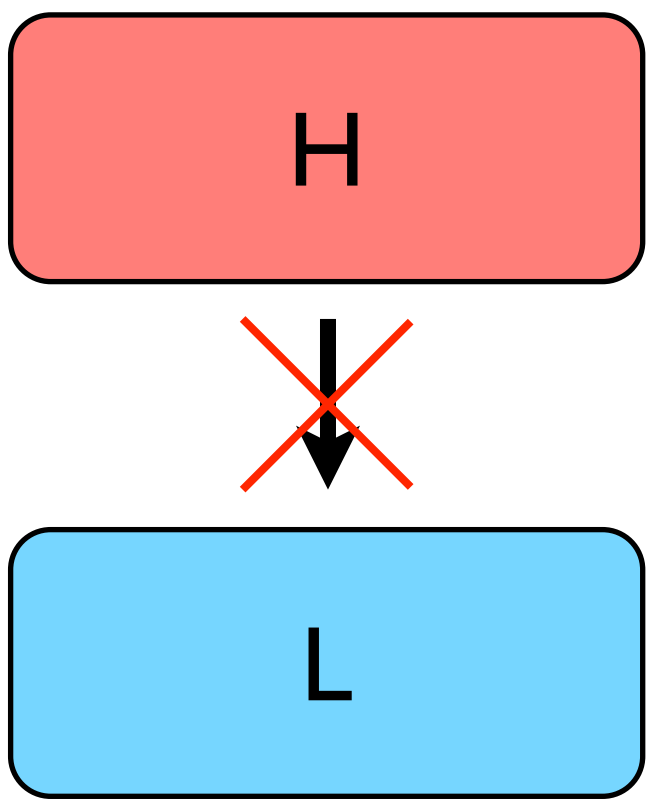
A Causal View of Non Interference

Paolo Baldan - Univ. Padova

Joint work with
Alberto Carraro - Univ. Ca' Foscari Venezia

Non interference

- Absence of undesired **information flows** between entities of a computer system



Access Control

- **Origin**:
  Access control policies for protecting the secrecy of user data
- **DAC** (discretionary access control)
  - subjects control access rights to their objects (Unix model)
  - malicious code (Trojan horse) can make user information globally visible

Multilevel security

- **MAC** (mandatory access control)
  - subjects and objects have a security level
  - system policy: no read-up, no write down
- Malicious code can leak **implicitly** information to lower level alterating the system behaviour (deadlocks, buffer full)
- Covert channels

Non Interference

- Control the **flow of information**, rather than the access of subjects to objects

Imperative languages

- Variables classified as low/high level
- The content of low level variables should not be influenced by high-level variables

Imperative languages

- Variables classified as low/high level
- The content of low level variables should not be influenced by high-level variables

```
h = l
```

ok

Imperative languages

- Variables classified as low/high level
- The content of low level variables should not be influenced by high-level variables

`h = l`   ok

`l = h`   no

Imperative languages

- Variables classified as low/high level
- The content of low level variables should not be influenced by high-level variables

```
h = l
```

ok

```
l = h
```

no

```
if h>0 then l++
       else l--
```

no

Imperative languages

- Variables classified as low/high level
- The content of low level variables should not be influenced by high-level variables

```
h = l
```

ok

```
l = h
```

no

```
while h != 0 do
    h++
```

no

Process calculi

- Concurrent and distributed setting

Process calculi

- Concurrent and distributed setting

- Action labels:

  $$Act = L \uplus H$$

Activities at different levels of confidentiality

Process calculi

- Concurrent and distributed setting

- Action labels:

$$Act = L \uplus H$$

Activities at different levels of confidentiality

- Observational semantics:

$$\approx$$

traces, bisimilarity, ...

Process calculi

- **Idea**: Any behaviour involving high level activities should be possible also without

- **Low-view** observational semantics

$$\approx_L$$

View of the low level user

- **Non interference** (NDC)

$$Sys \approx_L C_H[Sys] \qquad \forall C_H[\cdot]$$

Process calculi

- Non interference (NDC)

  $$Sys \approx_L (Sys \mid Sys_H) \setminus H$$

- Examples

Process calculi

- Non interference (NDC)

$$Sys \approx_L (Sys \mid Sys_H) \setminus H$$

- Examples

$$P = h.\,l.\,0$$

$$P \not\approx_L (P \mid 0) \setminus H$$

Process calculi

- Non interference (NDC)

$$Sys \approx_L (Sys \mid Sys_H) \setminus H$$

- Examples

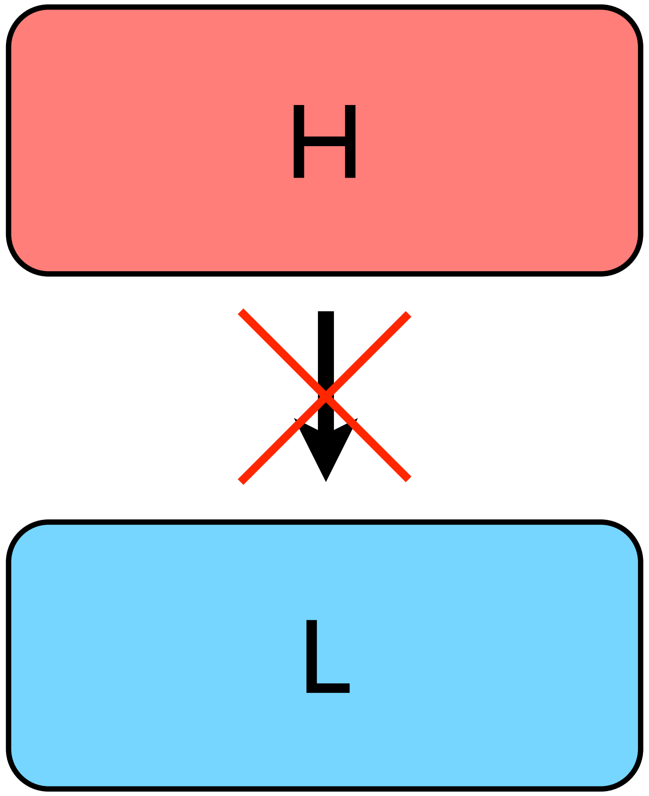
$$P = h.\,l.\,0$$

$$P \not\approx_L (P \mid 0) \setminus H$$

$$P' = h.\,l.\,0 + l.0$$

Non interference

- Absence of undesired **information flows** from H to L
- H does not **cause** visible effects on L



- Refers to an informal idea of causality, formalised in terms of interleaving semantics

Causality?

- A true concurrent / causal formalisation?
- Recover known non interference notions
  - conceptual clarity
  - efficiency (alleviate state space explosion)
- Rely on causal semantics for capturing stricter notions of non interference

Outline

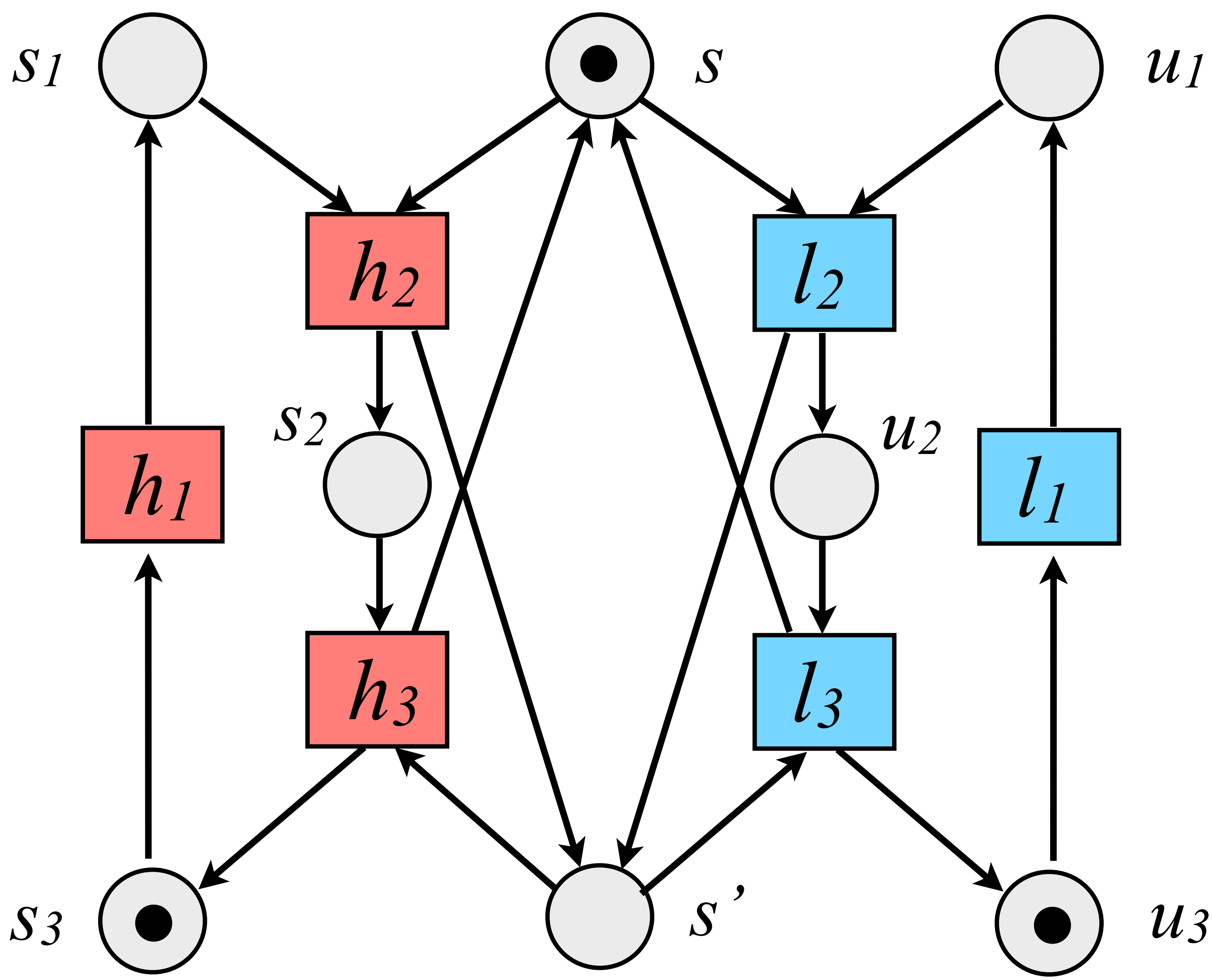
- Petri nets
- BNDC

[Busi, Gorrieri],
[Best, Darondeaux, Gorrieri]

- Semistructural characterisation
- Causal characterisation (based on the unfolding semantics)

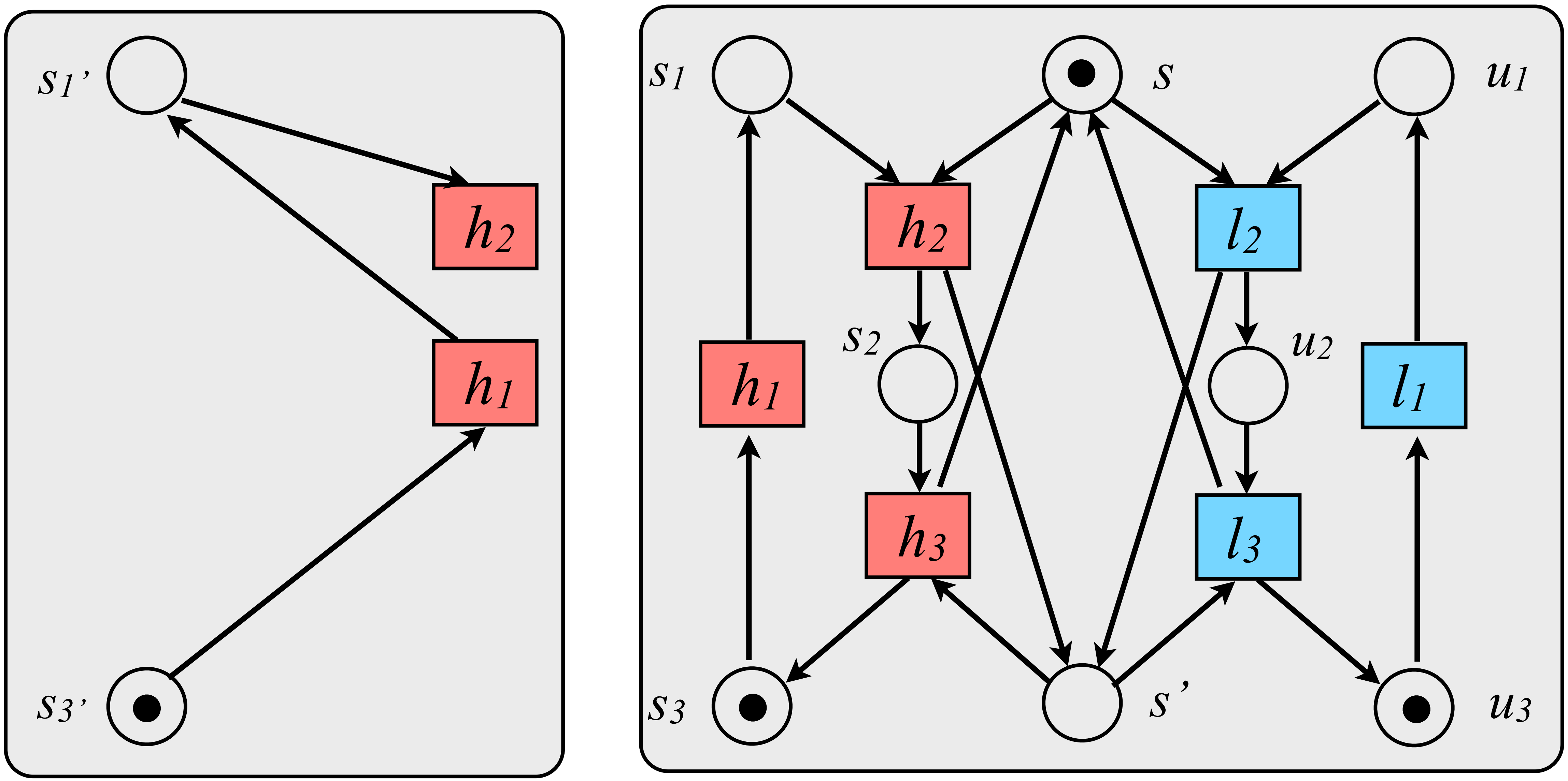
Petri nets

- High and low transitions $T = L \uplus H$



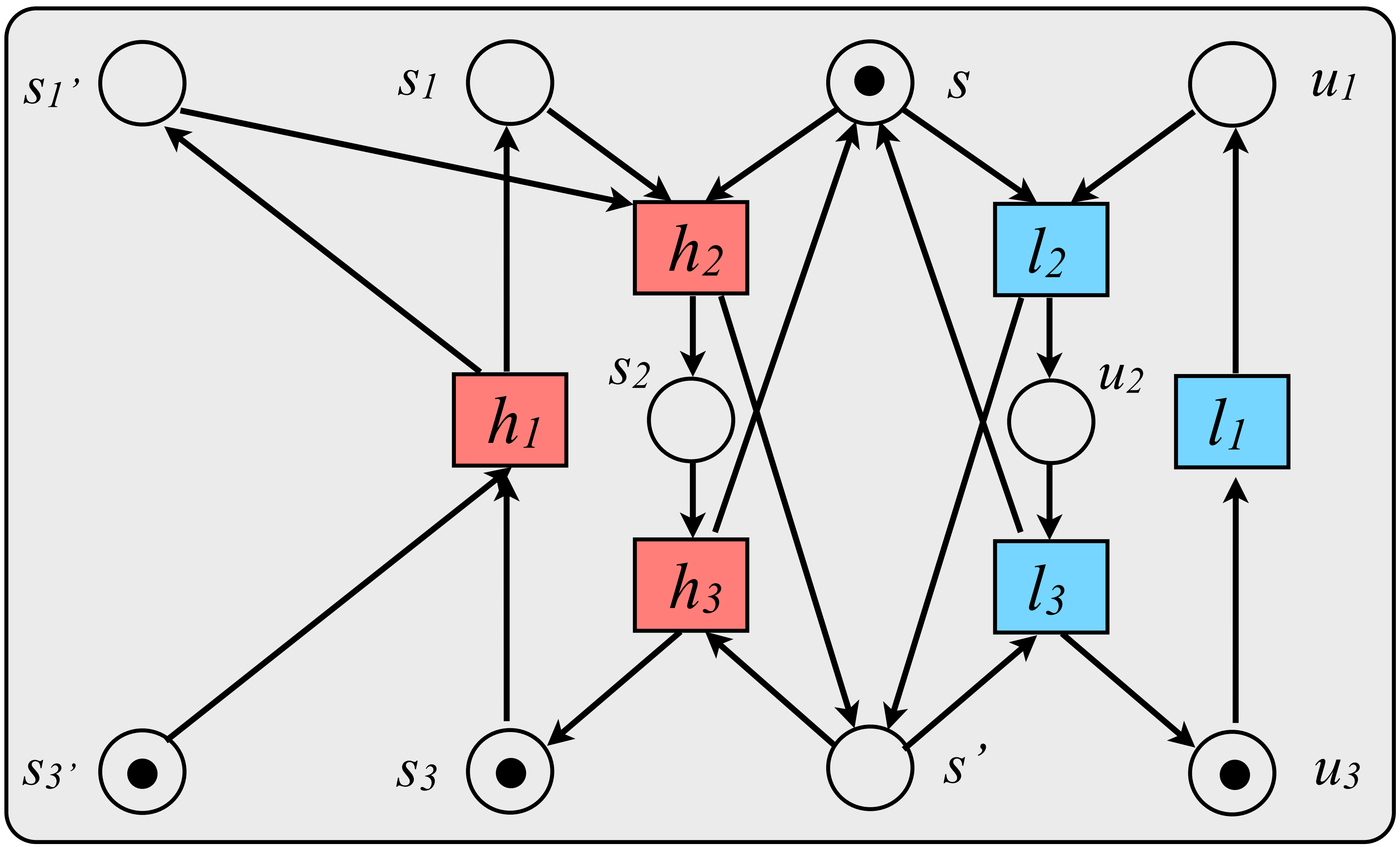
Parallel composition

- *N* | *N'* :parallel composition, synchronising on common transitions



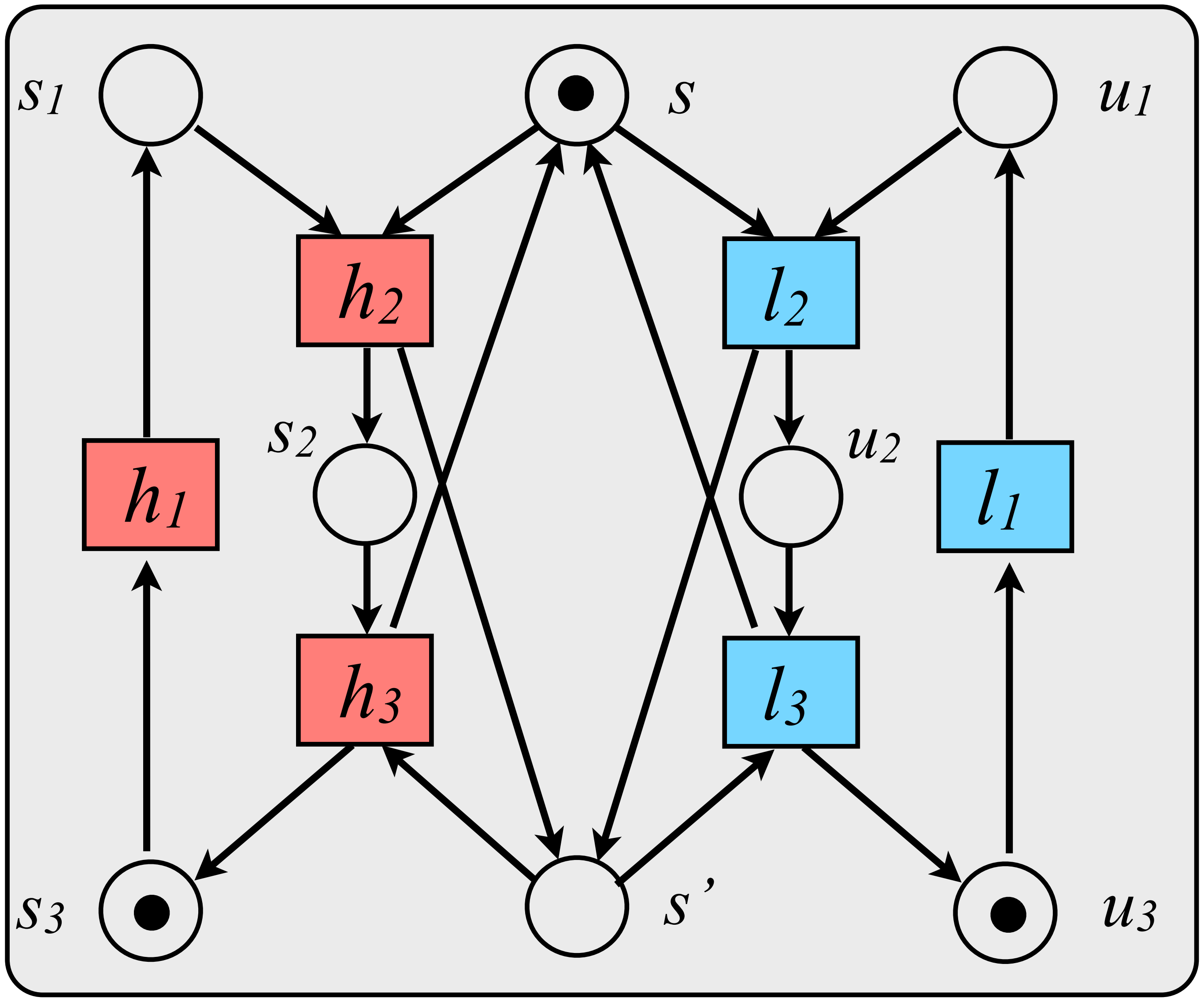
Parallel composition

- N | N' :parallel composition, synchronbising on common transitions



Restriction

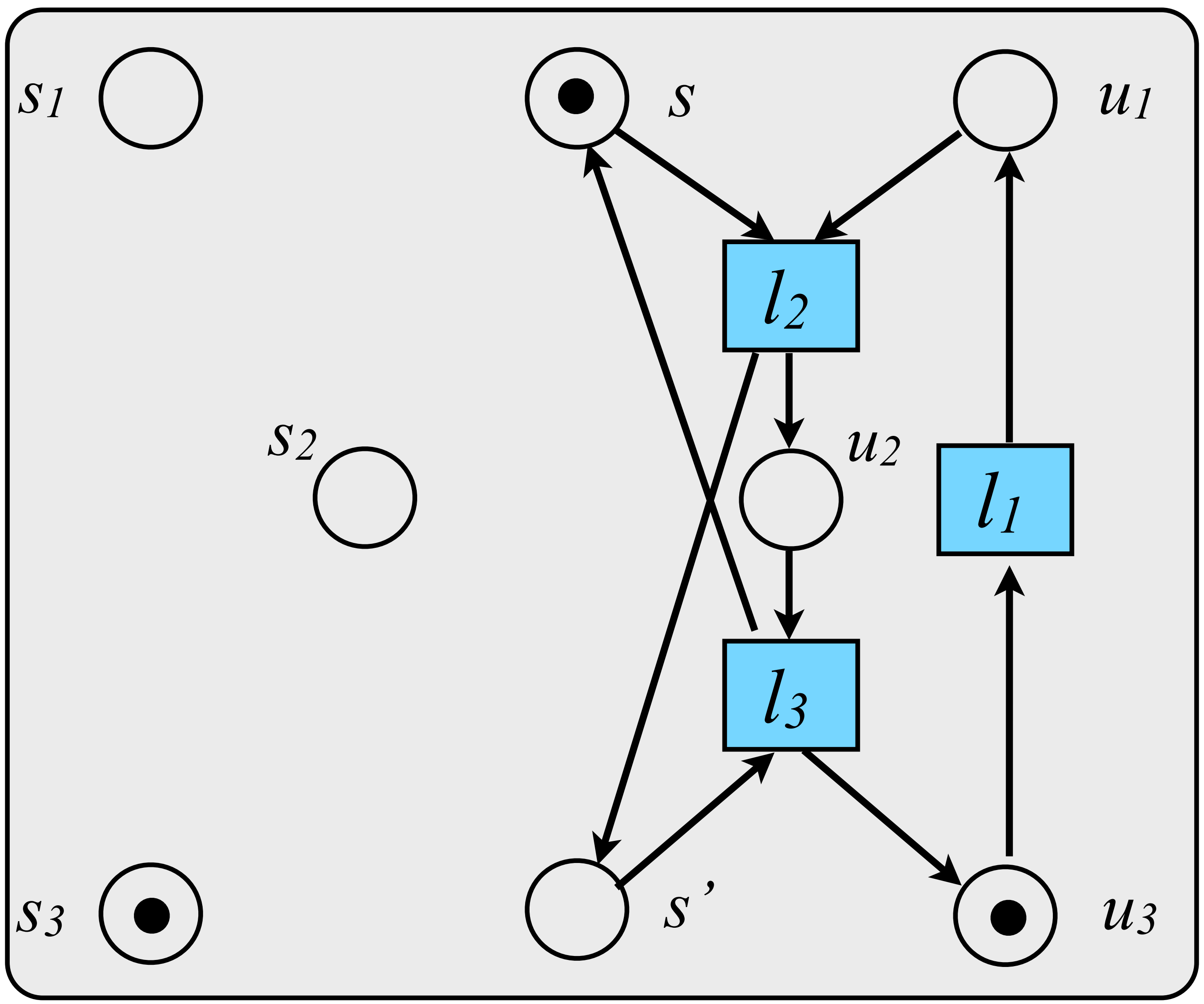
- N \ X: remove from N transitions in X



\ H

Restriction

- N \ X: remove from N transitions in X



BNDC

- A system N is **BNDC** if for any high-level system K

$$N \ \approx_L \ (N \mid K) \setminus (H - H_K)$$

[Busi, Gorrieri]

BNDC

- A system N is **BNDC** if for any high-level system K

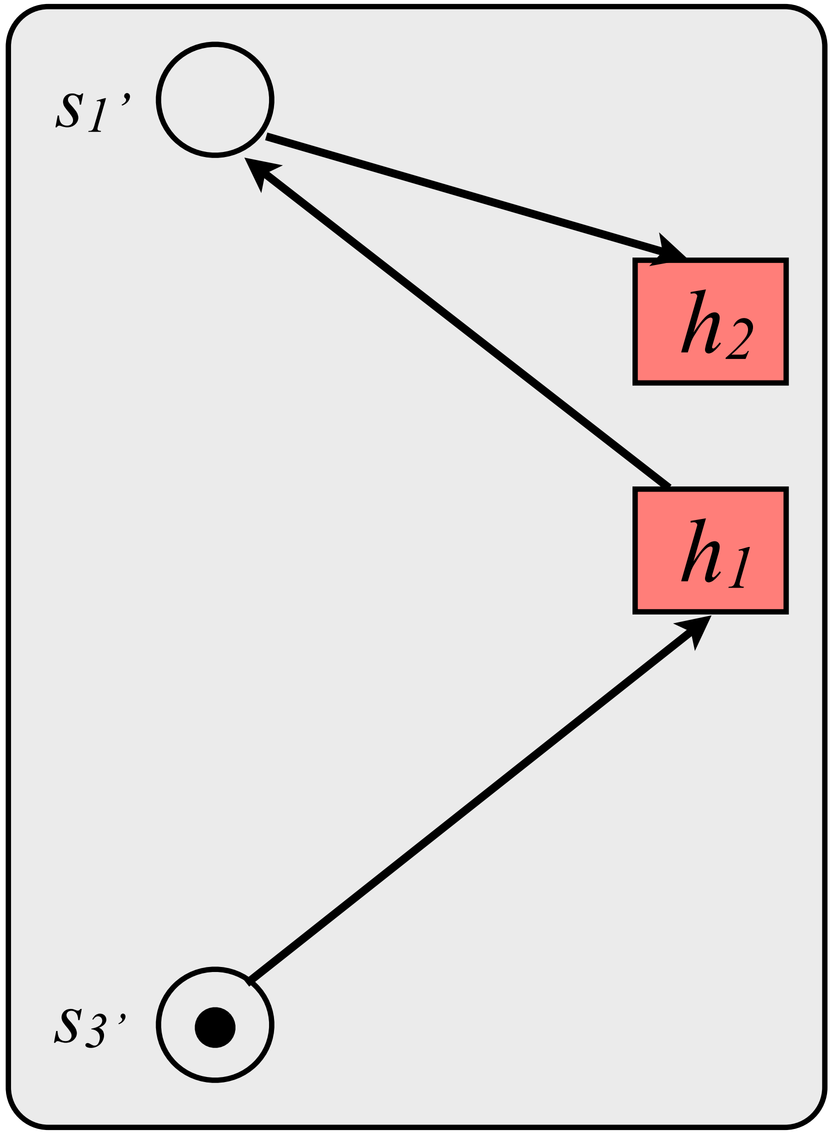
$$N \ \approx_L \ (N \mid K) \setminus (H - H_K)$$

low-view bisimulation
(high-level actions are invisible)

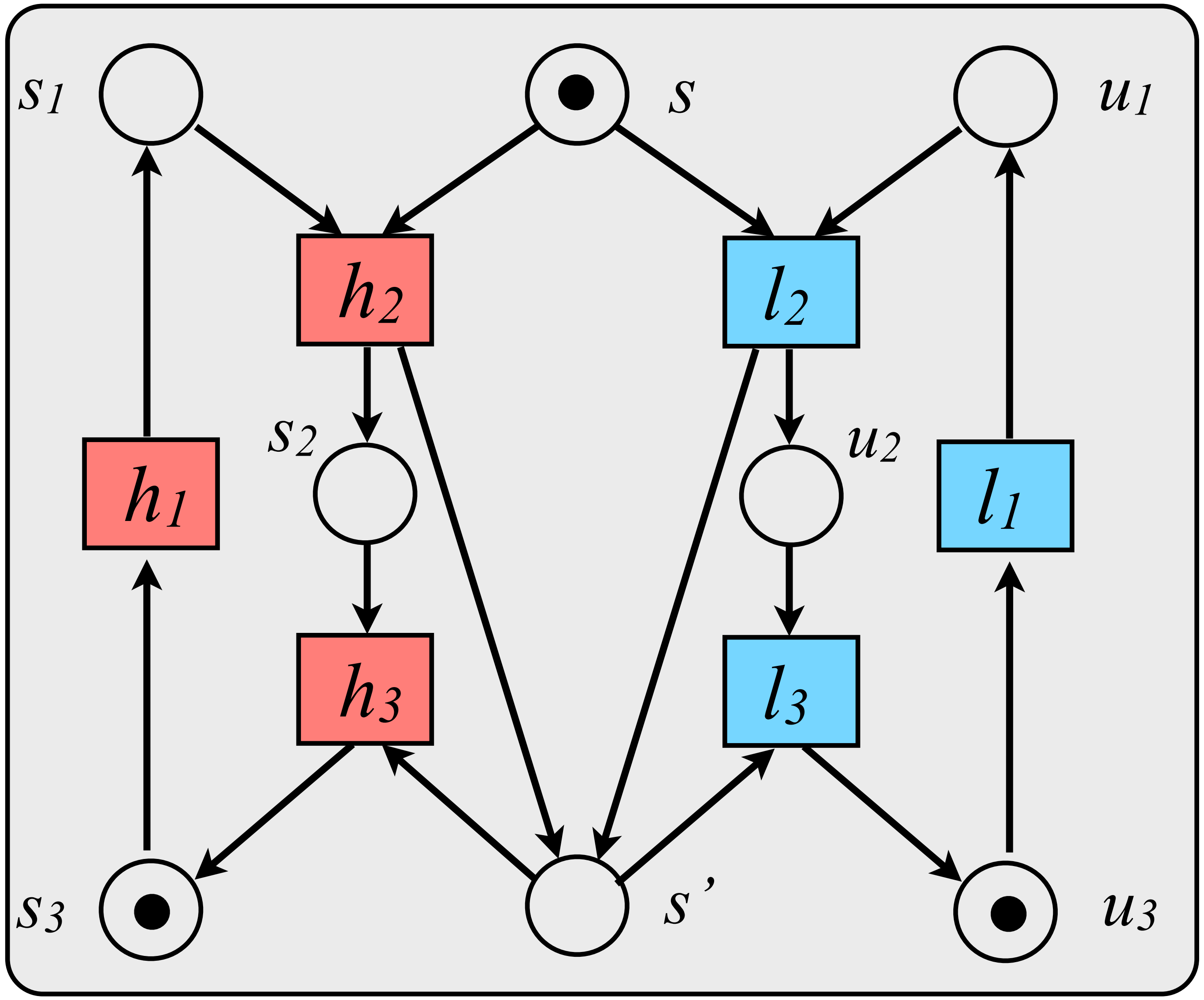
[Busi, Gorrieri]

Example

K

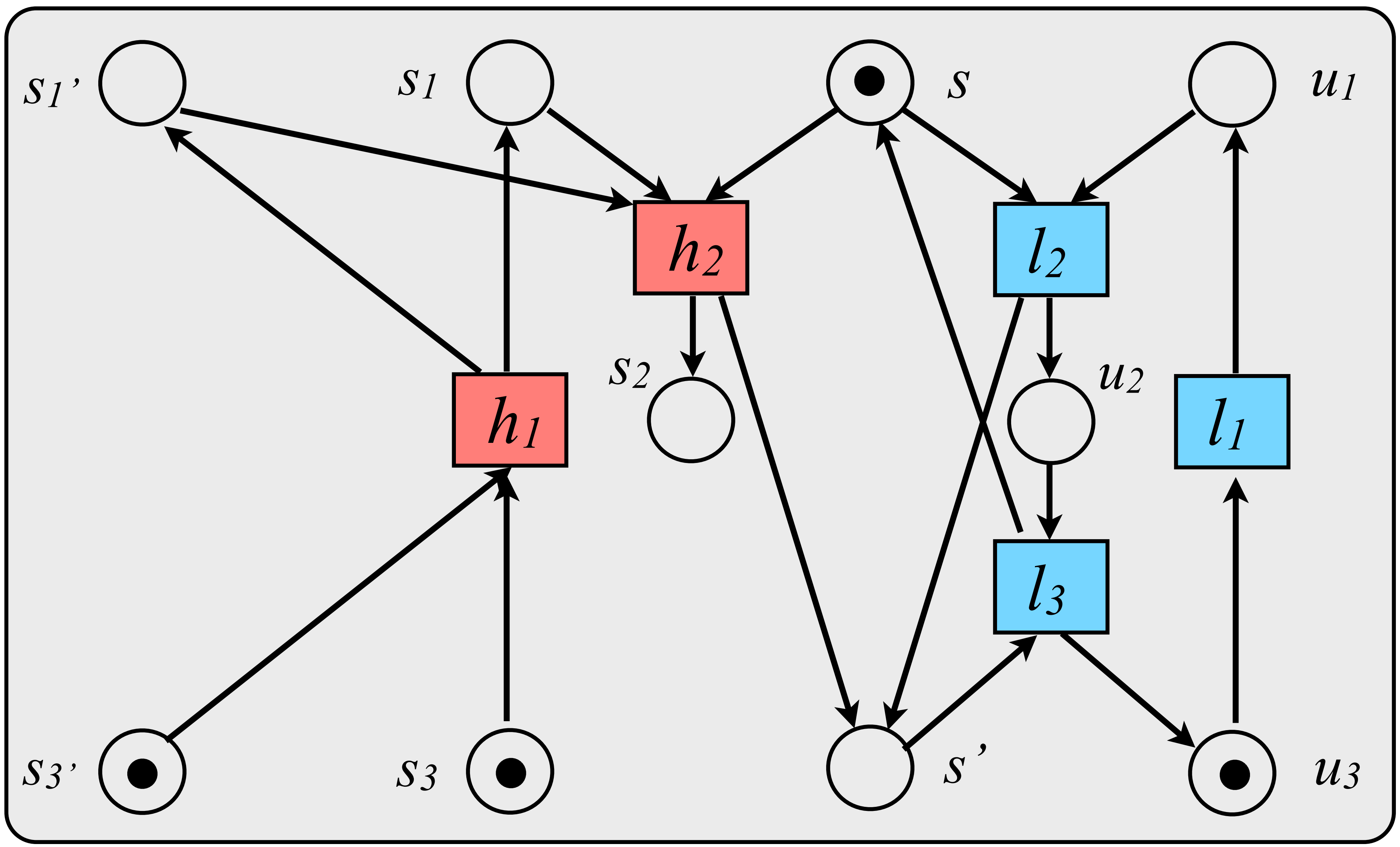


N



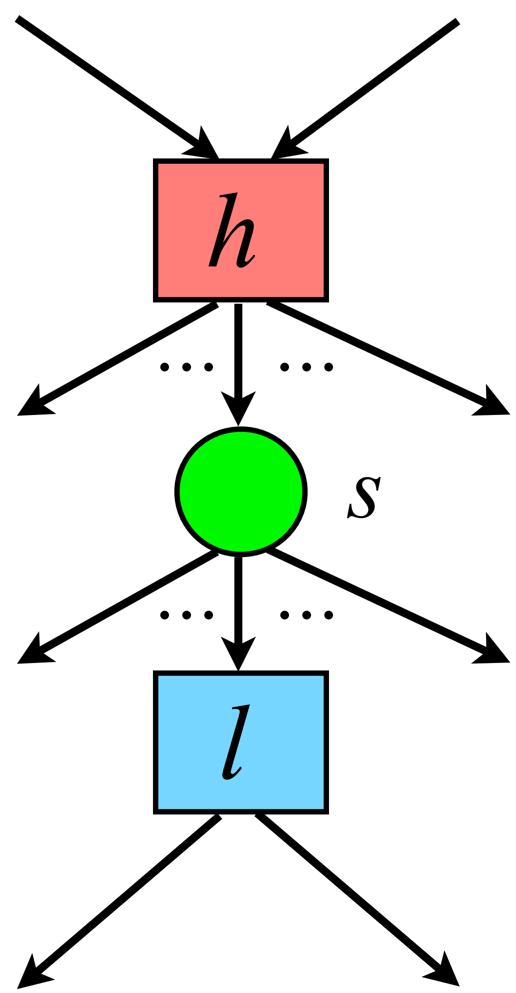
Example

$$(K \mid N) \setminus (H - H_K)$$



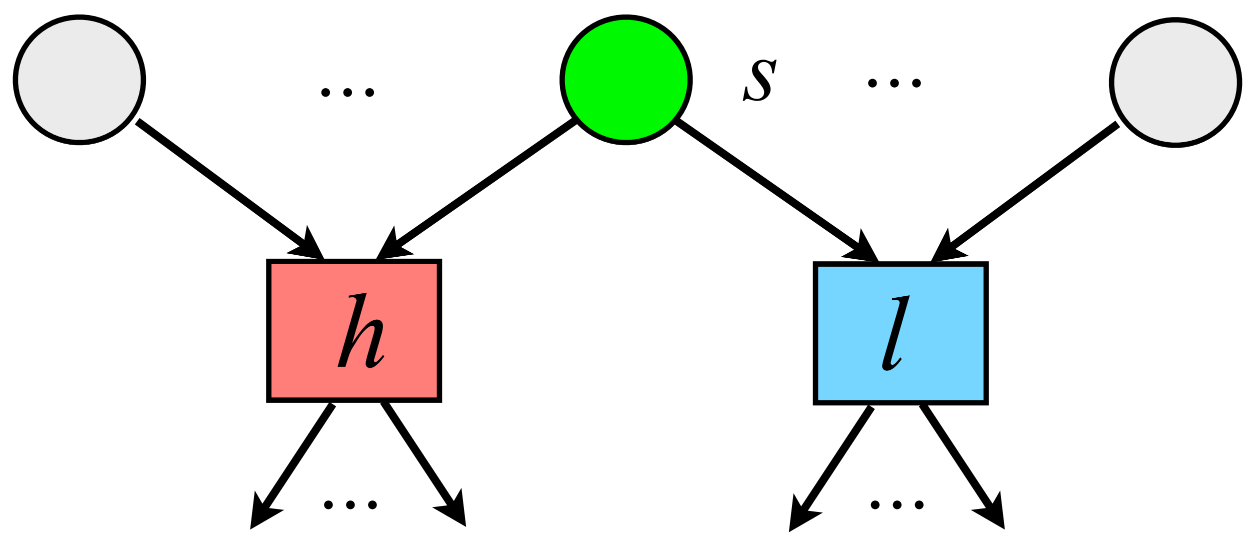
Potential interferences

- Potential **causal** place



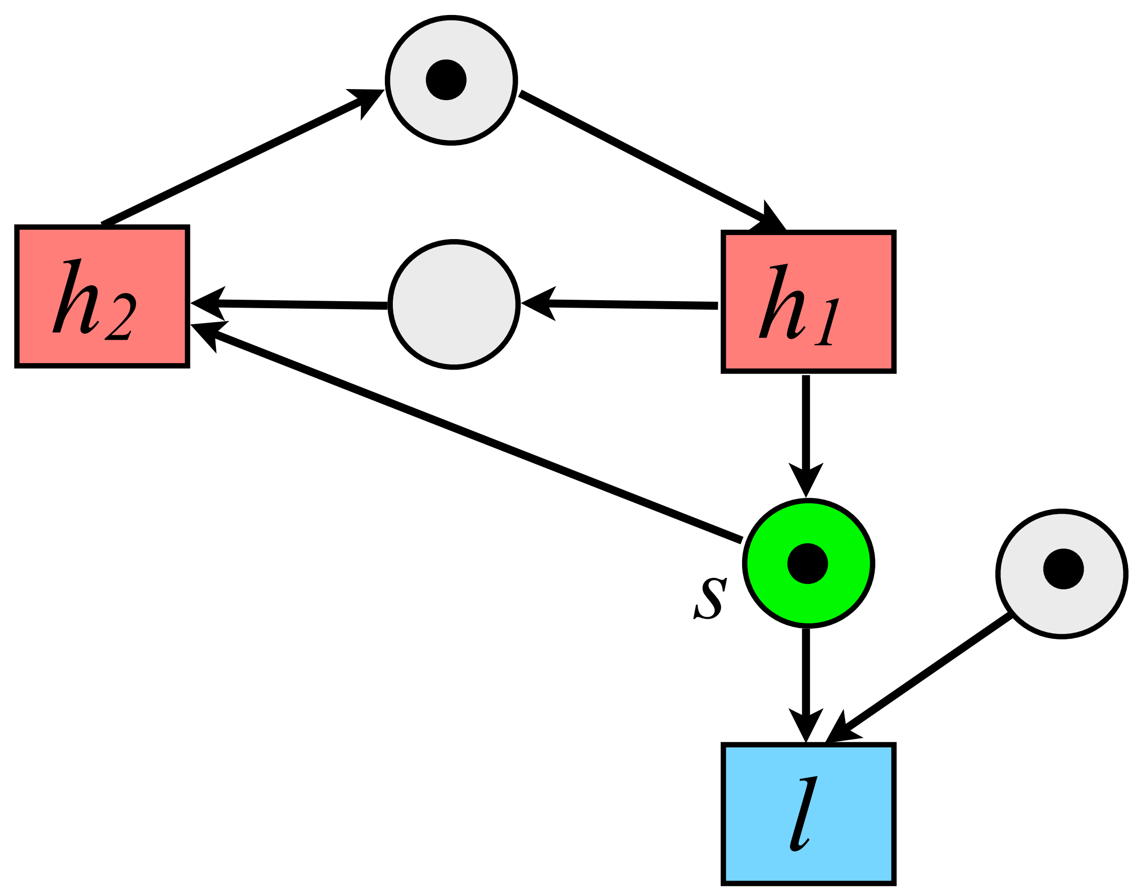
Potential interferences

- Potential **conflict** place



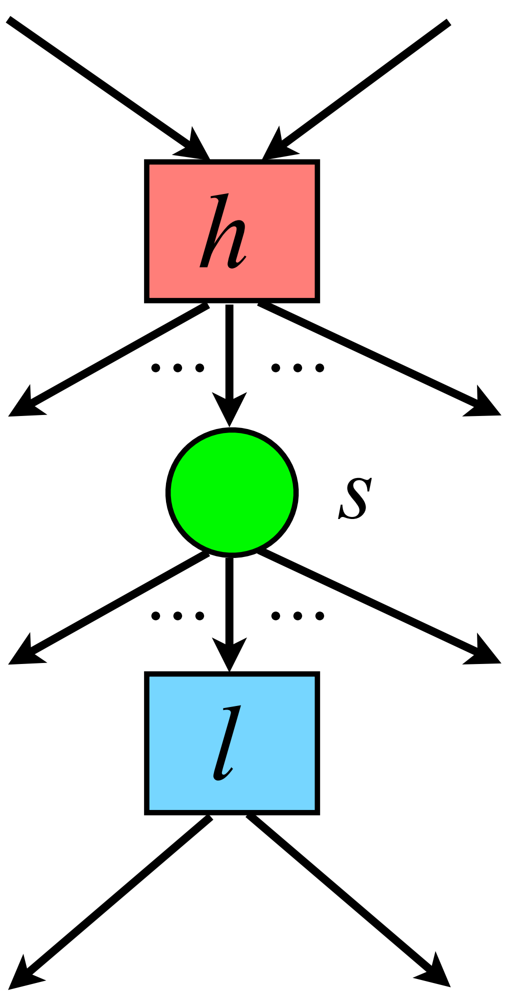
Potential interferences

- Not always a problem



Active causal places

- There is a computation in which l necessarily uses in s a token generated by h



$$\exists m \text{ reachable}$$

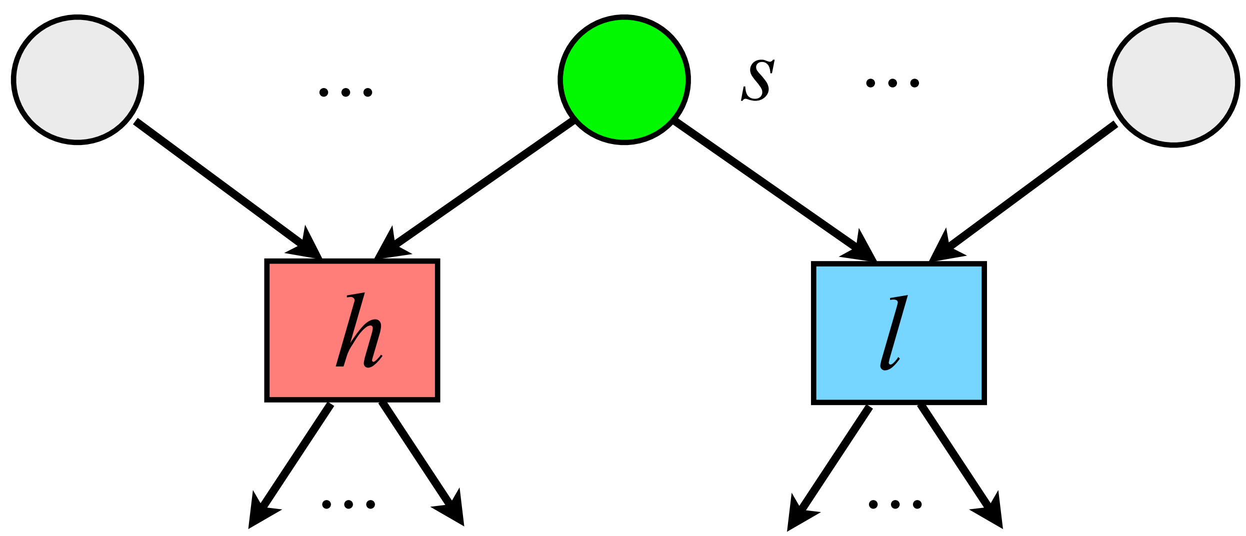
$$m[h\, t_1 \ldots t_k\, l\rangle$$

$$m[t_1 \ldots t_k\rangle m'$$

$$m'(s) < {}^{\bullet}l(s)$$

Active conflict places

- There is a computation in which necessarily l competes with h for a token in s



$\exists m$ reachable

$$m[h\, t_1 \ldots t_k\rangle m'$$

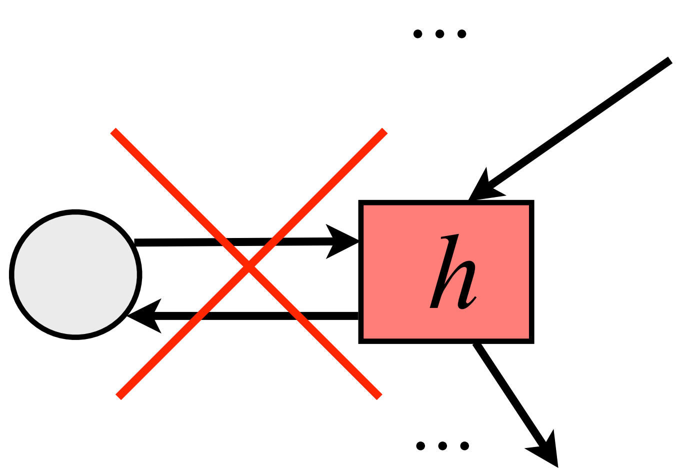
$$m[t_1 \ldots t_k\, l\rangle \qquad m'(s) < {}^{\bullet}l(s)$$

BNDC from active places

Theorem: N is BNDC iff
it does not include active conflict or causal places

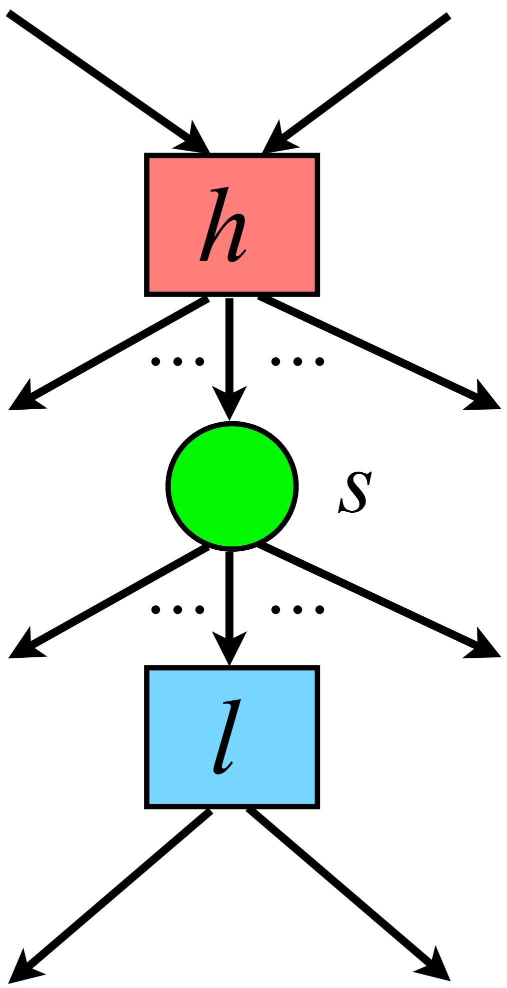
A more efficient characterisation

- If self-loops are forbidden



A more efficient characterisation

- Active causal place



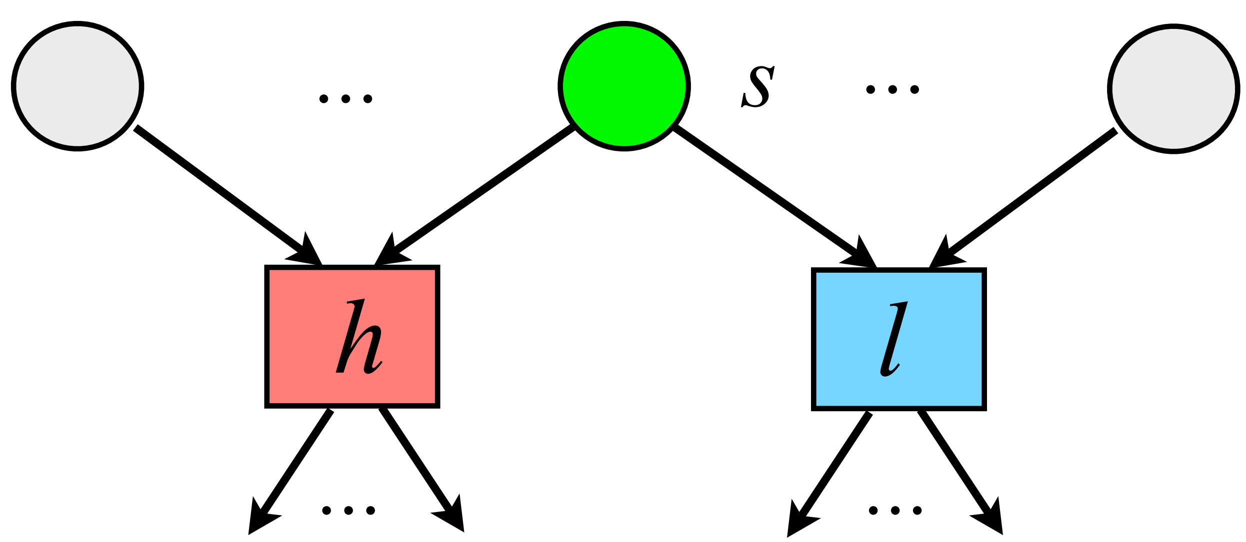
$\exists m$ reachable

$m[h\,l\rangle$

$m(s) < {}^{\bullet}l(s)$

A more efficient characterisation

- Active conflict place



$$\exists m \text{ reachable}$$

$$m[h\rangle m' \qquad m'(s) < {}^{\bullet}l(s)$$
$$m[l\rangle$$

Checking non interference on the MG

- **For bounded PNs**:
  inspection of the the Marking Graph
  $O(2^{|S|})$

- **Original algorithm** [BG]
  a modified Marking Graph for any marking which covers a potential causal/conflict place
  $O(2^{2|S|})$

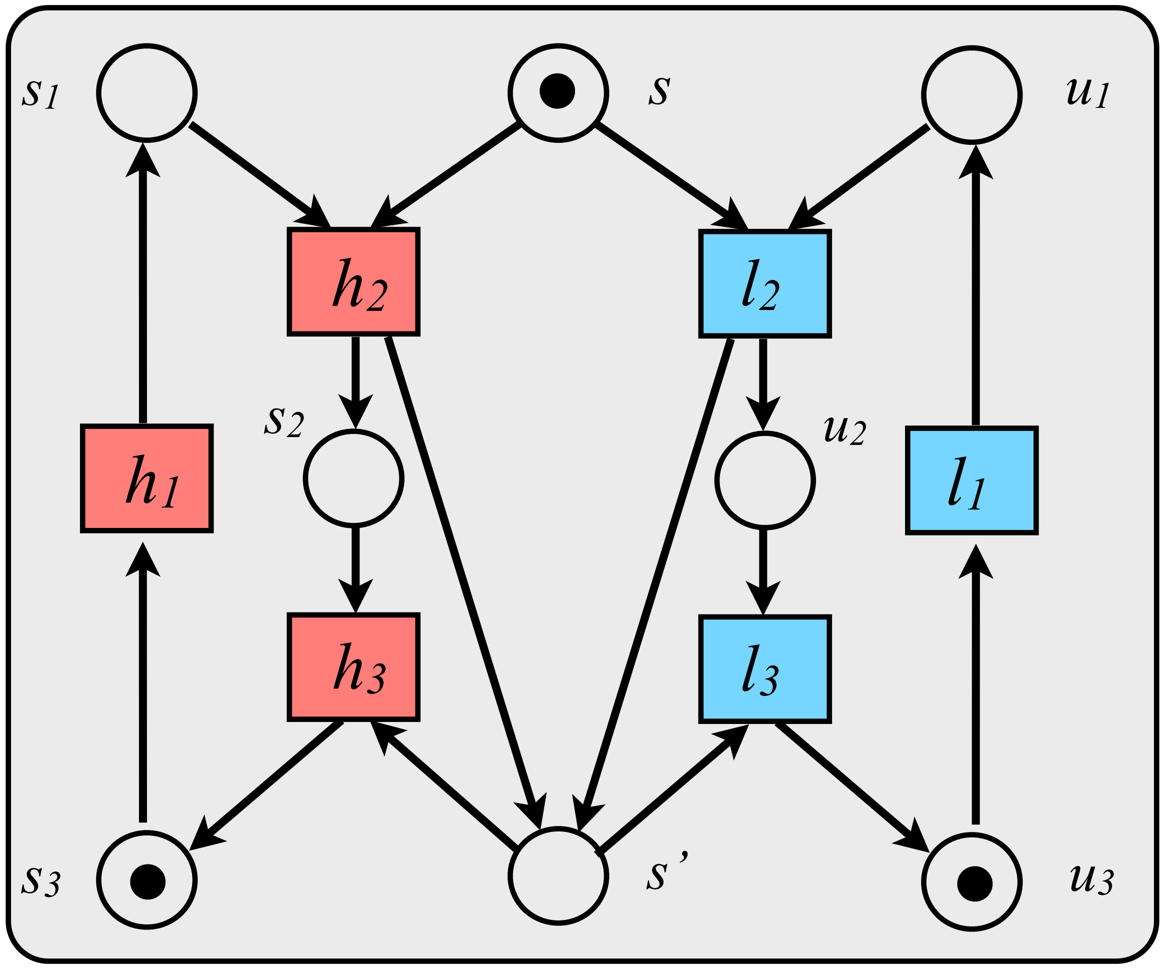
Causal characterisation?

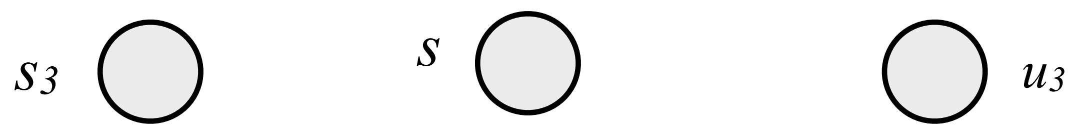
- Still an interleaving characterisation, based on the marking graph
- Not what we aimed at
- Use a true concurrent semantics of PNs!

Unfolding semantics

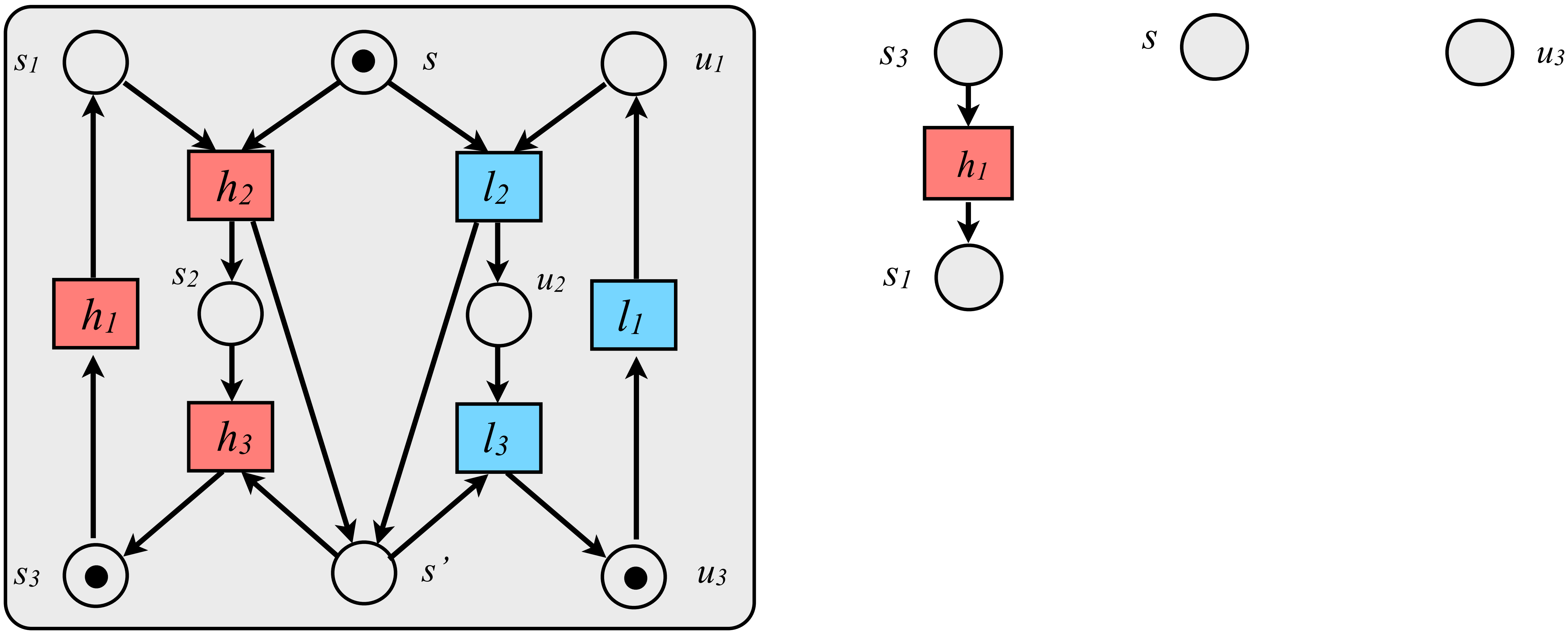
- Unfold a net N generating a nondeterministic, branching structure $U(N)$ including possible event and token occurrences
  - causality
  - conflict
  - concurrency

Unfolding semantics

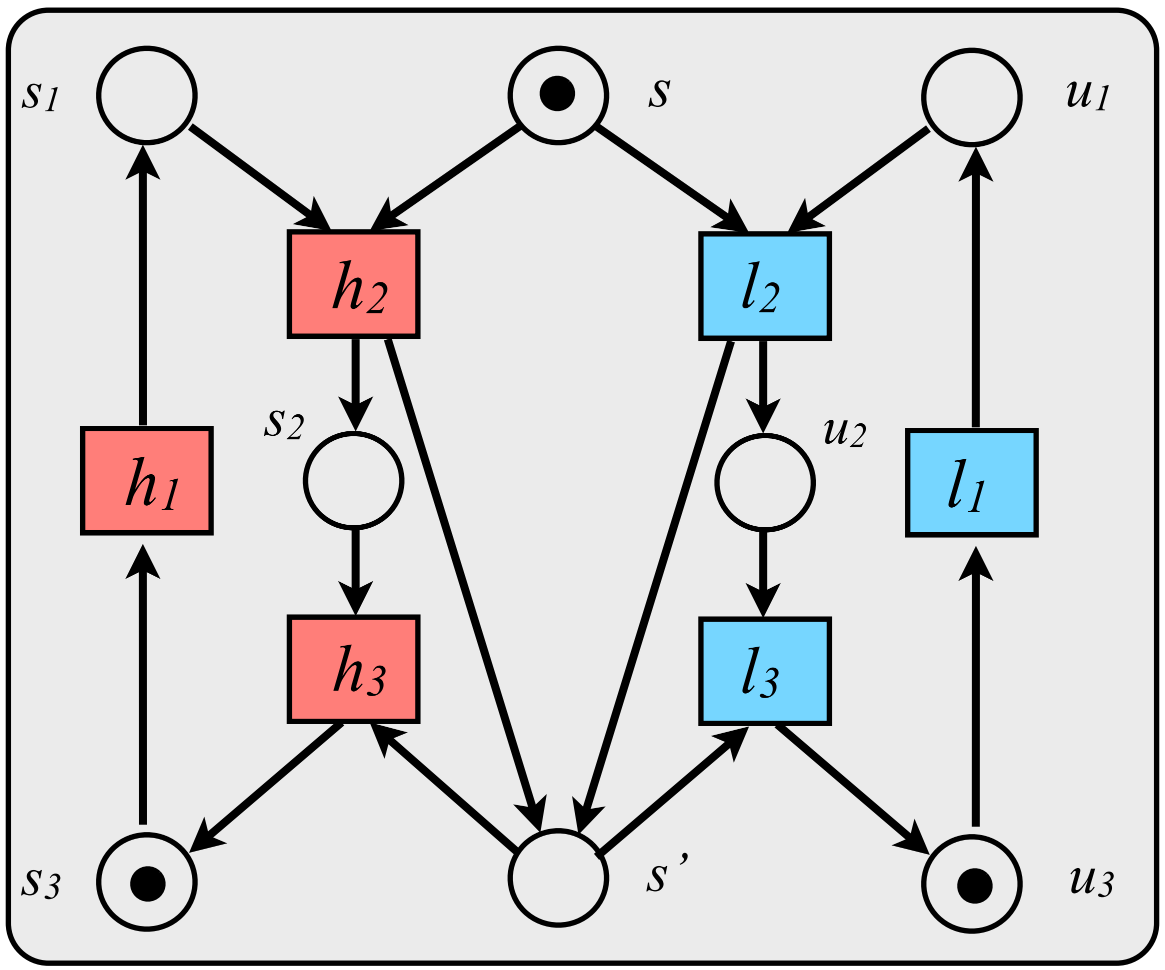


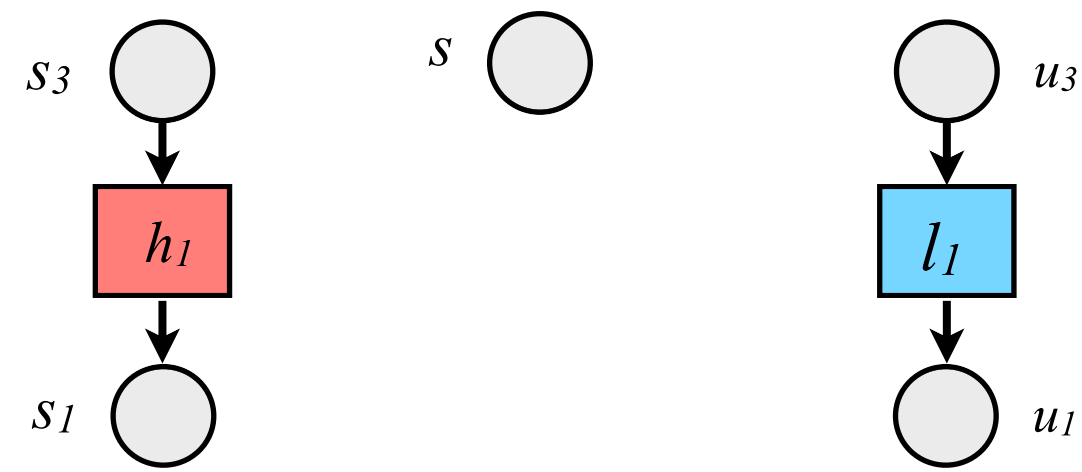


Unfolding semantics

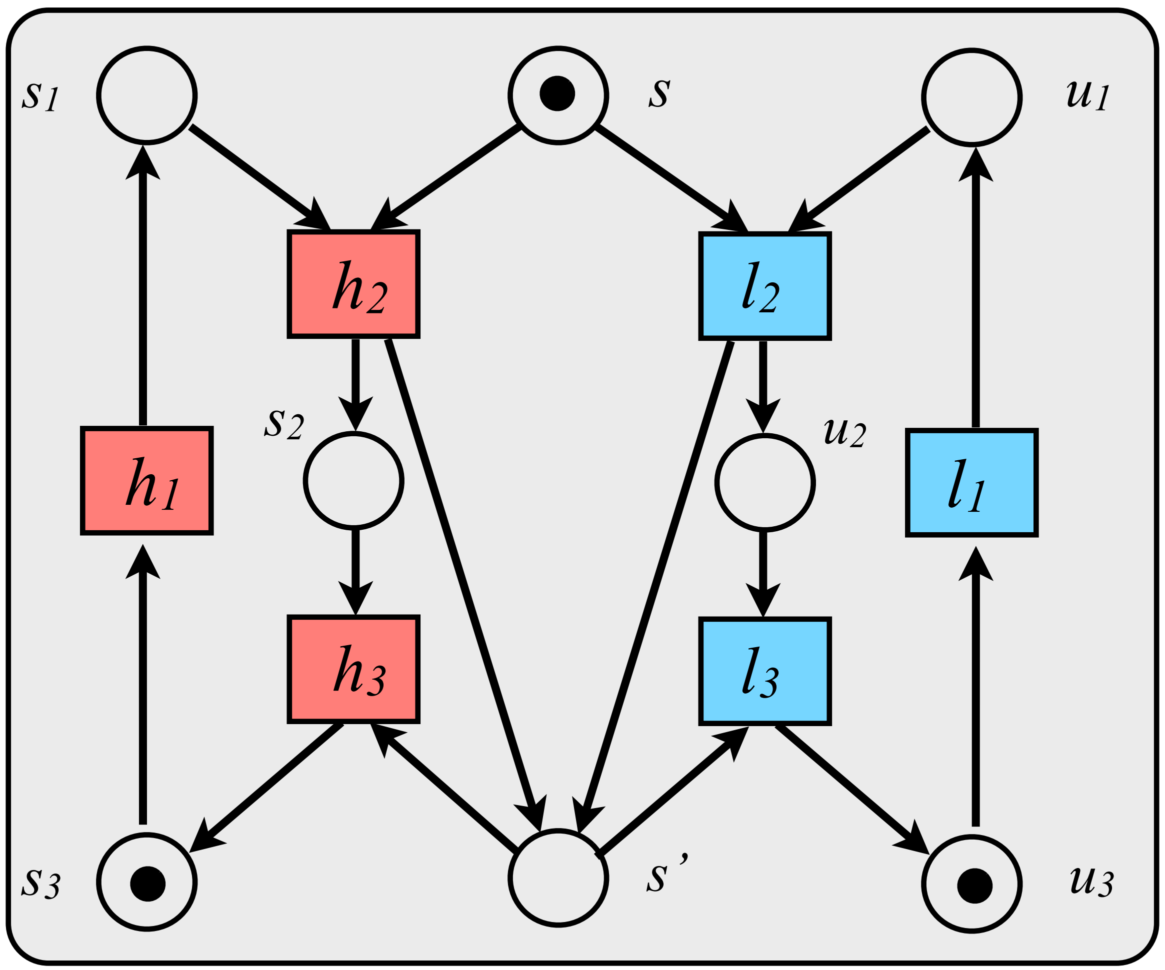


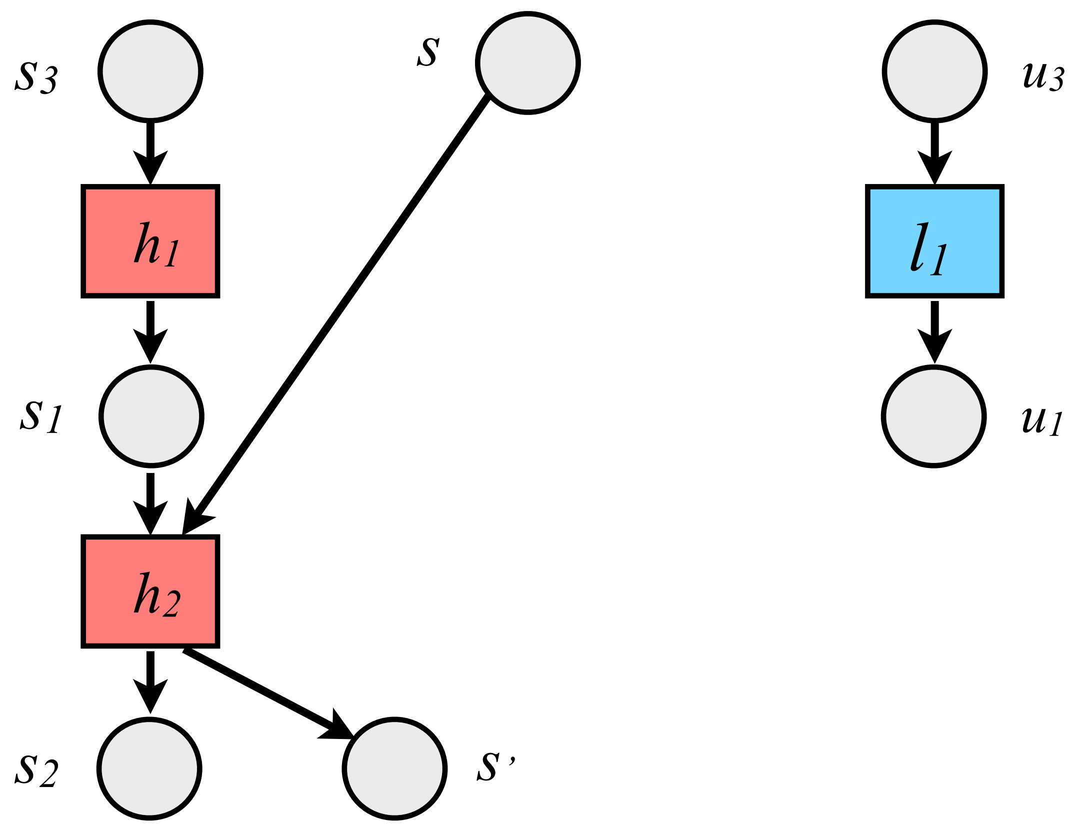
Unfolding semantics



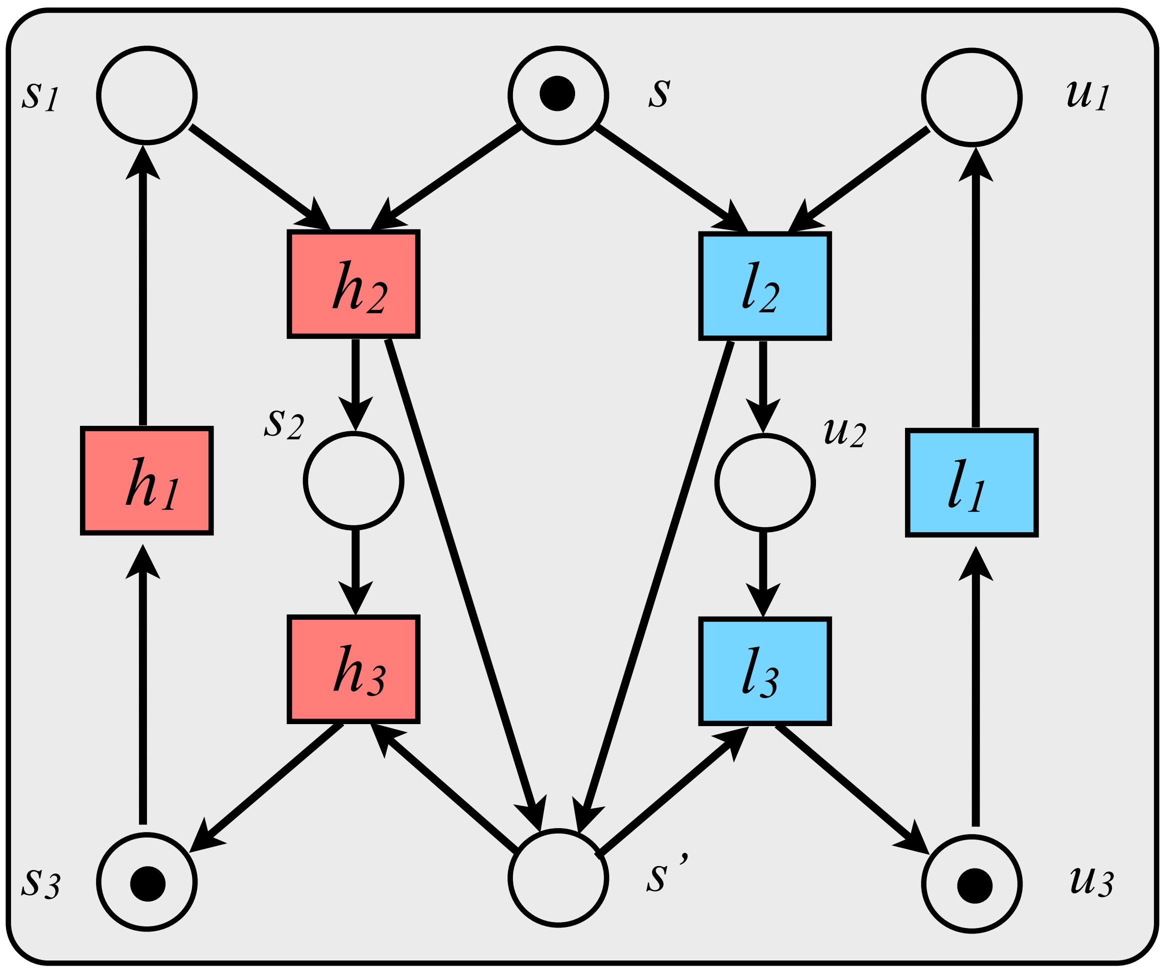


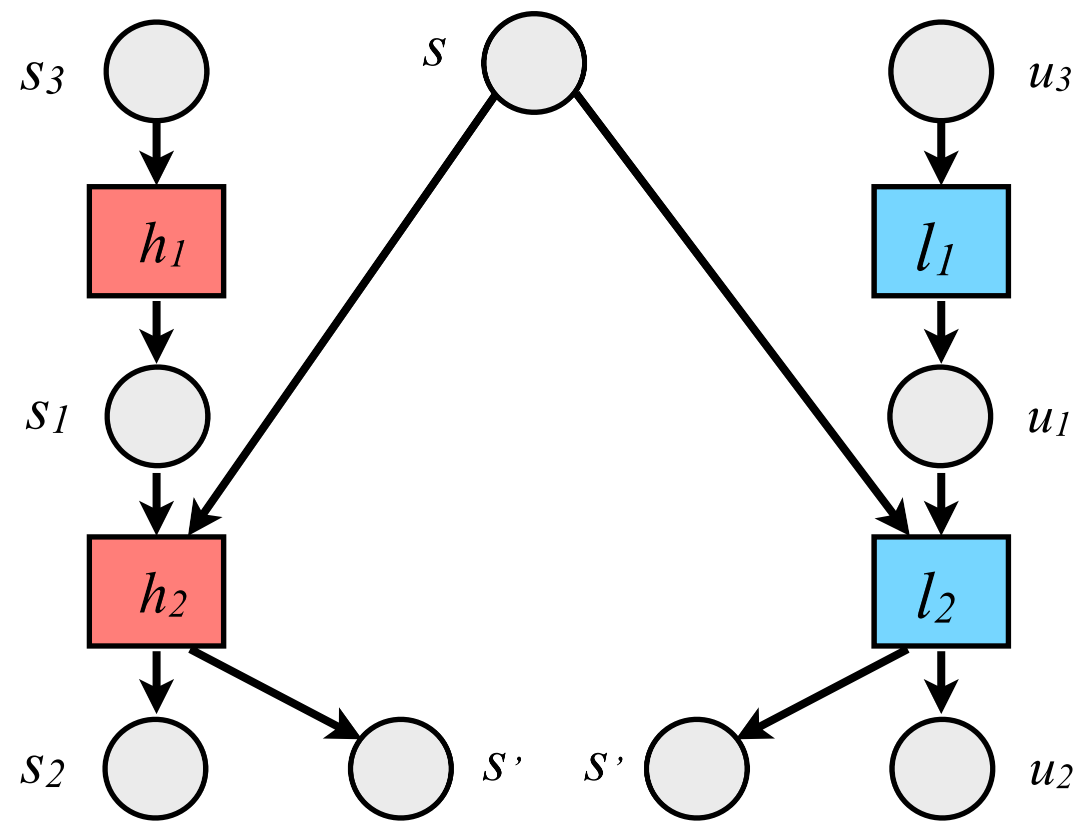
Unfolding semantics



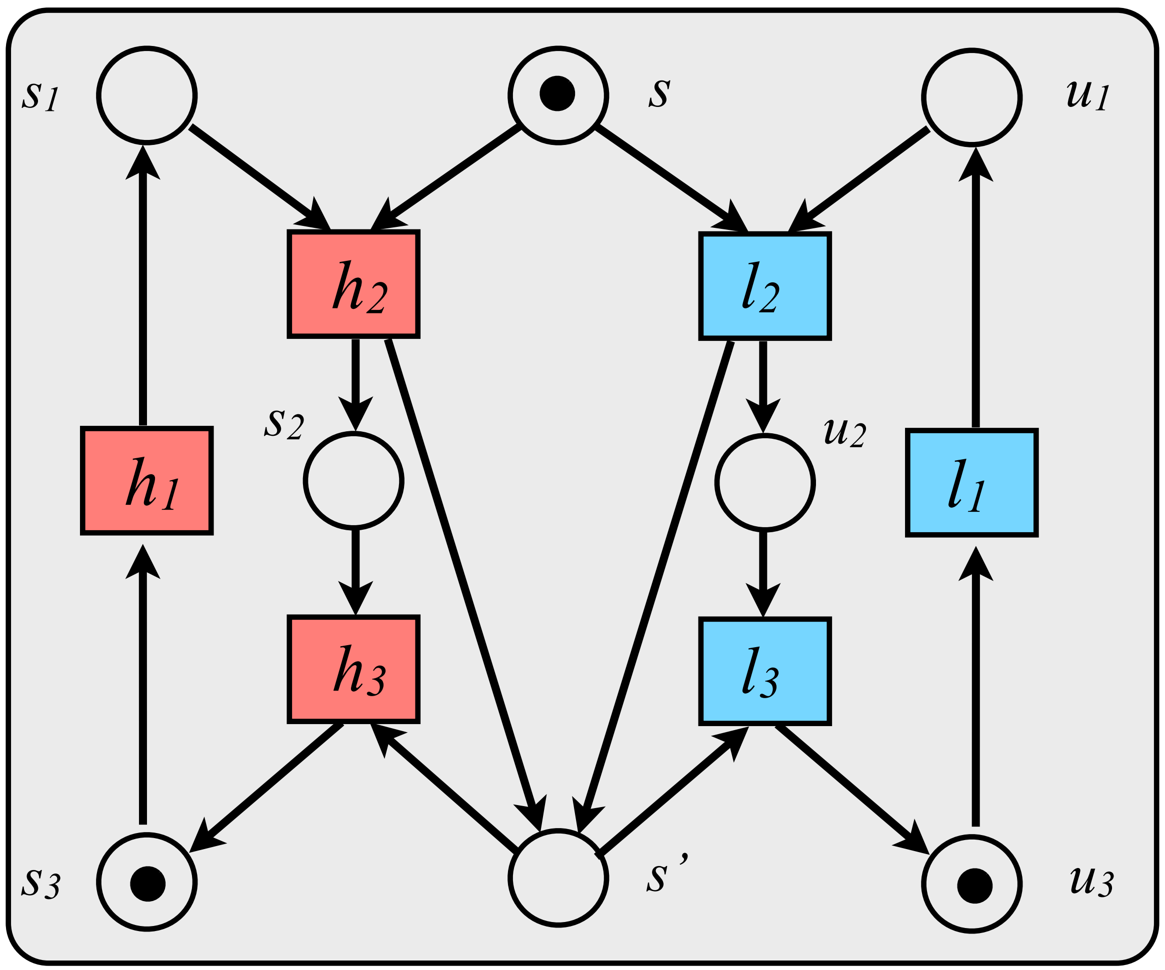


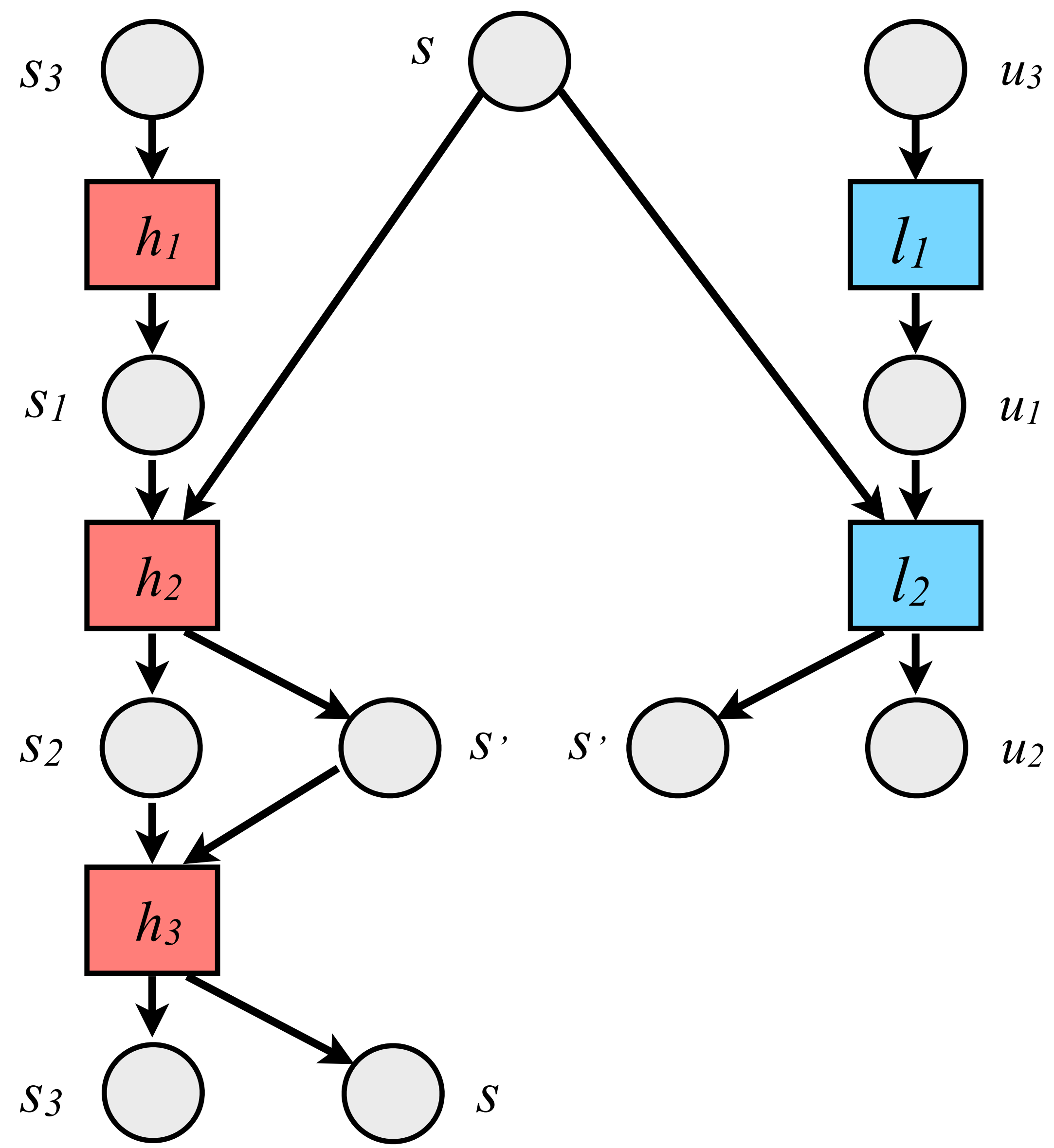
Unfolding semantics



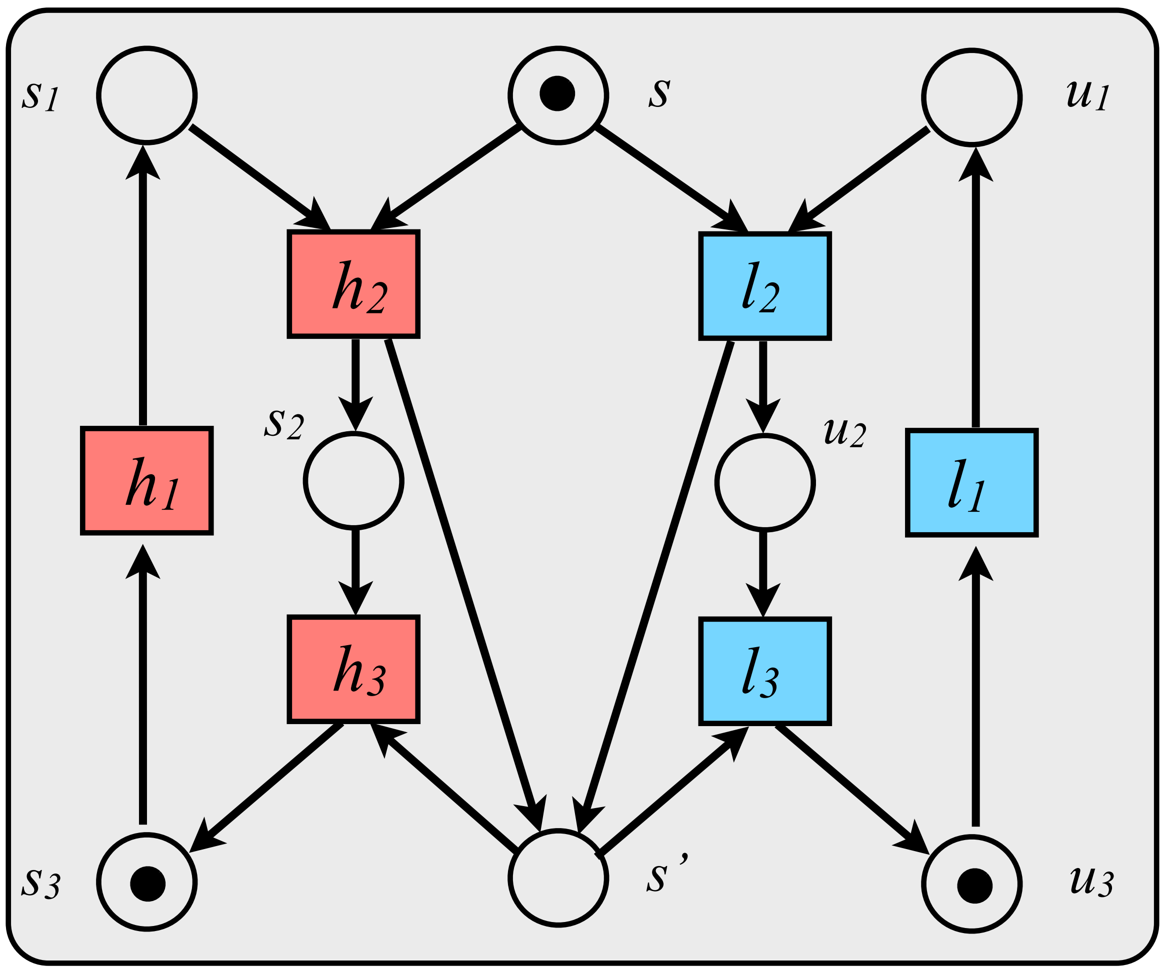


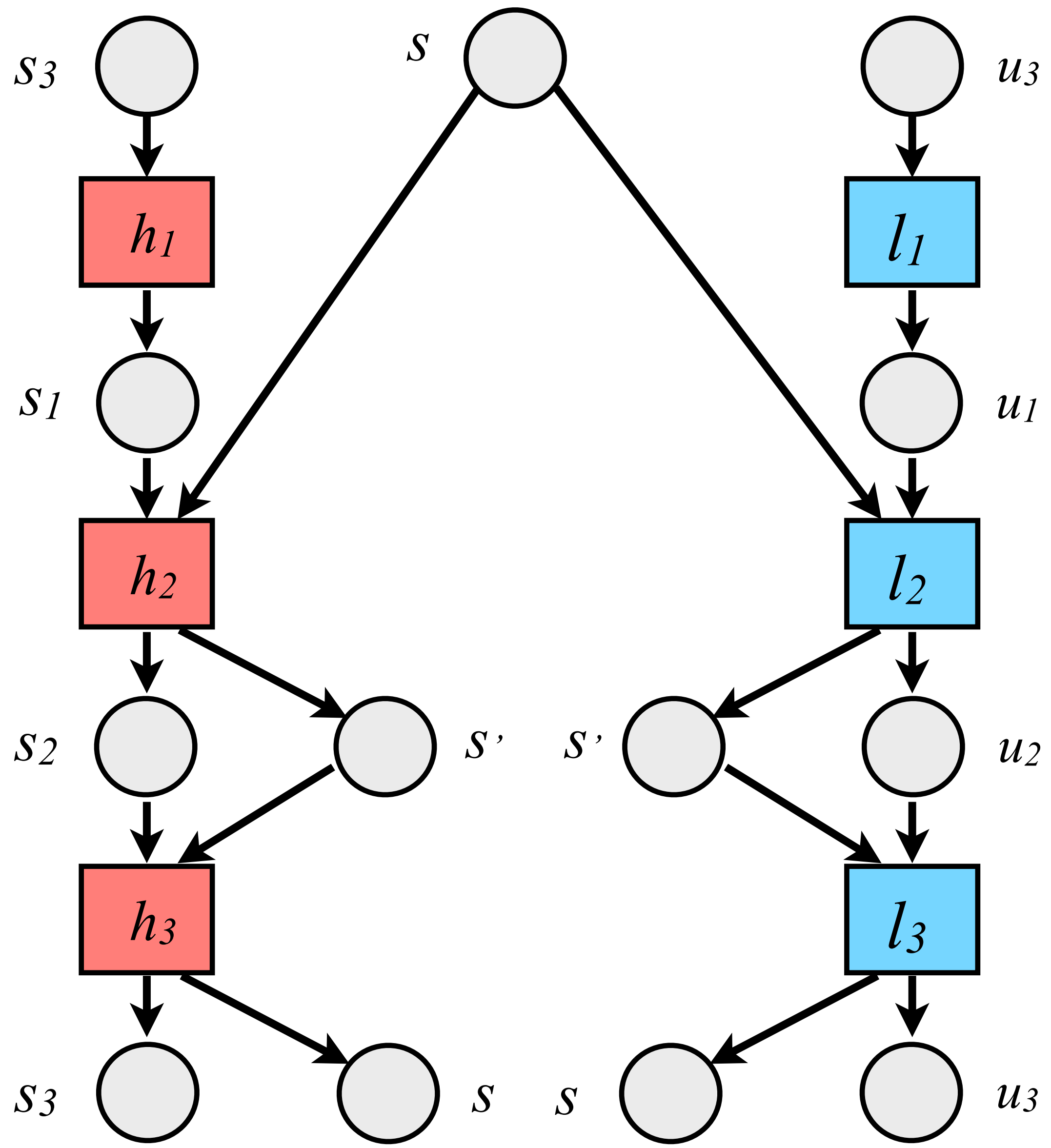
Unfolding semantics



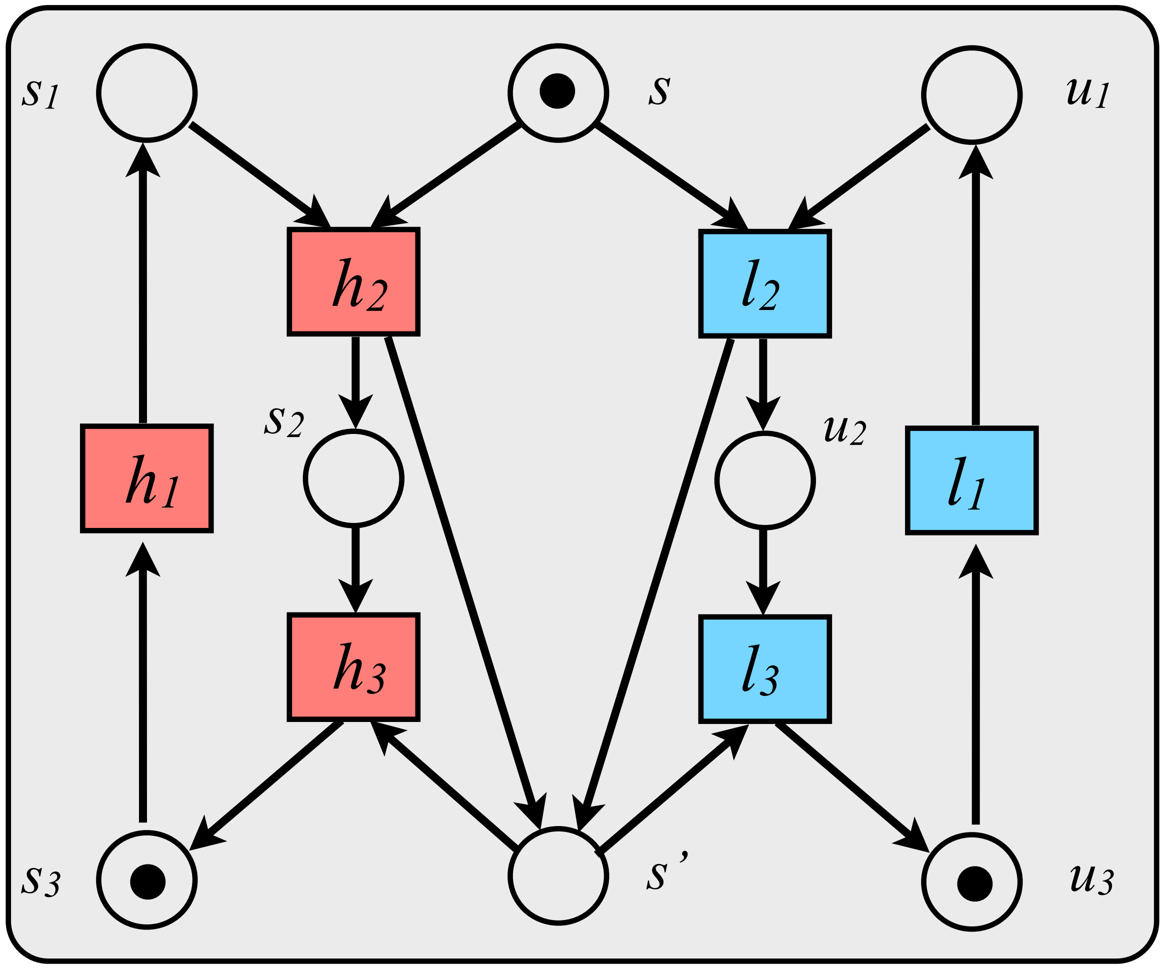


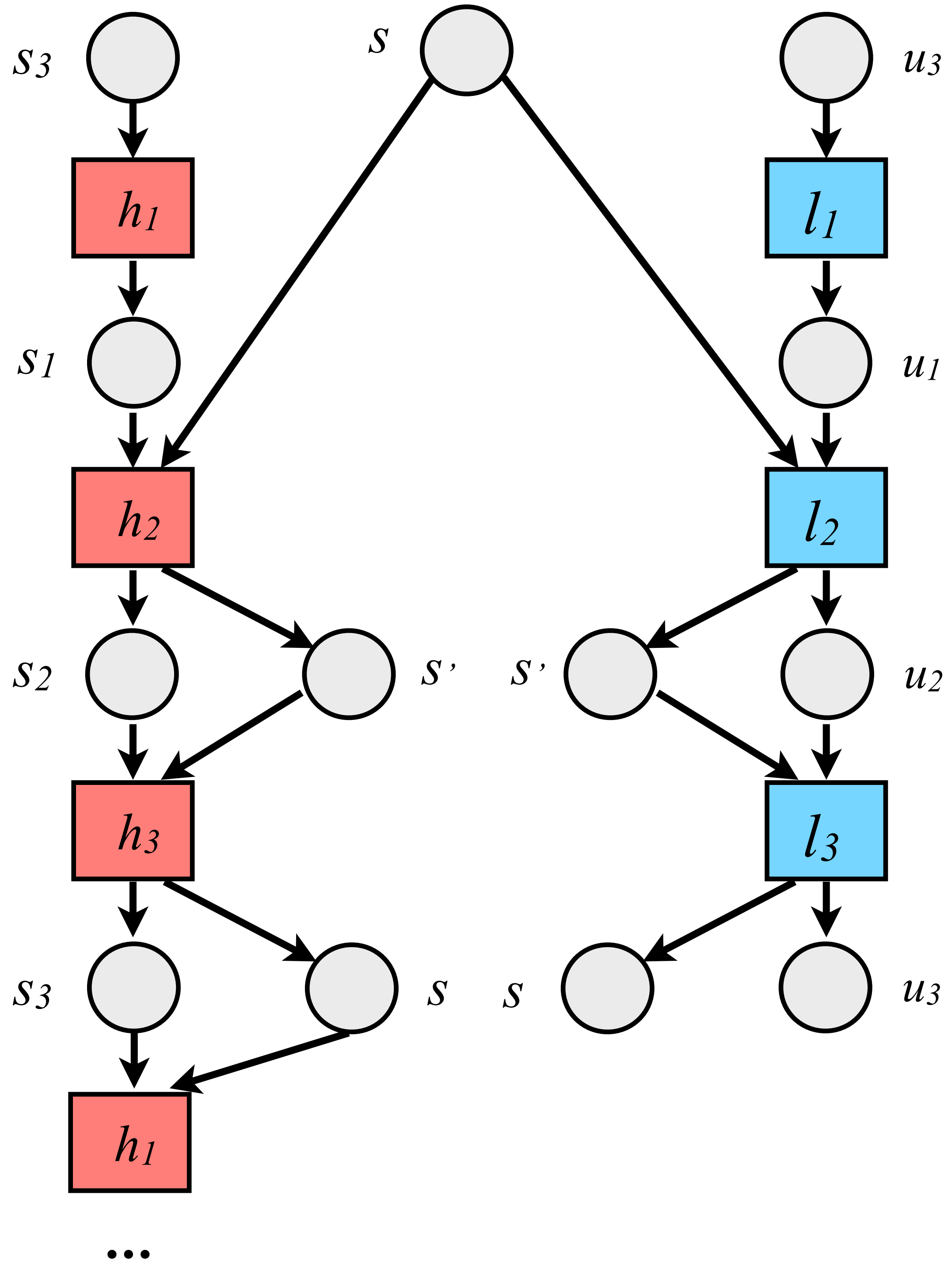
Unfolding semantics



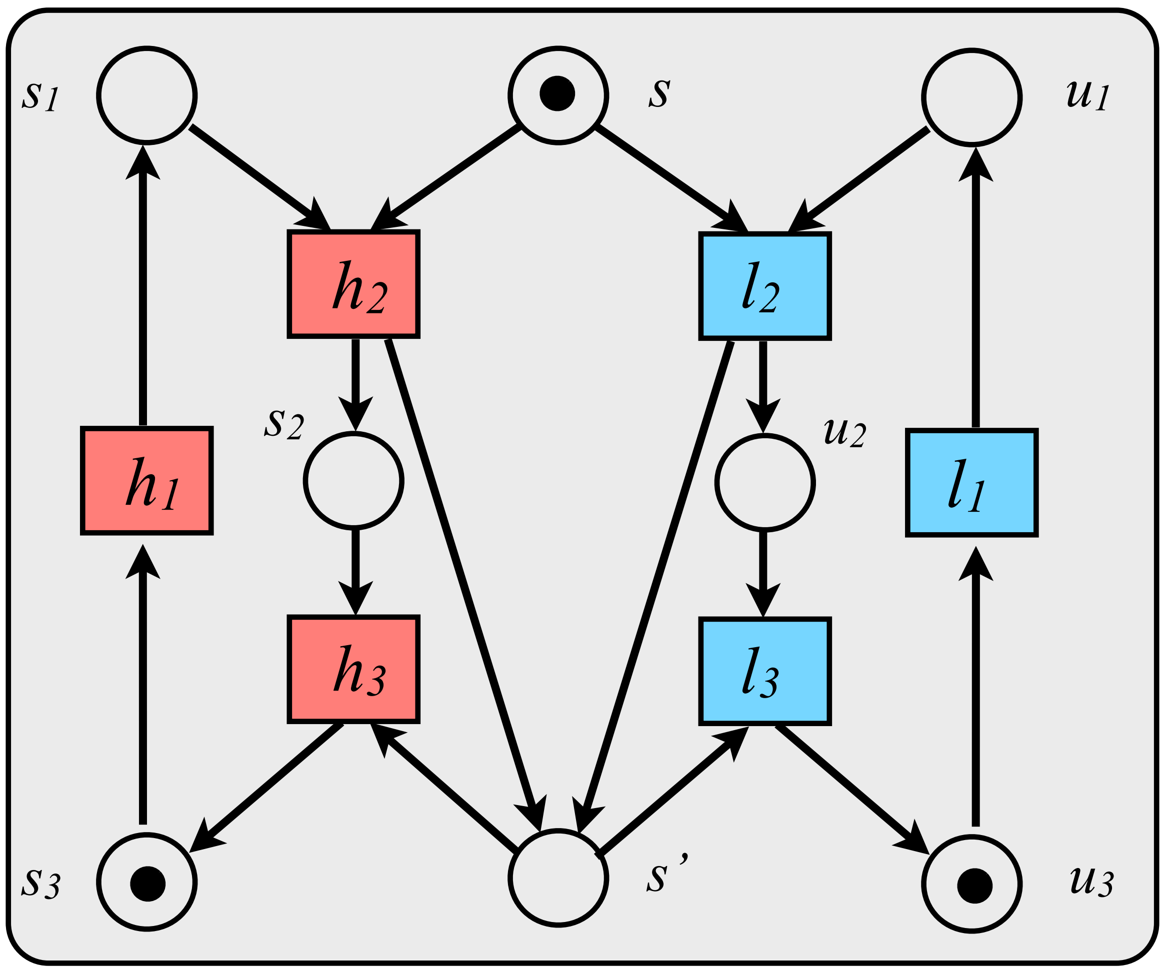


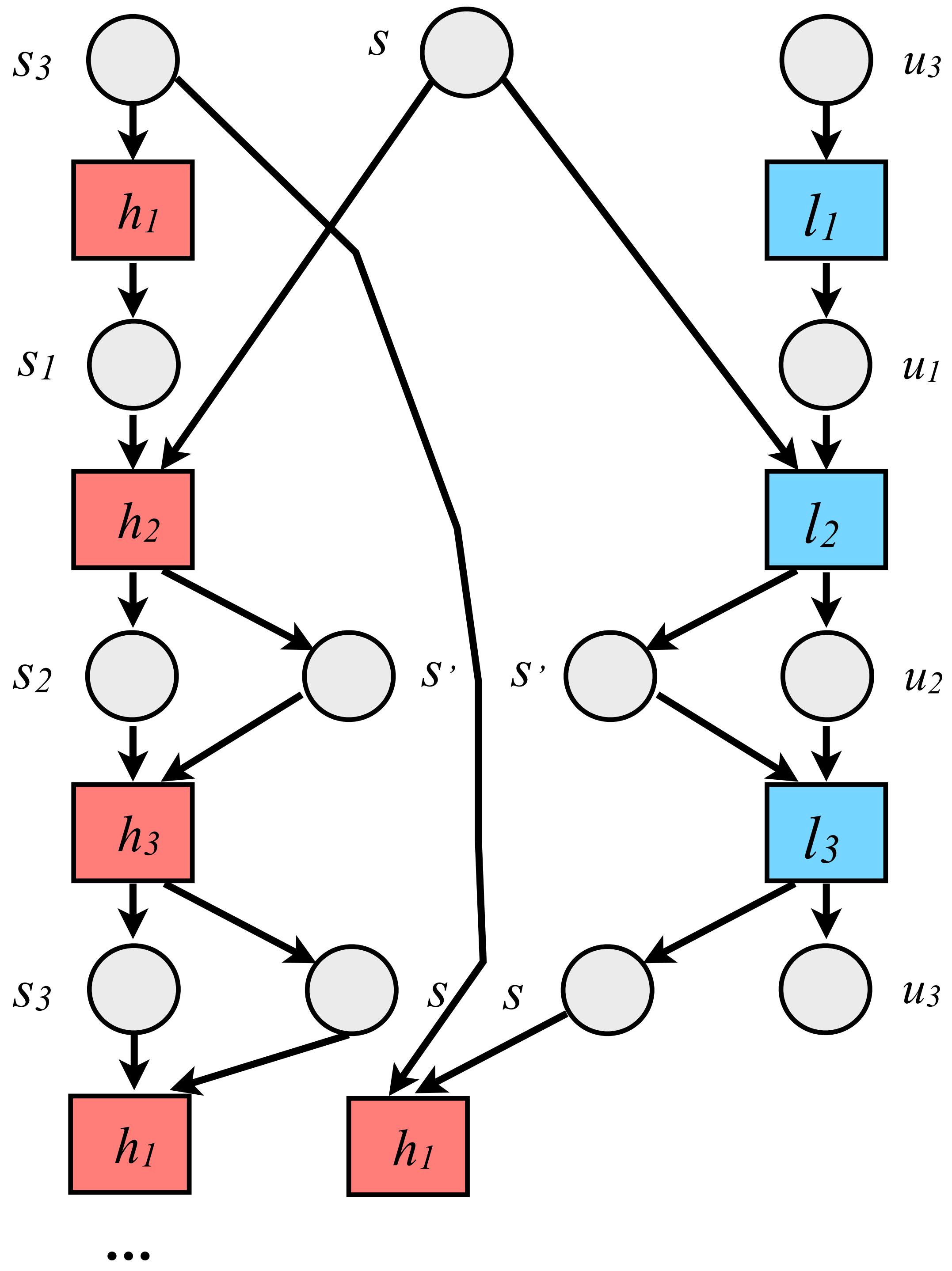
Unfolding semantics





Unfolding semantics





Safe nets

- Each place contains at most a token

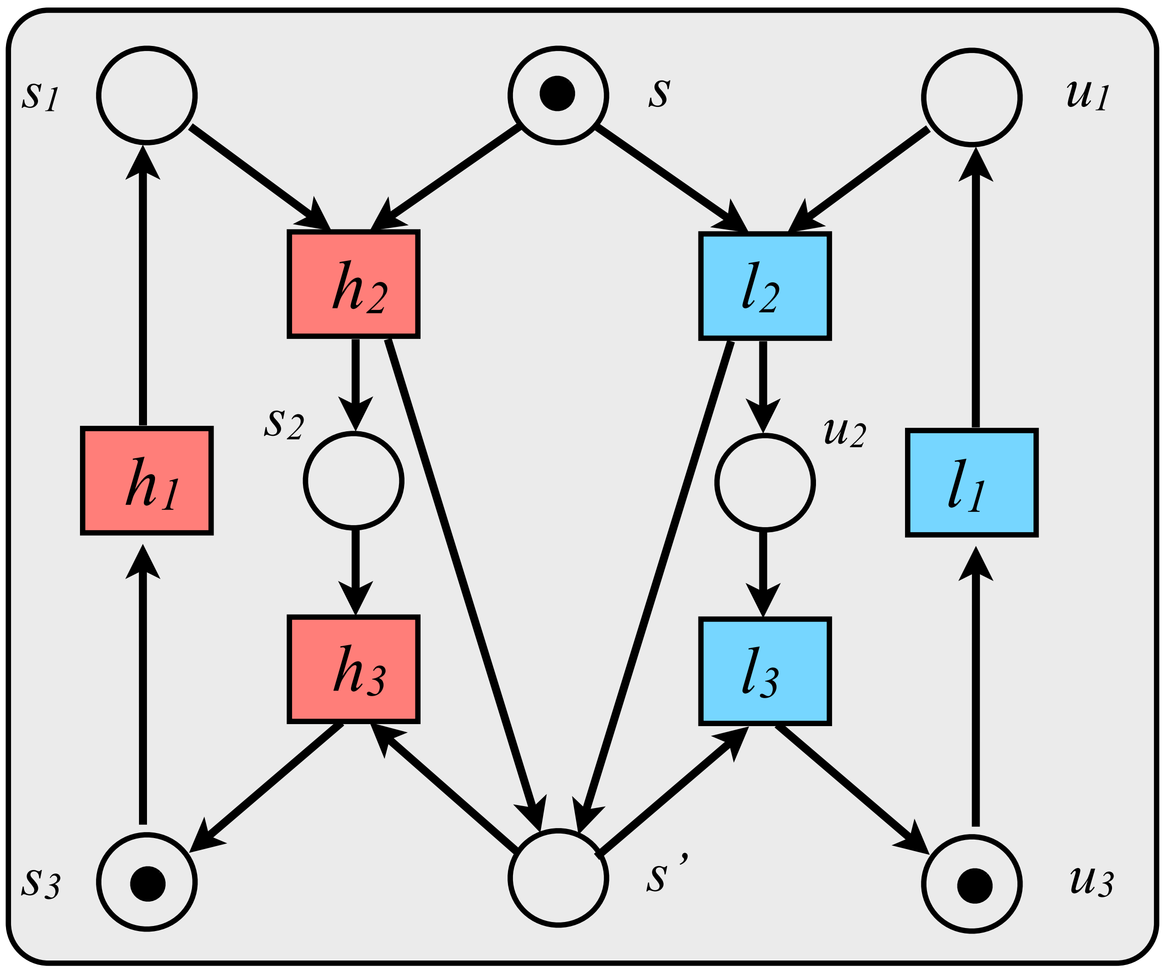
Safe nets

- Each place contains at most a token
- A place s in N is active causal/conflict **iff** there is an occurrence of s in $U(N)$ which is potentially causal/conflict

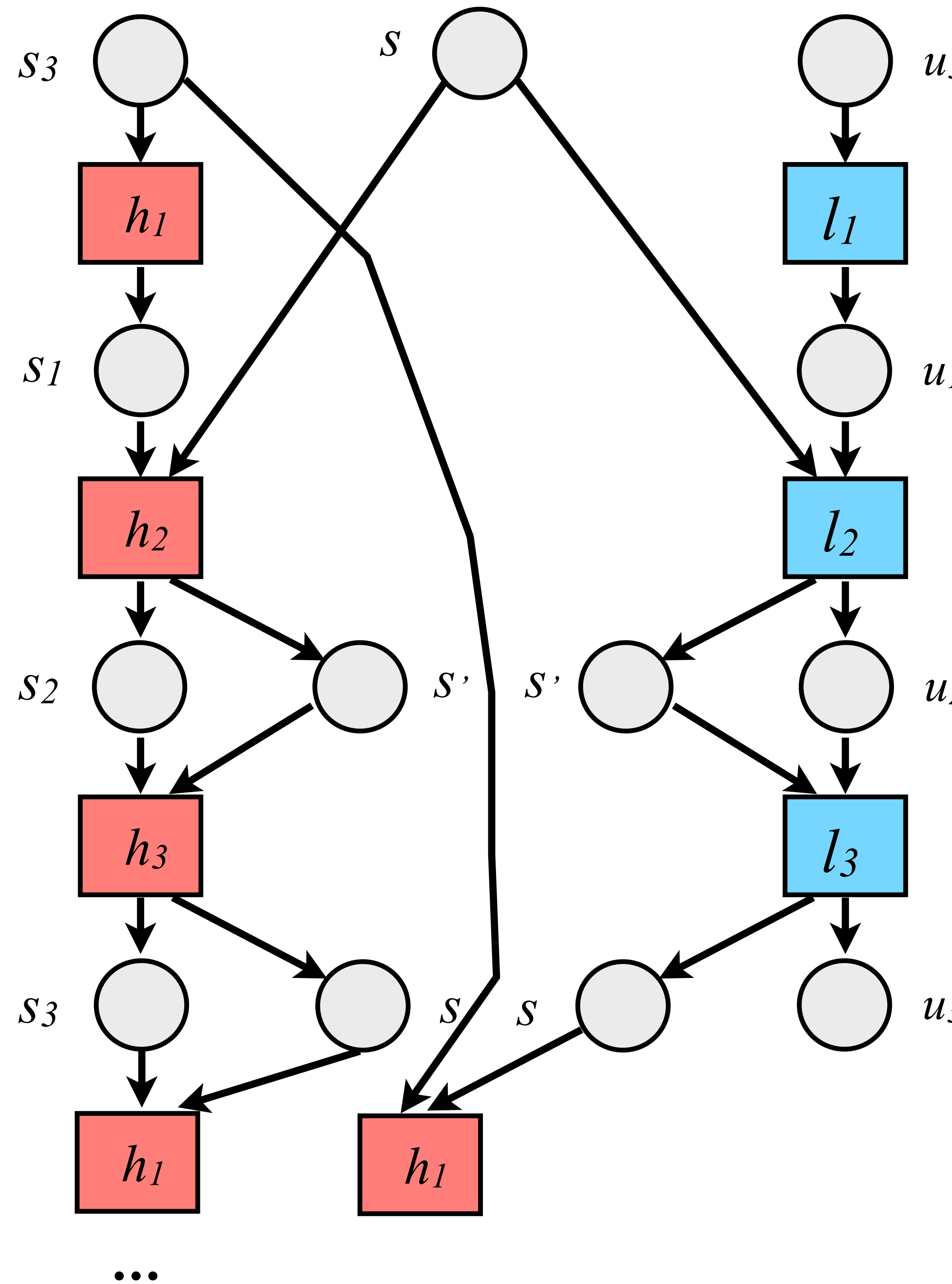
Safe nets

- **Theorem**: net N is BNDC iff there are no h, l in $U(N)$ such that
  - h is a **direct cause** of l
  - h is in **direct conflict** with l

Example

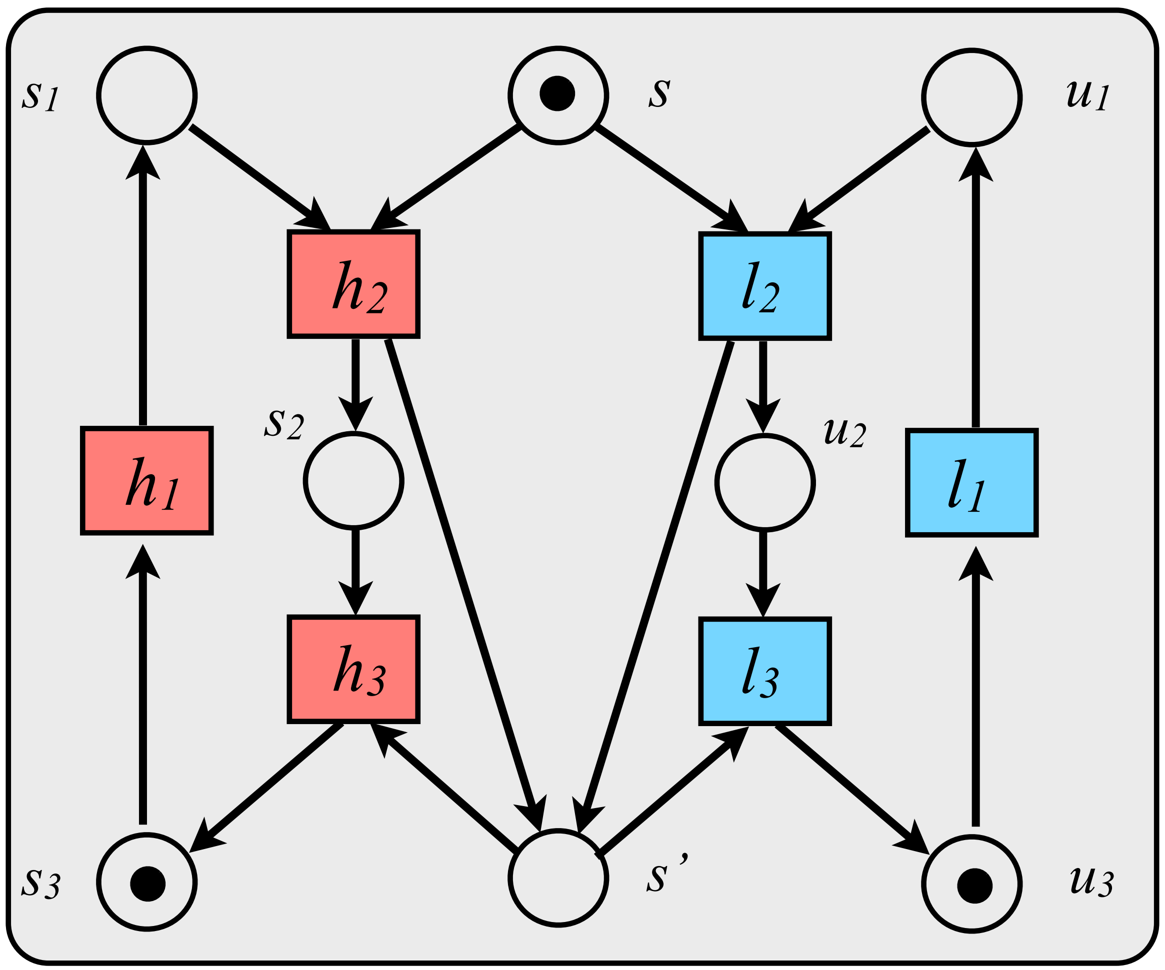


s_1
s
u_1
h_2
l_2
s_2
u_2
h_1
l_1
h_3
l_3
s_3
s'
u_3

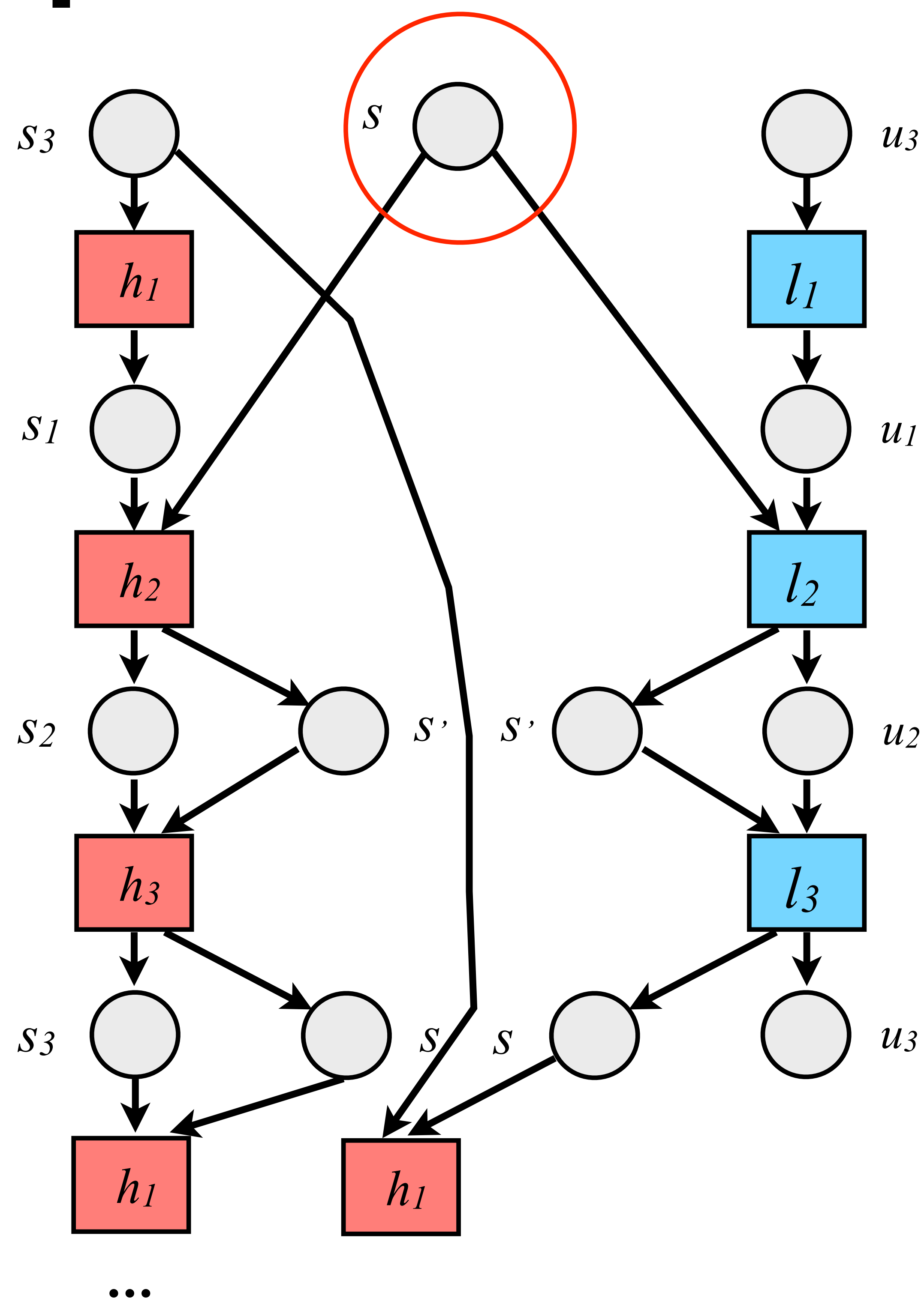


s_3
s
u_3
h_1
l_1
s_1
u_1
h_2
l_2
s_2
s'
s'
u_2
h_3
l_3
s_3
s
s
u_3
h_1
h_1
...

Example

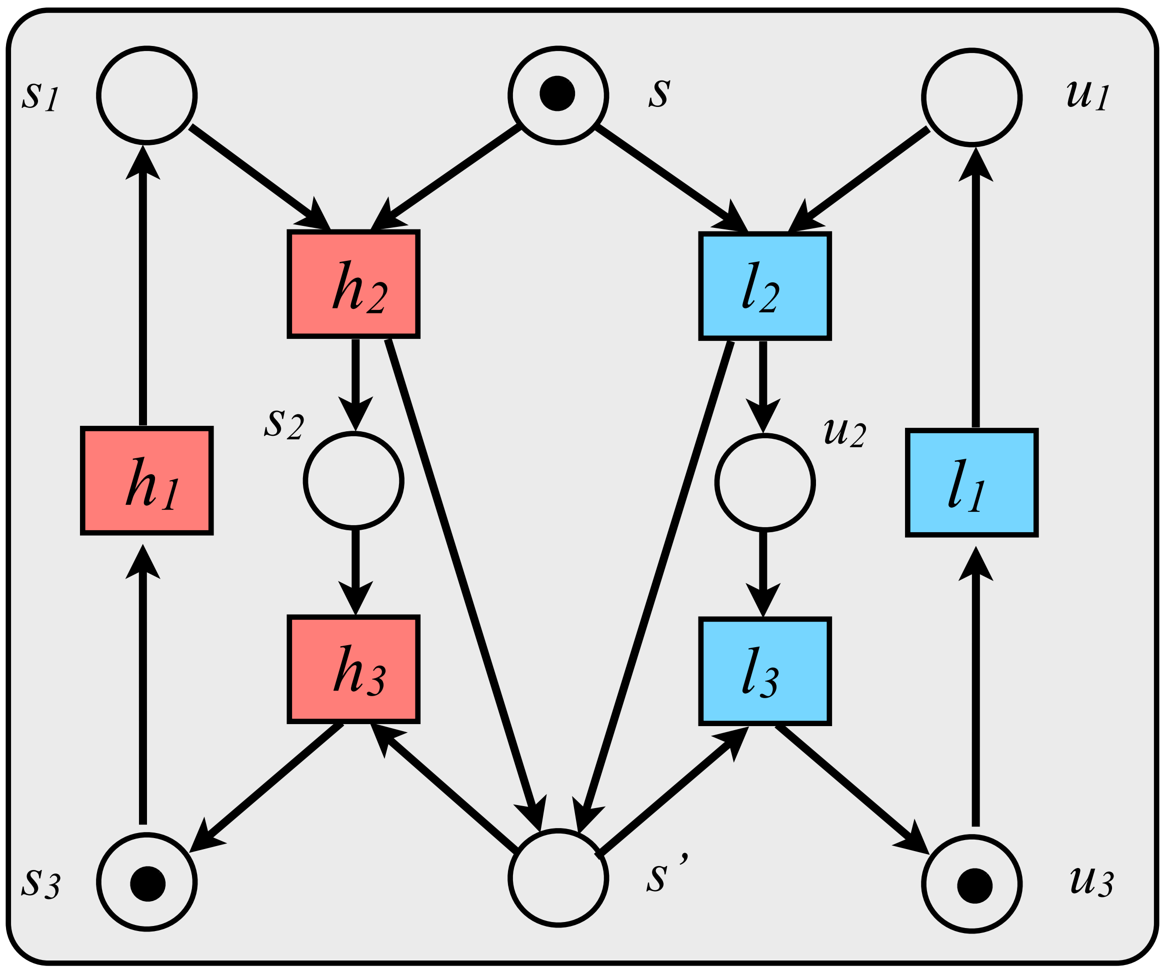


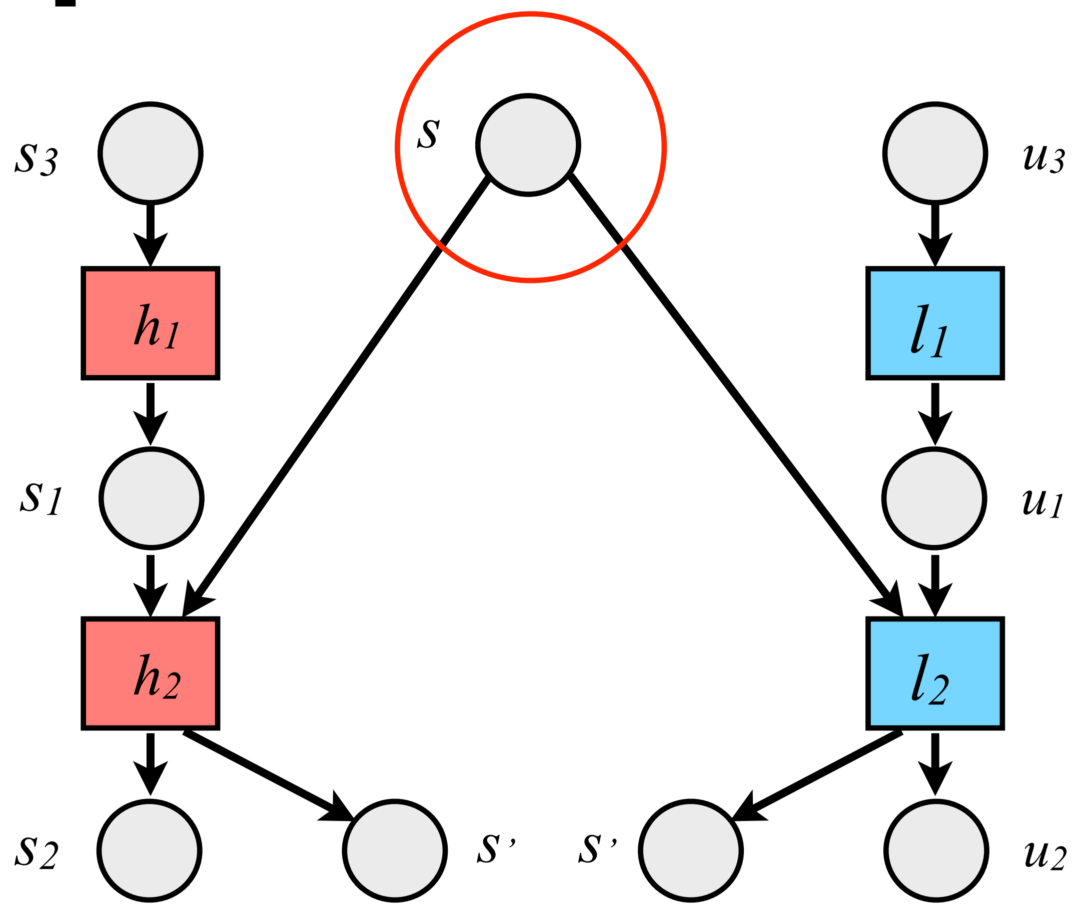
s1
s
u1
h2
l2
s2
u2
h1
l1
h3
l3
s3
s'
u3



s3
s
u3
h1
l1
s1
u1
h2
l2
s2
s'
s'
u2
h3
l3
s3
s
s
u3
h1
h1
...

Example





Algorithm on the unfolding

- The unfolding is **infinite!**

Algorithm on the unfolding

- The unfolding is **infinite!**
- For finite state PNs one can generate a **finite prefix**, complete (for reachability)

Algorithm on the unfolding

- The unfolding is **infinite!**
- For finite state PNs one can generate a **finite prefix**, complete (for reachability)
- **Idea** [McMillan]
  - **Cut-off**: event that generates the same marking as an event with smaller history
  - **Algorithm**: unfold stopping at cut-offs

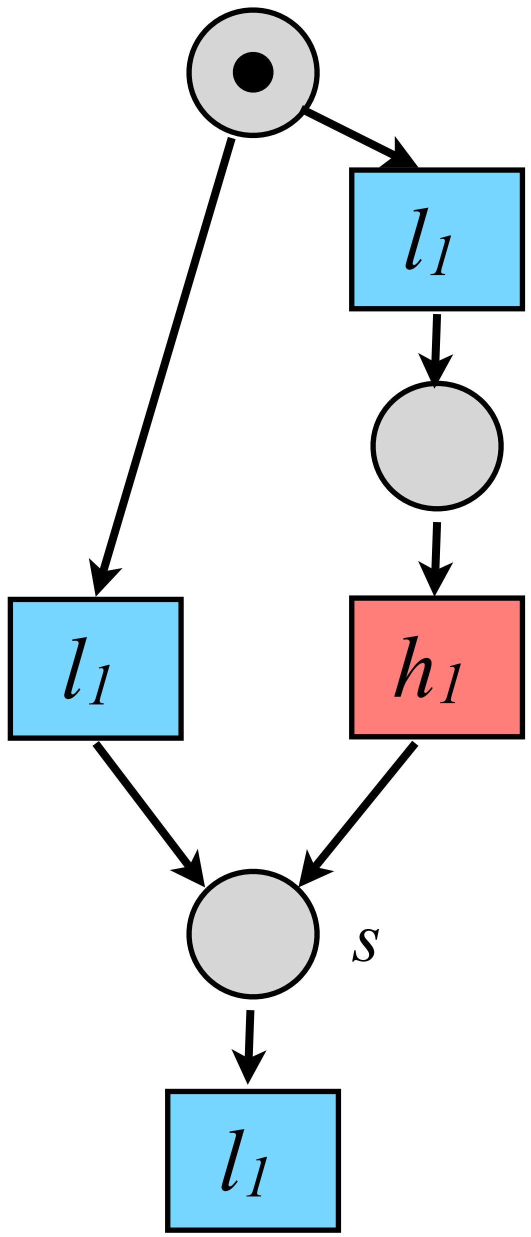
Algorithm on the unfolding

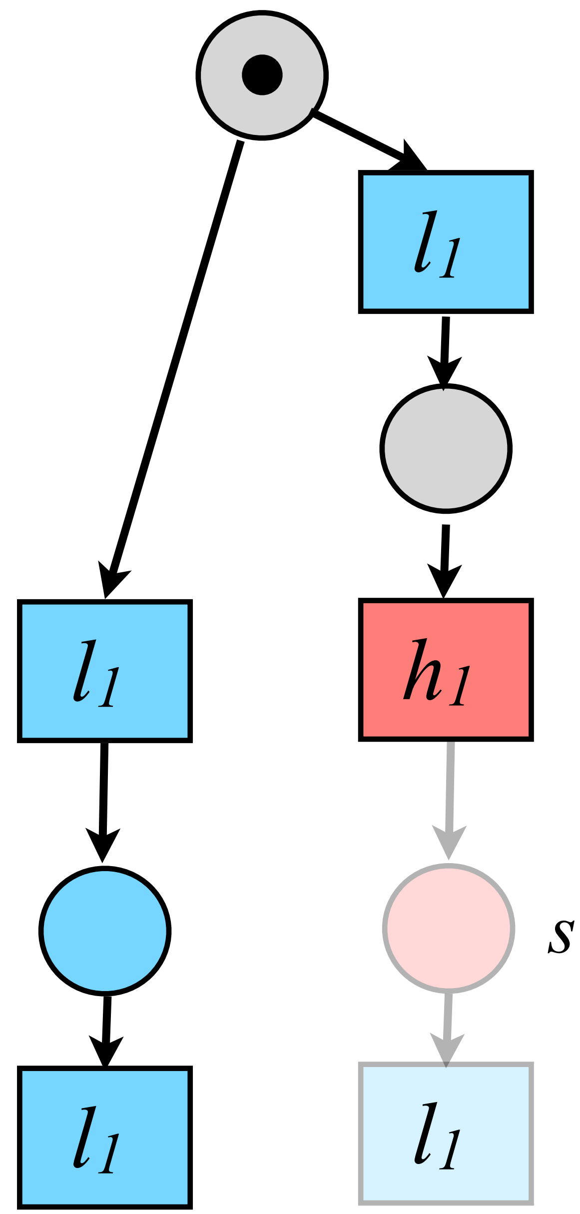
- Finite prefix construction can be adapted to guarantee **completeness for interferences**

Algorithm on the unfolding

- Finite prefix construction can be adapted to guarantee **completeness for interferences**
- **Idea**: record in the token the level (high/ low) of the generating transition

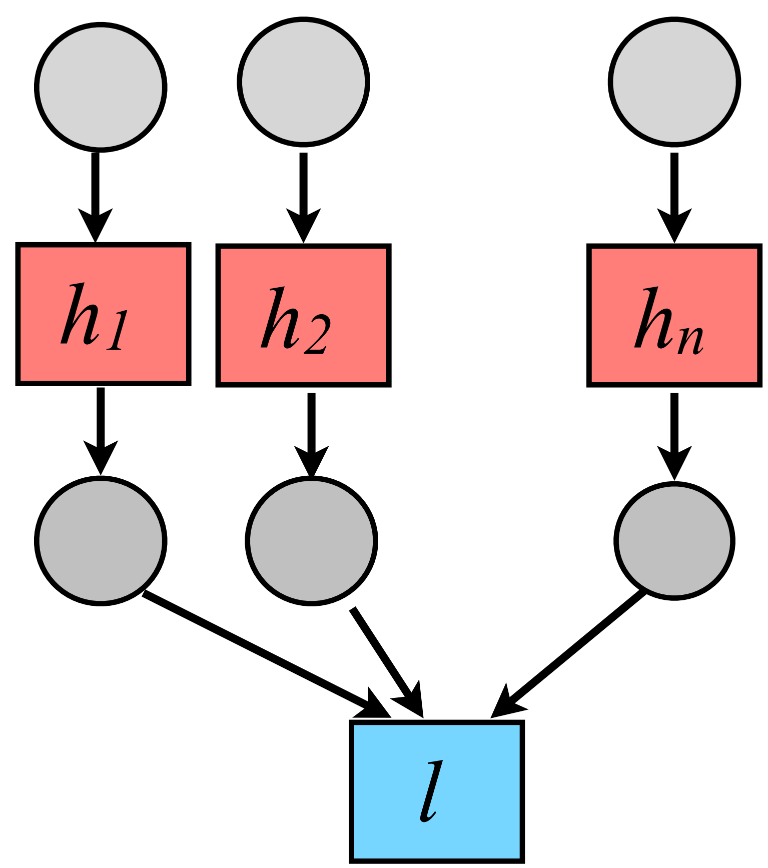
Algorithm on the unfolding





Algorithm on the unfolding

- The finite prefix can be **exponentially smaller** than the marking graph



General P/T nets

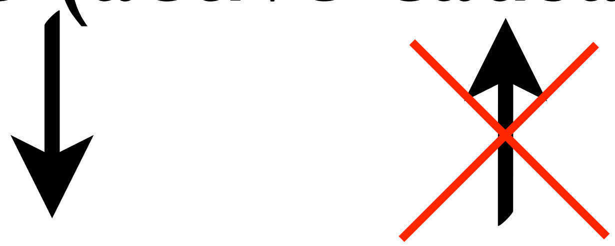
interference (active causal/conflict place)

$\Downarrow \qquad \cancel{\Uparrow}$

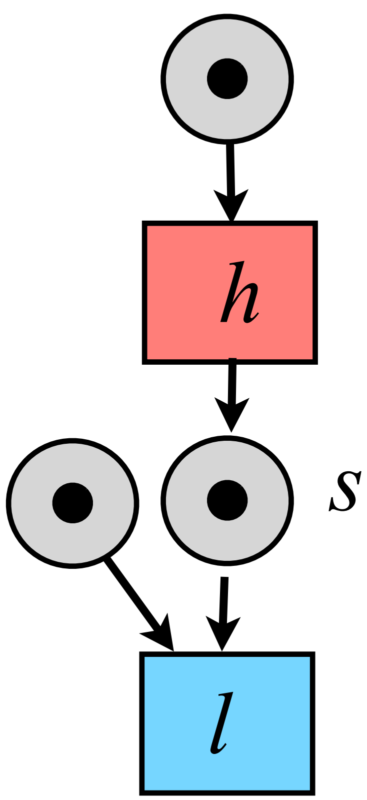
direct causality/conflict $h < l$ or $h \# l$

General P/T nets

interference (active causal/conflict place)



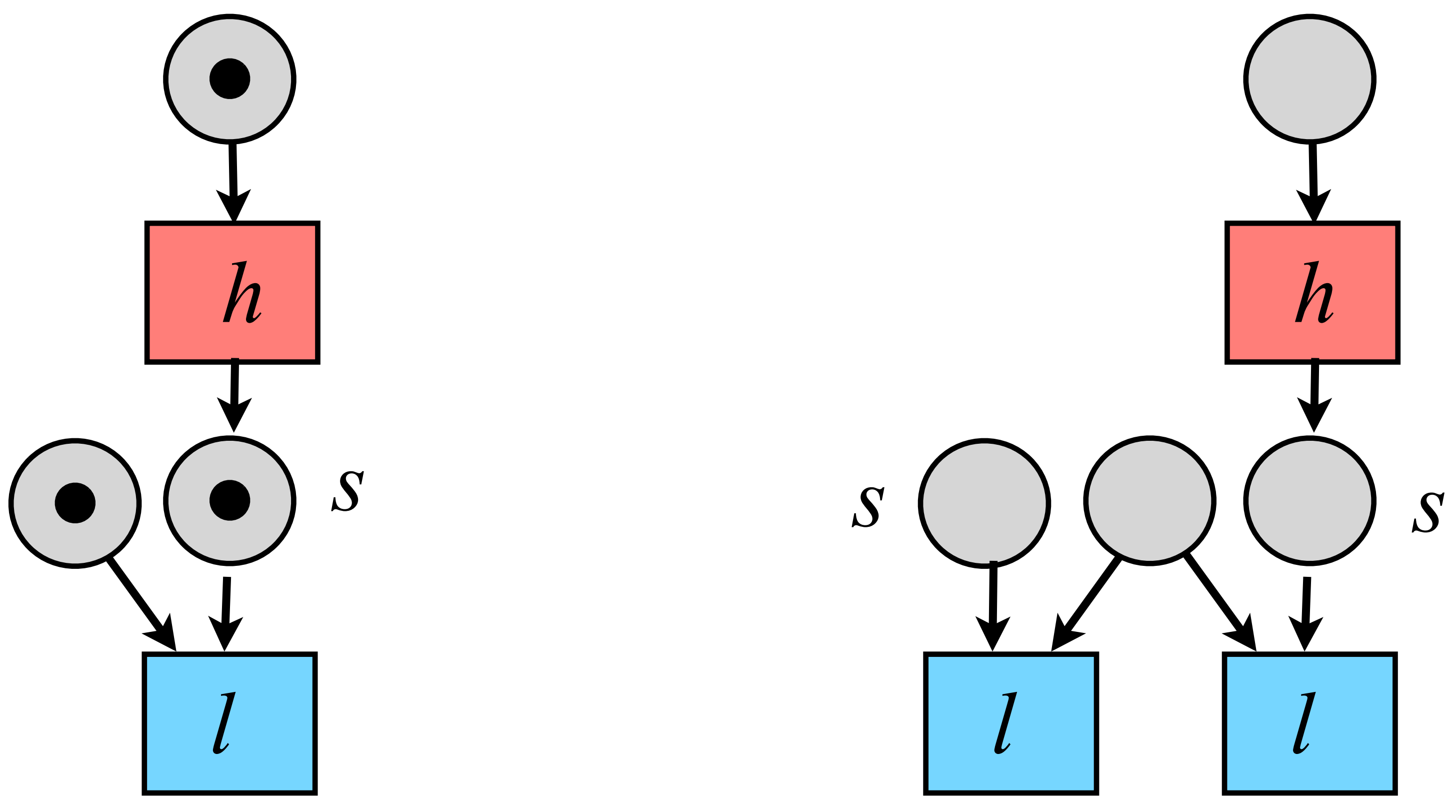
direct causality/conflict $h < l$ or $h \# l$



General P/T nets

interference (active causal/conflict place)

direct causality/conflict $h < l$ or $h \# l$



Conclusion

- Characterisation of non interference for Petri nets in terms of a concurrent semantics
- Efficient checking

Perspectives

- Non interference on
    - imperative languages (encoding control & data flow in PNs)
    - process calculi
- Study non interference properties arising in standard definitions when replacing interleaving with concurrent observational equivalences