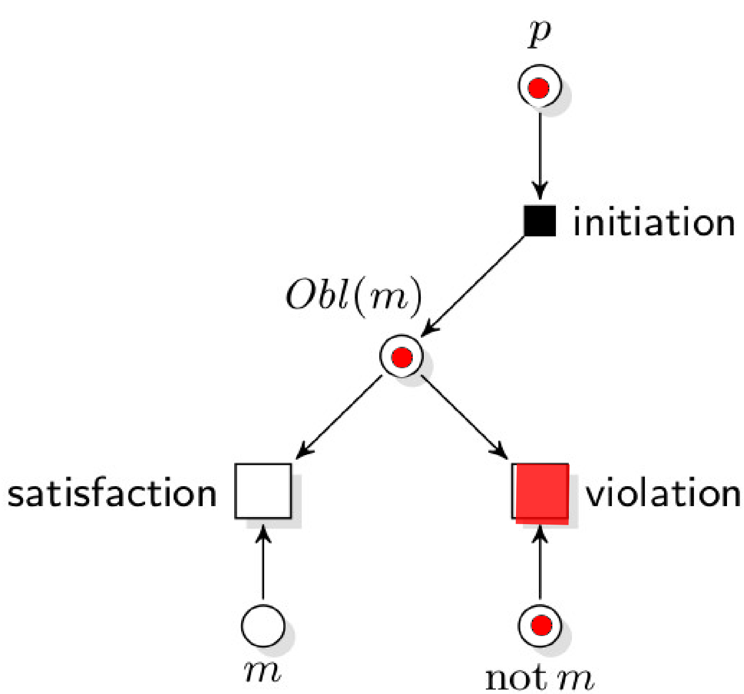
Derived obligations

- *Bob's promise to meet you commits him to meeting you.*

  `p → Obl (m)`

- *It is obligatory that if Bob promises to meet you, he does so.*

  `Obl (p → m)`



- p,
- either m, or ¬m

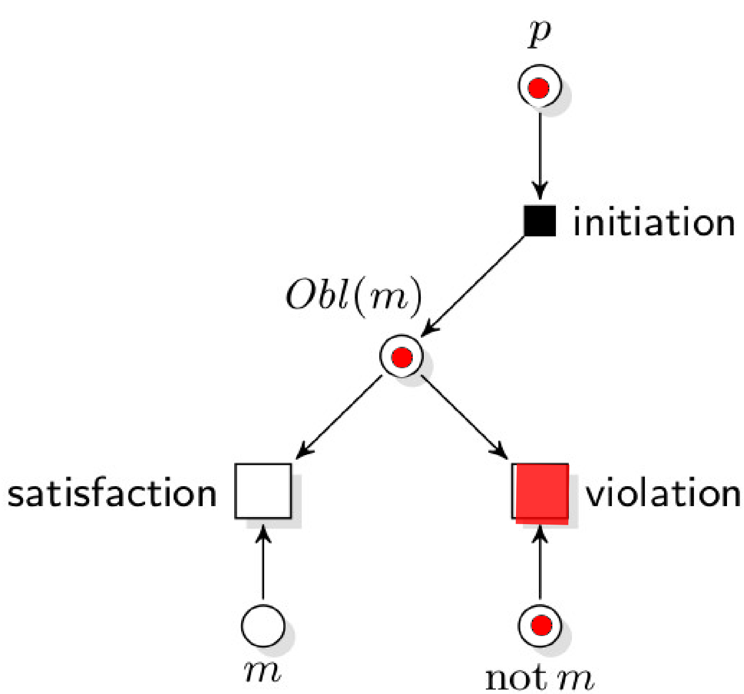
Derived obligations

- *Bob's promise to meet you commits him to meeting you.*

    `p → Obl (m)`

- *It is obligatory that if Bob promises to meet you, he does so.*

    `Obl (p → m)`



- p,
- either m, or ¬m

- *but what about ¬p?*

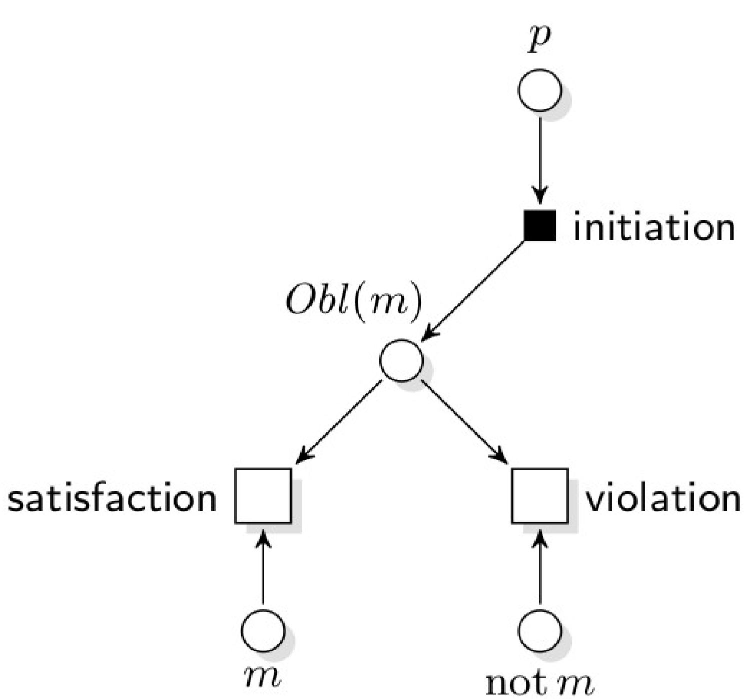
Derived obligations

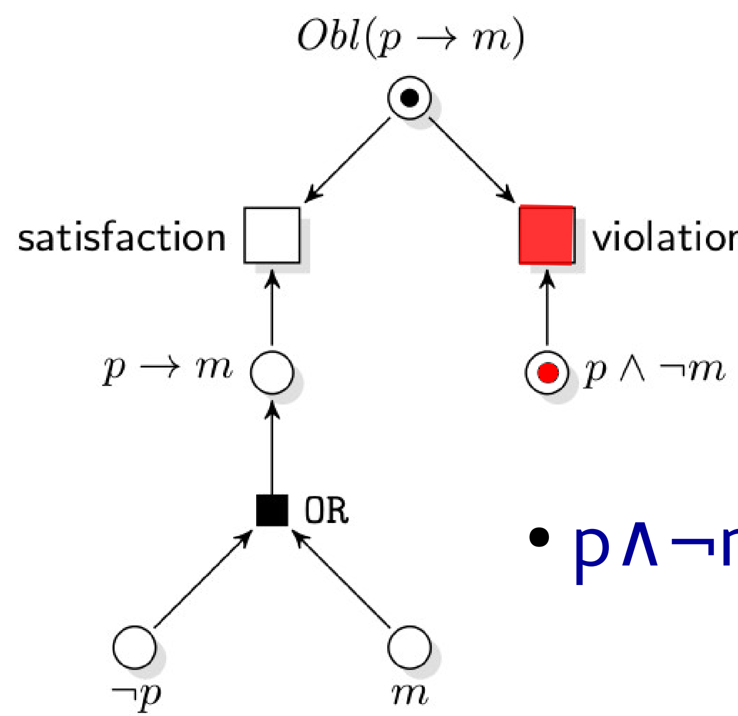
- *Bob's promise to meet you commits him to meeting you.*

  `p → Obl (m)`

- *It is obligatory that if Bob promises to meet you, he does so.*

  `Obl (p → m)`





- $p \wedge \neg m$

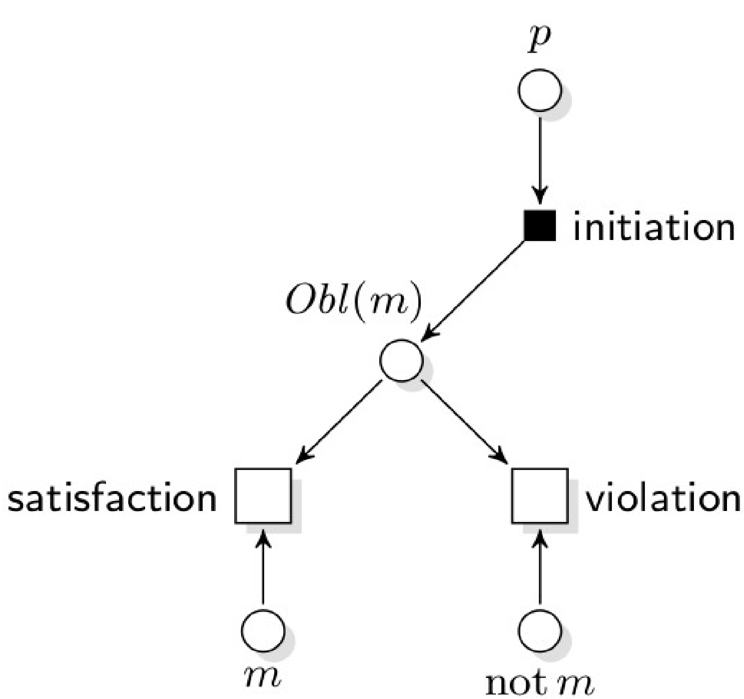
Derived obligations

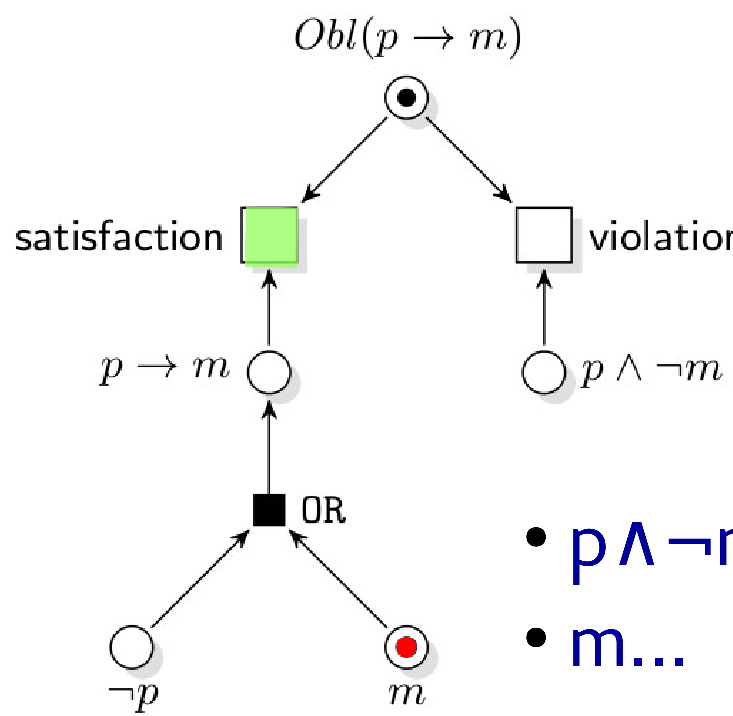
- *Bob's promise to meet you commits him to meeting you.*

  `p → Obl (m)`

- *It is obligatory that if Bob promises to meet you, he does so.*

  `Obl (p → m)`





- $p \wedge \neg m$
- m...

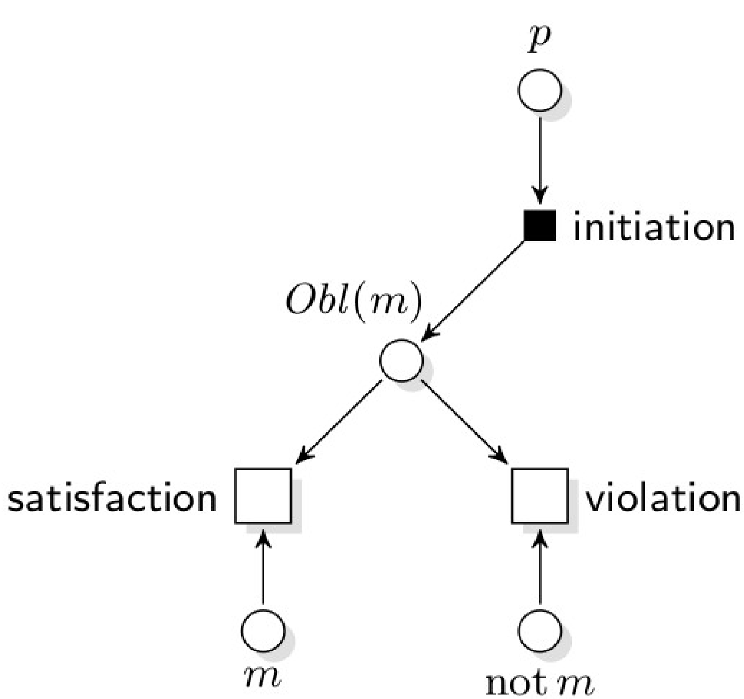
Derived obligations

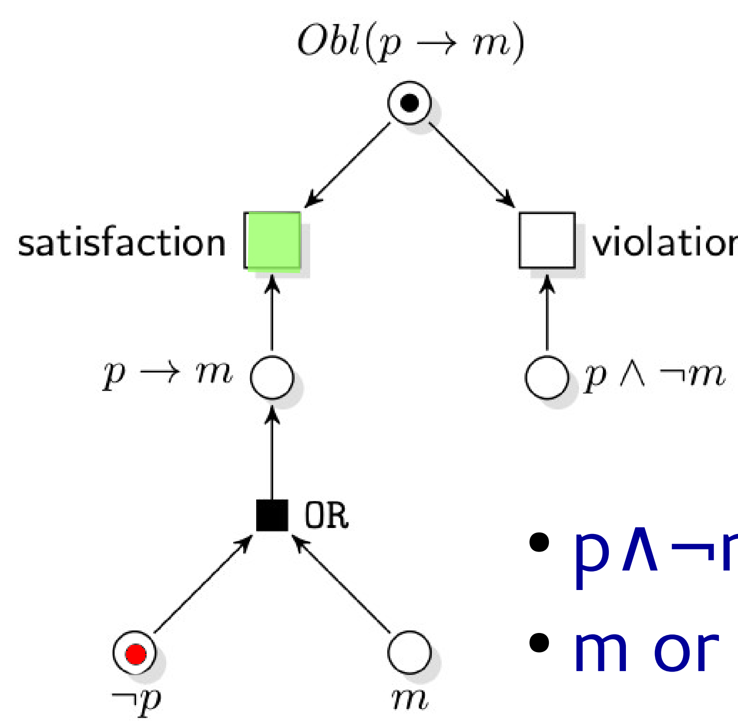
- *Bob's promise to meet you commits him to meeting you.*

  `p → Obl (m)`

- *It is obligatory that if Bob promises to meet you, he does so.*

  `Obl (p → m)`





- p∧¬m
- m or ¬p

Chisholm's paradox

[Chisholm, 1963]

Chisholm's paradox

[Chisholm, 1963]

- *It ought to be that Jones goes (to the assistance of his neighbors).*
- *It ought to be that if Jones goes, then he tells them he is coming.*
- *If Jones doesn't go, then he ought not tell them he is coming.*
- *Jones doesn't go.*

Chisholm's paradox

[Chisholm, 1963]

```
Obl (go)
Obl (go → tell)
¬go → Forb(tell)
¬go
```

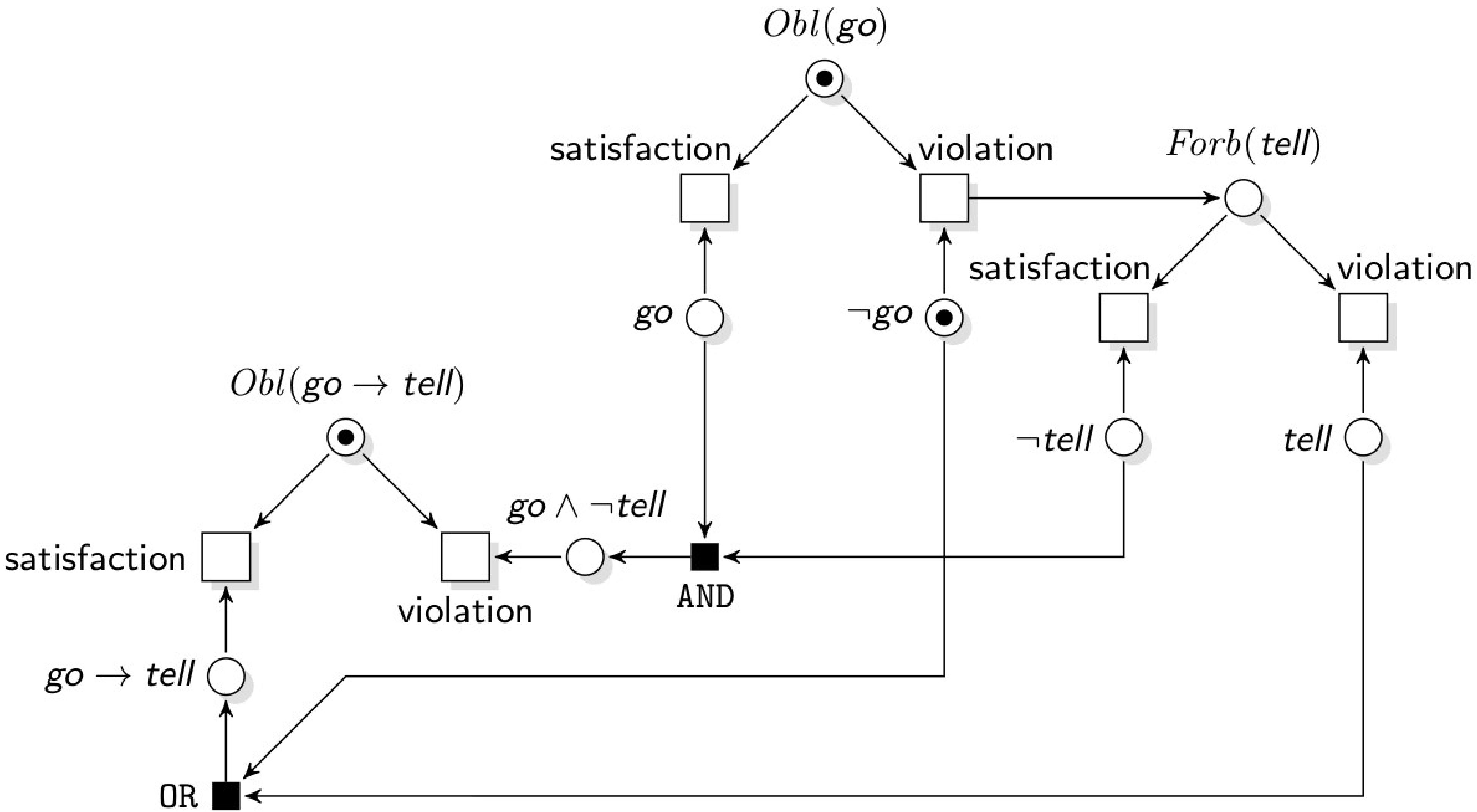
- *It ought to be that Jones goes (to the assistance of his neighbors).*
- *It ought to be that if Jones goes, then he tells them he is coming.*
- *If Jones doesn't go, then he ought not tell them he is coming.*
- *Jones doesn't go.*

Chisholm's paradox

[Chisholm, 1963]

```
Obl (go)
Obl (go → tell)
¬go → Forb(tell)
¬go
```

- You don't go.

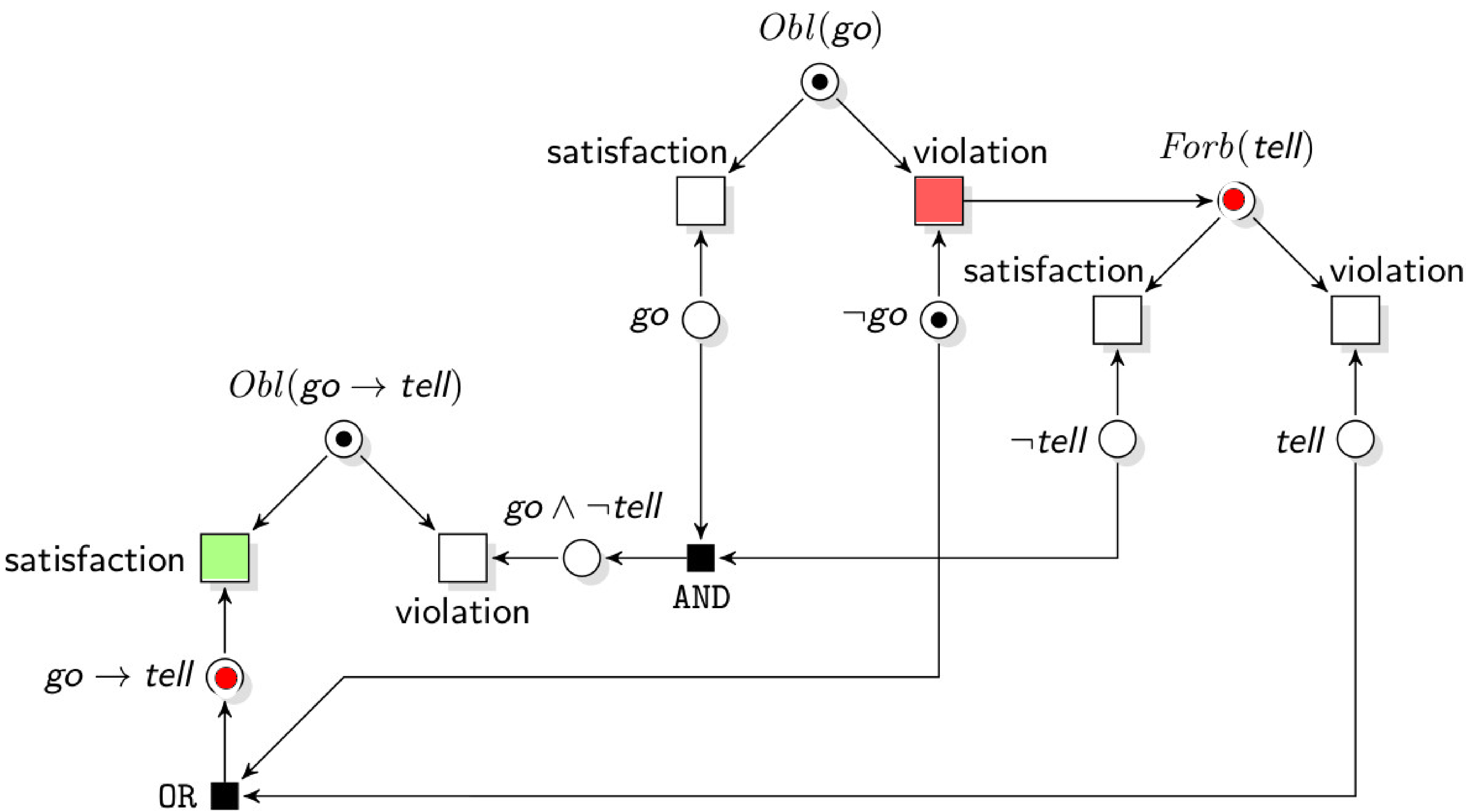


Chisholm's paradox

[Chisholm, 1963]

```
Obl (go)
Obl (go → tell)
¬go → Forb(tell)
¬go
```

- You don't go.

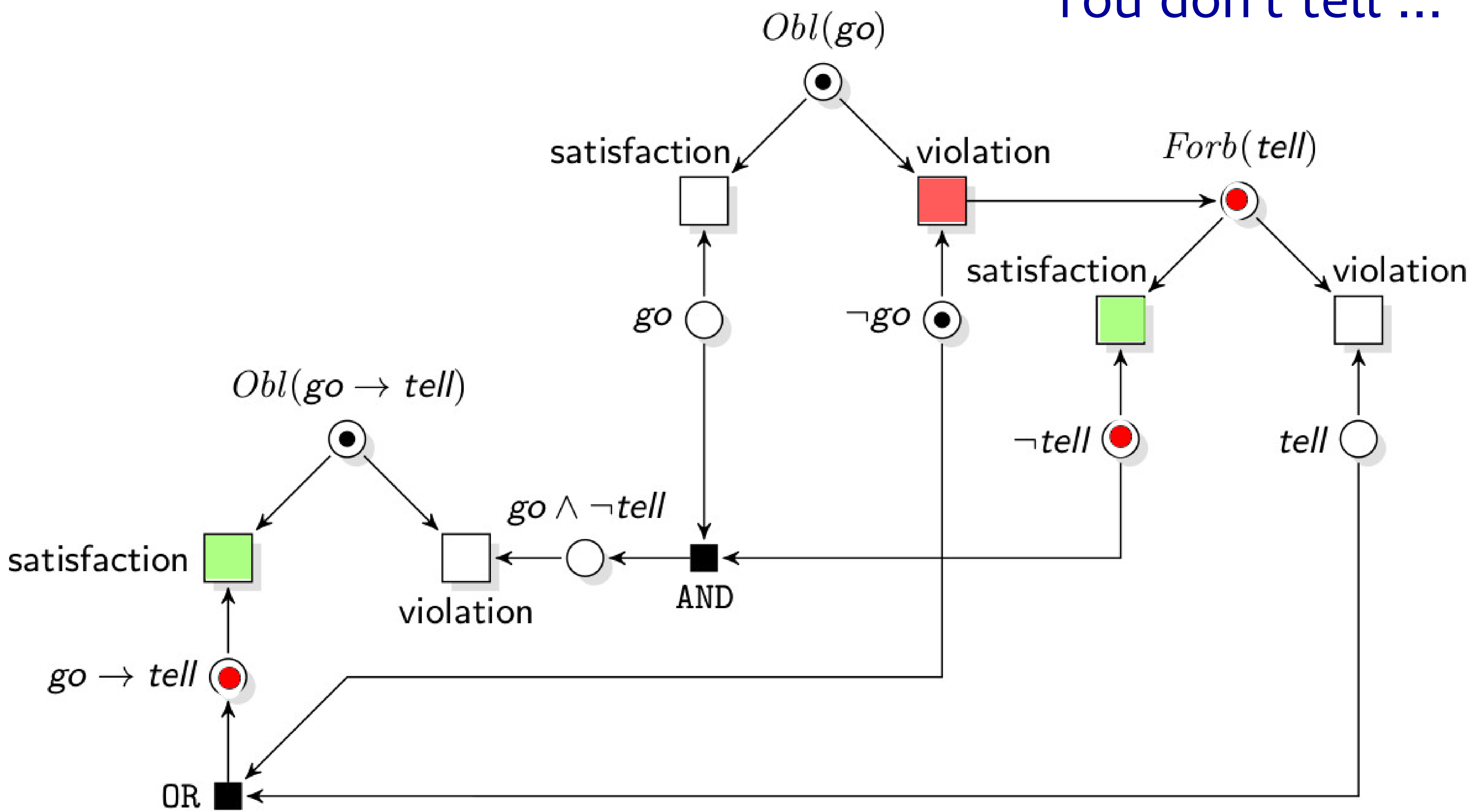


Chisholm's paradox

[Chisholm, 1963]

```
Obl (go)
Obl (go → tell)
¬go → Forb(tell)
¬go
```

- You don't go ...
  You don't tell ...

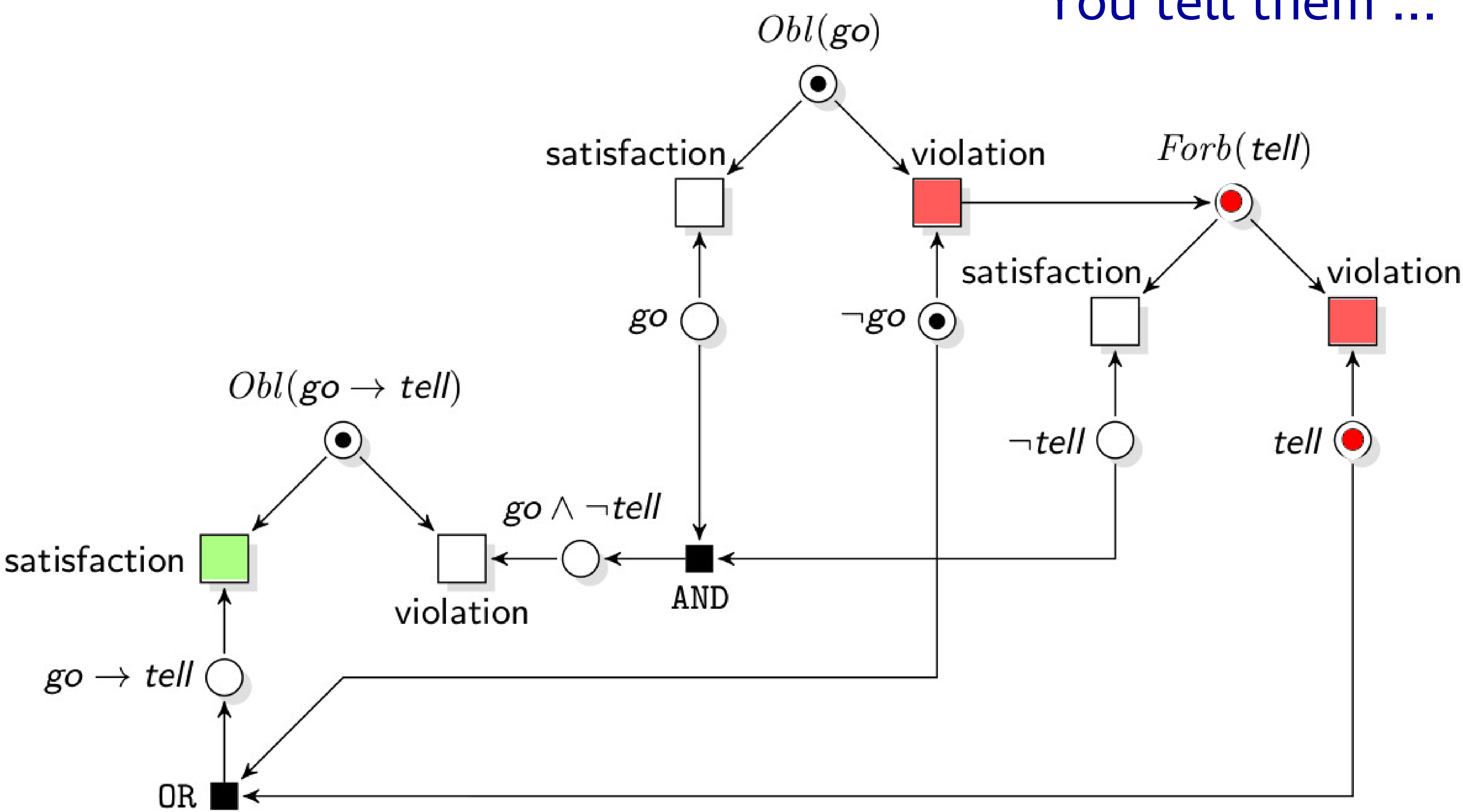


Chisholm's paradox

[Chisholm, 1963]

```
Obl (go)
Obl (go → tell)
¬go → Forb(tell)
¬go
```

- You don't go ...
  You tell them ...



Conclusions

Conclusions

- We presented examples of application of a notation introduced for wider modeling purposes, but which can be applied for normative scenarios:

Conclusions

- We presented examples of application of a notation introduced for wider modeling purposes, but which can be applied for normative scenarios:
  - paradoxical cases are easily unveiled

Conclusions

- We presented examples of application of a notation introduced for wider modeling purposes, but which can be applied for normative scenarios:
  - paradoxical cases are easily unveiled
  - direct link with business process practices

Conclusions

- We presented examples of application of a notation introduced for wider modeling purposes, but which can be applied for normative scenarios:
  - paradoxical cases are easily unveiled
  - direct link with business process practices
  - visual notation (animation of models through simulation make them in principle more accessible)

Conclusions

- We presented examples of application of a notation introduced for wider modeling purposes, but which can be applied for normative scenarios.
- Defaults, exceptions, suspensions required extensions.

Conclusions

- We presented examples of application of a notation introduced for wider modeling purposes, but which can be applied for normative scenarios.
- Defaults, exceptions, suspensions required extensions.
- ~ **minimal commitment** (cf. *input/output* logic)

$$\mathrm{Obl}\,(A) \wedge \mathrm{Obl}\,(B) \ \ ?? \ \ \mathrm{Obl}(A \wedge B)$$

Conclusions

- We presented examples of application of a notation introduced for wider modeling purposes, but which can be applied for normative scenarios.
- Defaults, exceptions, suspensions required extensions.
- Interesting semantic correspondence with ***language***
  - nouns, (imperfect) verbs ~ places/tokens
  - (perfect) verbs ~ transition/transition events

Conclusions

- We presented examples of application of a notation introduced for wider modeling purposes, but which can be applied for normative scenarios.
- Defaults, exceptions, suspensions required extensions.
- Interesting semantic correspondence with ***language***
  - nouns, (imperfect) verbs ~ places/tokens
  - (perfect) verbs ~ transition/transition events
- Working hypothesis: does the locutor usually provide the right granularity to avoid the paradox?