

Annotating Coloured Petri Nets

Bo Lindstrøm and Lisa Wells

August 28, 2002
CPN'02

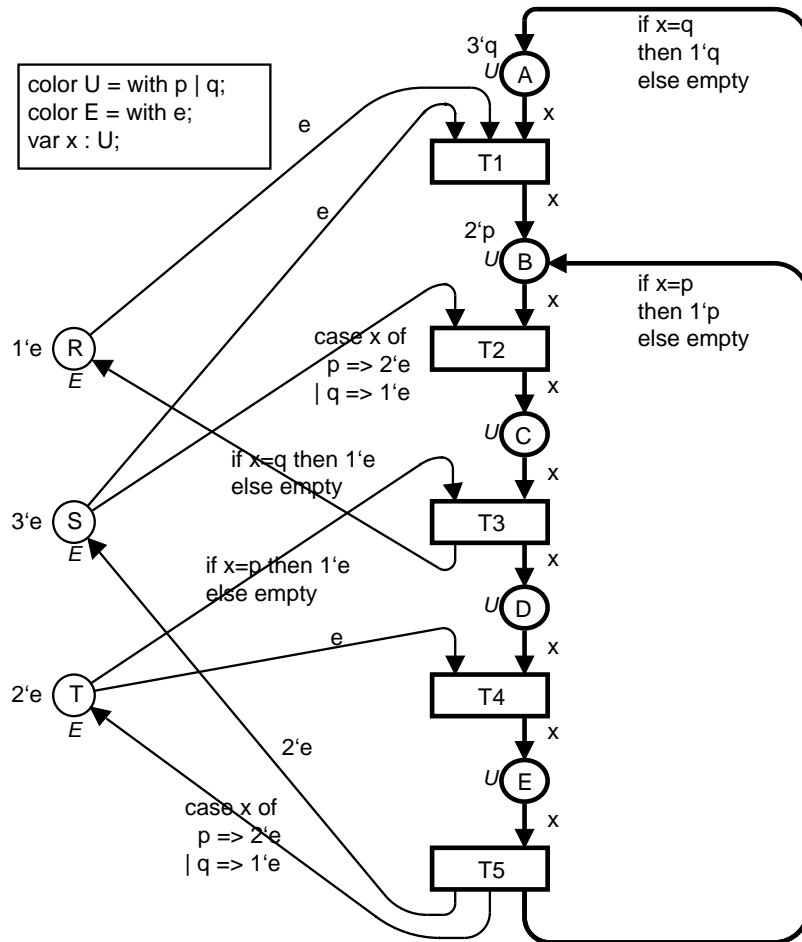
Plan:

- Motivation
- Annotating Coloured Petri Nets
- Practical Use of Annotations
- Future Work and Conclusions

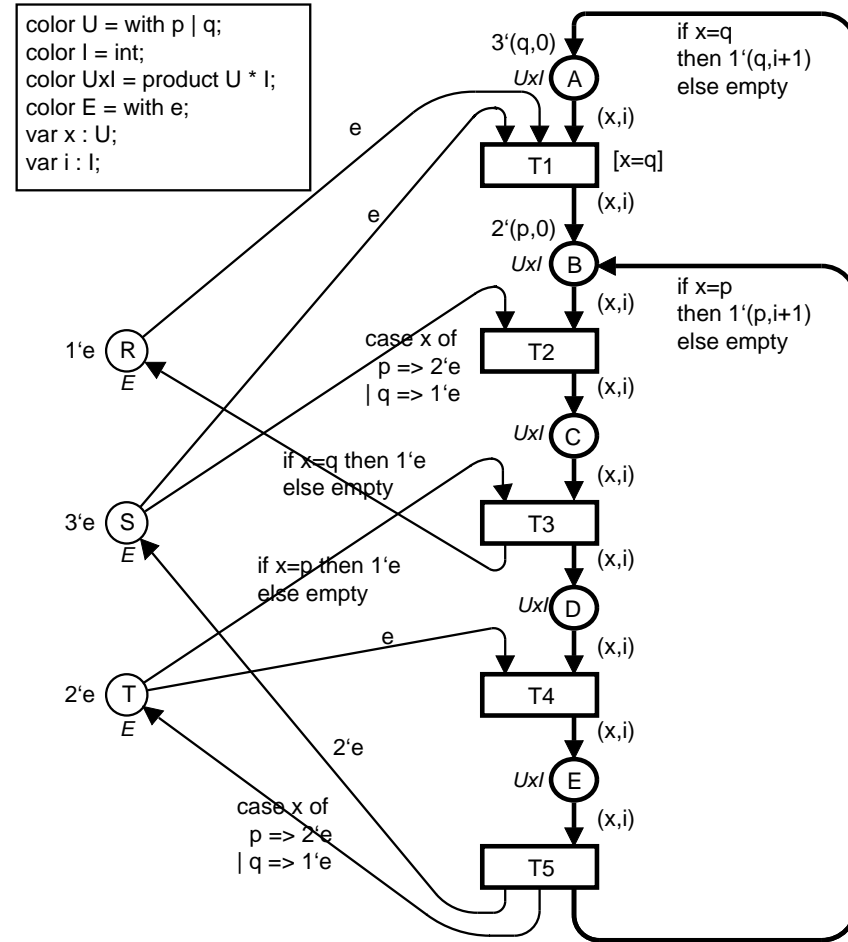
Motivation

- ❑ CP-nets can be used for several fundamentally different purposes, such as functional analysis, performance analysis, and visualisation.
- ❑ There are a number of libraries and tool extensions for Design/CPN that support these uses.
- ❑ It is seldom the case that the same CP-net can be used for a variety of different purposes.
- ❑ Frequently necessary to modify a CP-net to obtain a CP-net that is appropriate for a given purpose.

Example: Resource Allocation System



Basic CP-net



Extended CP-net

Problems

- ❑ Maintaining several versions of a CP-net is time consuming.
 - Extra information needs to be added or removed manually.
- ❑ Modifying a CP-net may introduce undesirable or unexpected behaviour.
- ❑ Until recently, it has only been partially possible to use a CP-net for different purposes without having to modify the net.

Plan

- Motivation
- **Annotating Coloured Petri Nets**
 - Annotations
 - Annotation Layers
 - Matching CP-nets
 - Matching Behaviour
- Practical Use of Annotations
- Future Work and Conclusions

Annotations

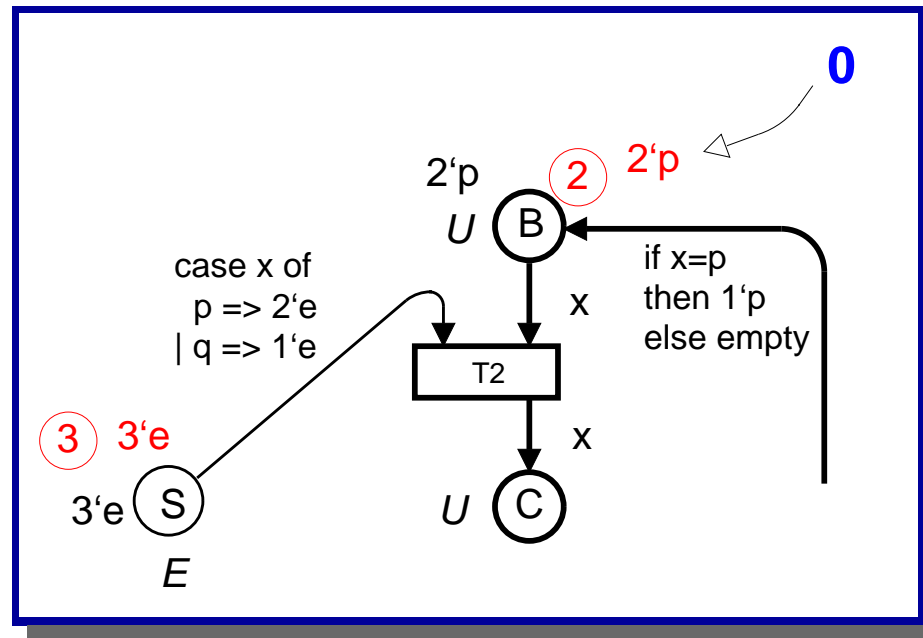
annotation (*noun*) a note added by way of comment or explanation

— *Merriam-Webster's Collegiate Dictionary*

- ❑ Method for adding **auxiliary information** to tokens in CP-nets.
- ❑ An **annotation** is a data value that is attached to a token.
- ❑ Annotations can be simple or arbitrarily complex.

Annotations

- Example: Adding cycle counters as annotations.



Annotation Layer

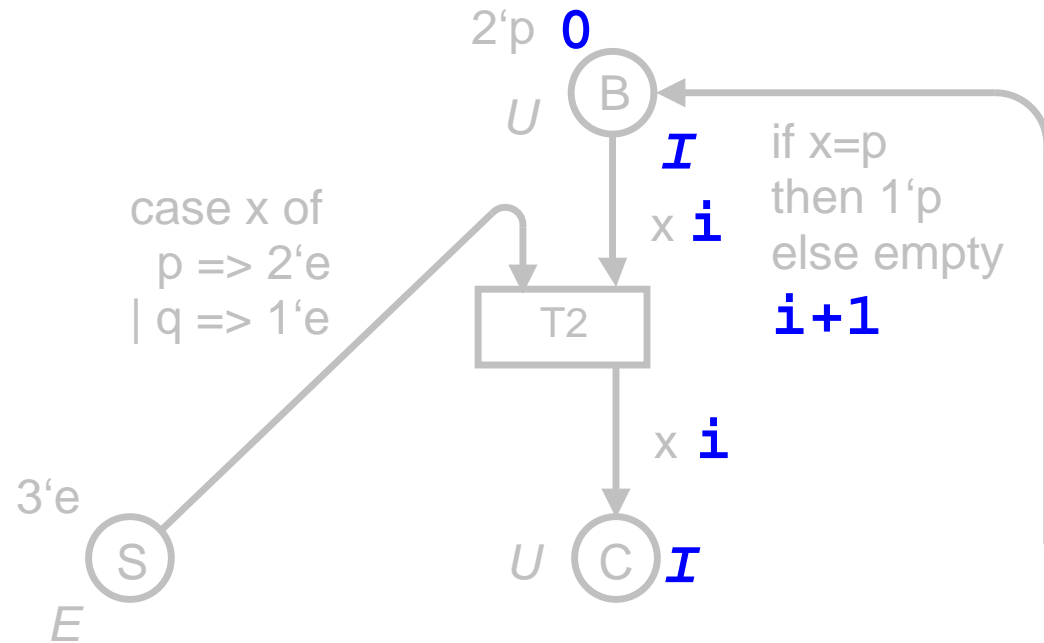
- An **annotation layer** is used to define annotations and how the annotations are to be associated with tokens in a particular CP-net.
- Annotations are defined separately from a CP-net.
- An annotation layer is defined for one CP-net, called the **underlying CP-net** of the annotation layer.
- An **annotated CP-net** is a pair consisting of a regular CP-net and an annotation layer for the CP-net.

Example: Annotated CP-net

```

color U = with p | q;
color E = with e;
var x : U;

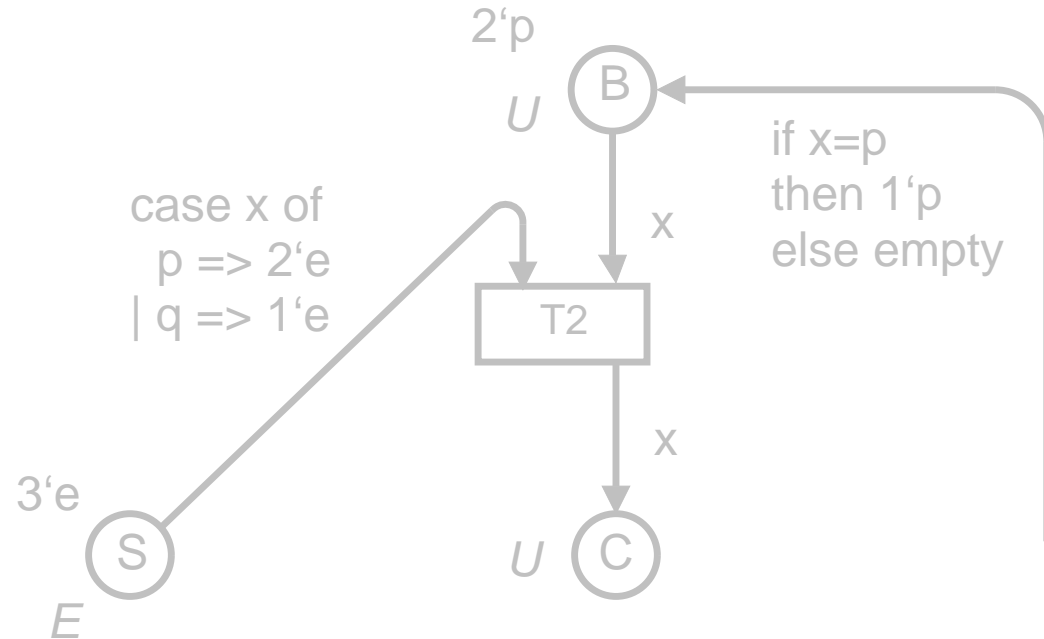
color I = int;
var i : I;
    
```



Annotation layers contain **auxiliary declarations** and **auxiliary net inscriptions**.

Example: Auxiliary declarations

```
color U = with p | q;  
color E = with e;  
var x : U;  
color I = int;  
var i : I;
```

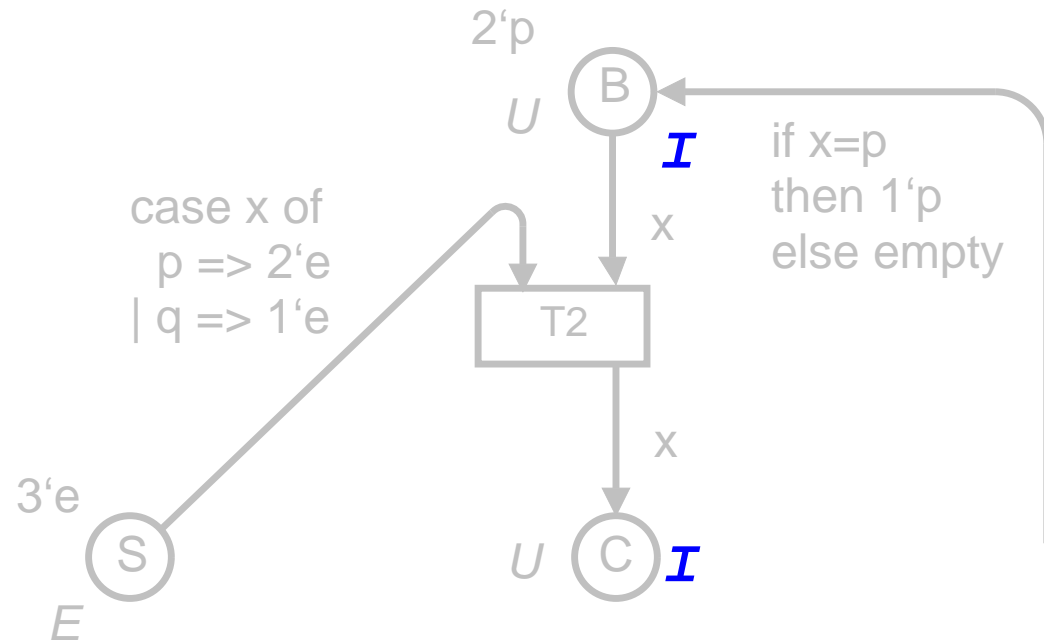


Elements that are declared as **auxiliary declarations** can only be used in the annotation layer.

Example: Auxiliary colour sets

```

color U = with p | q;
color E = with e;
var x : U;
    
```

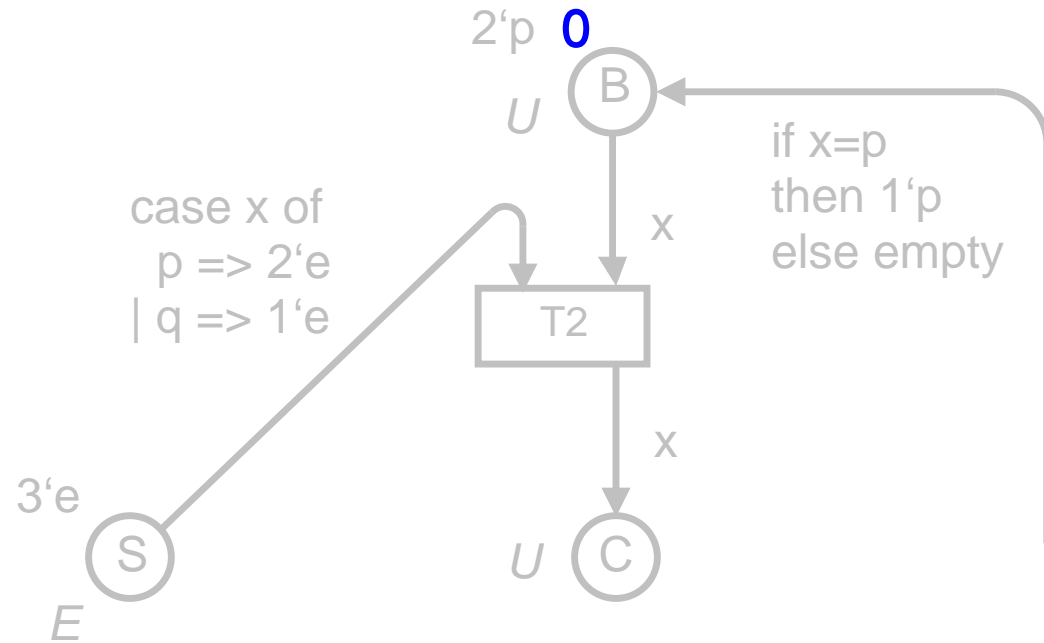


Auxiliary colour sets are associated with annotated places.

— Example: Auxiliary initialisation inscriptions —

```

color U = with p | q;
color E = with e;
var x : U;
    
```

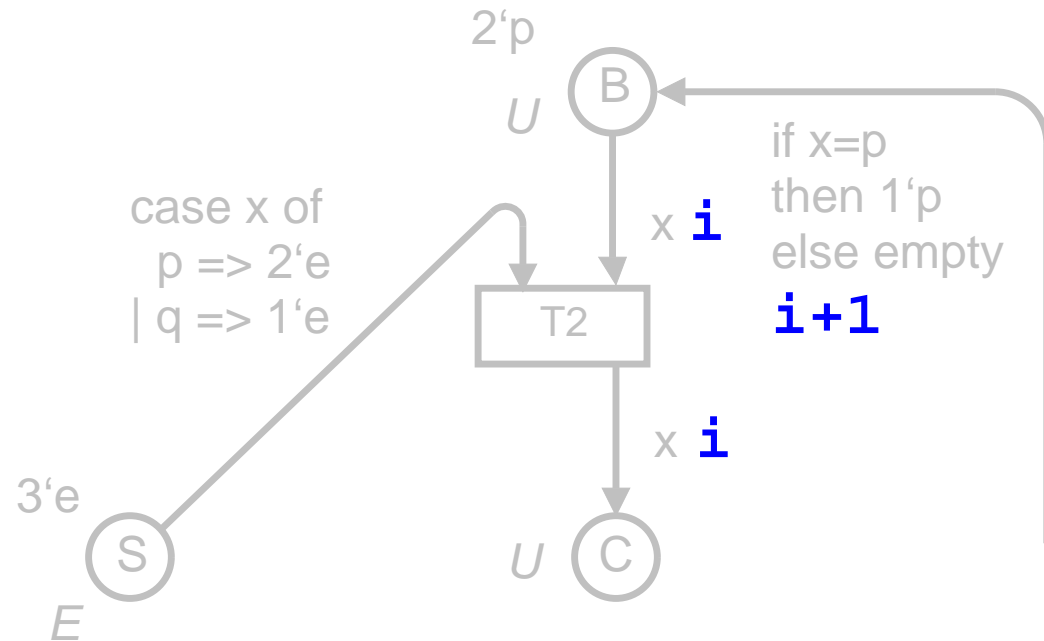


Auxiliary initialisation inscriptions must evaluate to a single annotation.

Example: Auxiliary arc inscriptions

```

color U = with p | q;
color E = with e;
var x : U;
    
```

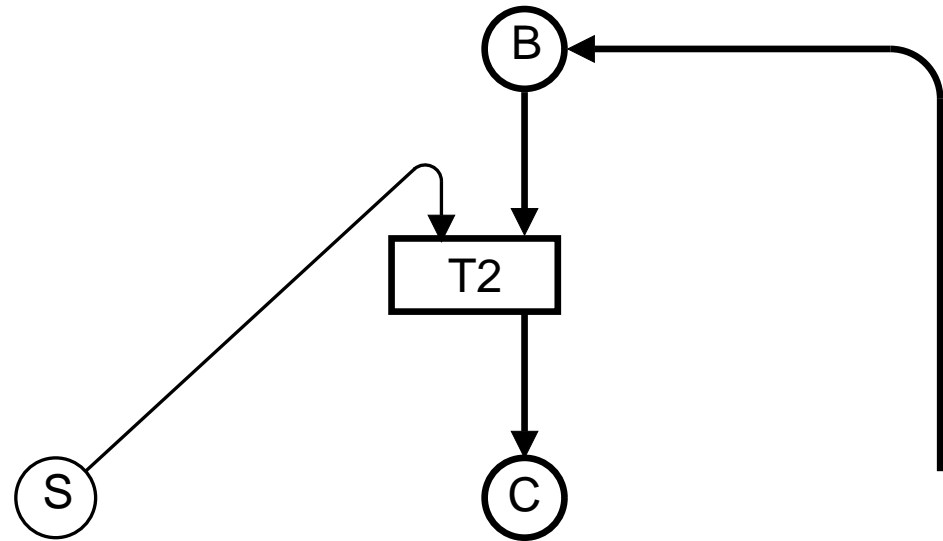


Auxiliary arc inscriptions must evaluate to a single annotation.

Matching CP-nets

- ❑ No new semantics are defined for annotated CP-nets.
- ❑ Each annotated CP-net can be translated to a **matching CP-net**.
- ❑ Translation is straightforward and can be automated in a CPN tool.
- ❑ If certain conditions are fulfilled, then the behaviour of the matching CP-net is said to **match** the behaviour of its underlying CP-net.

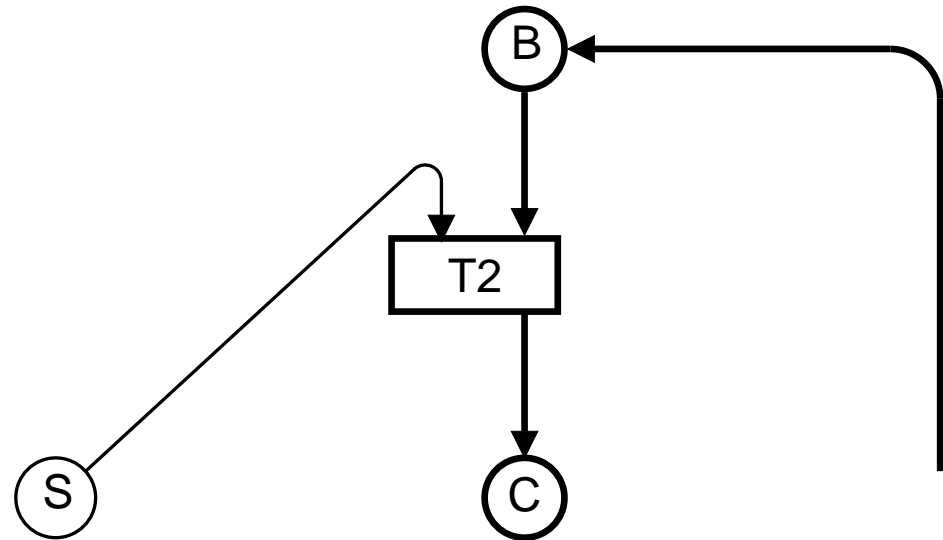
Example: Matching CP-net (1)



Same net structure as underlying CP-net.

Example: Matching CP-net (2)

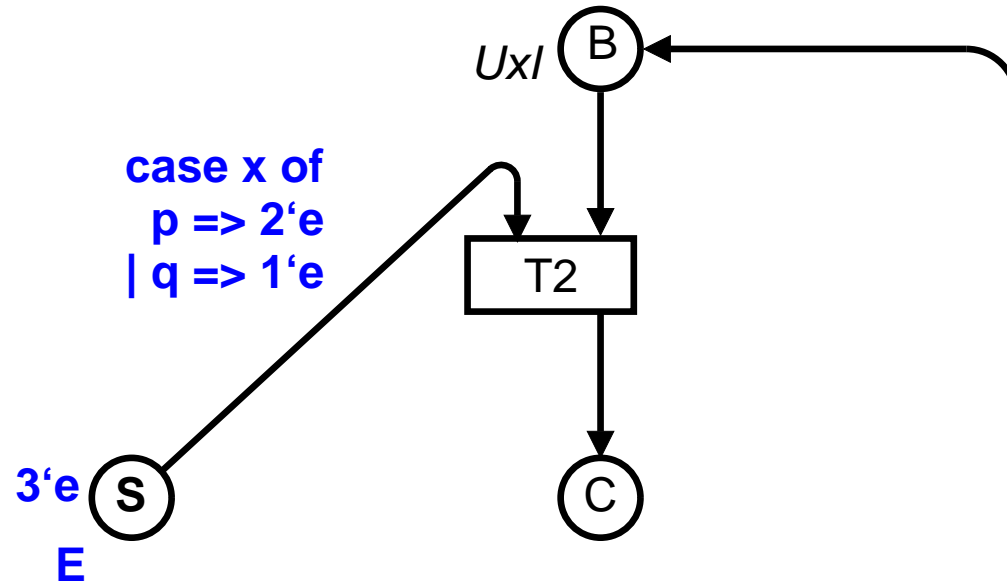
```
color U = with p | q;  
color E = with e;  
var x : U;  
  
color I = int;  
var i : I;  
color UxI = product U * I;
```



Declarations from underlying CP-net and annotation layer are combined.

Example: Matching CP-net (3)

```
color U = with p | q;  
color E = with e;  
var x : U;  
  
color I = int;  
var i : I;  
color UxI = product U * I;
```



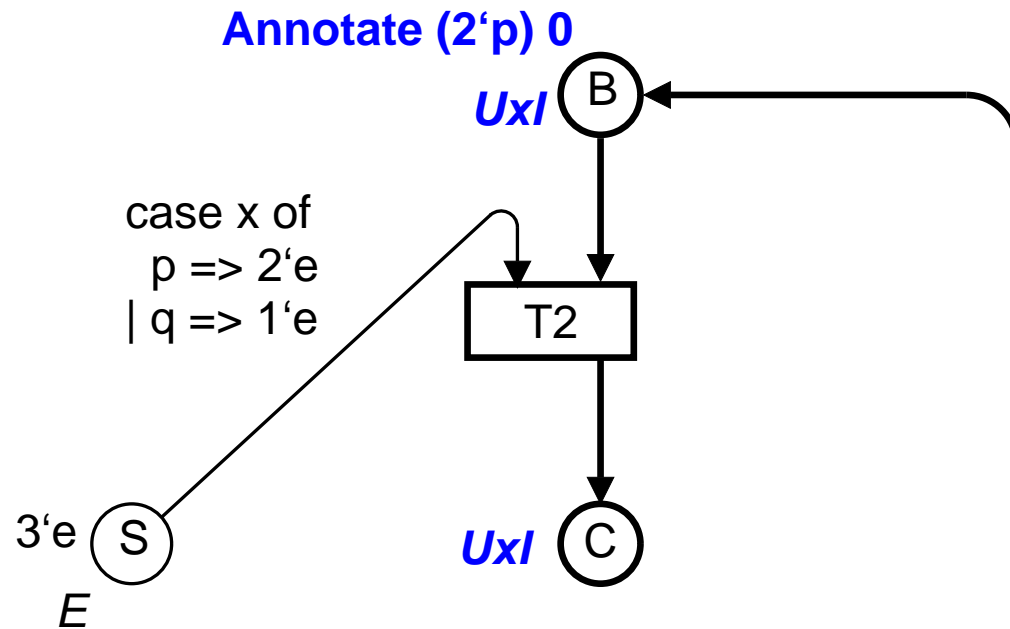
Inscriptions for **non-annotated places** and **non-annotated arcs** are taken from the underlying CP-net.

Example: Matching CP-net (4)

```

color U = with p | q;
color E = with e;
var x : U;

color I = int;
var i : I;
color UxI = product U * I;
    
```



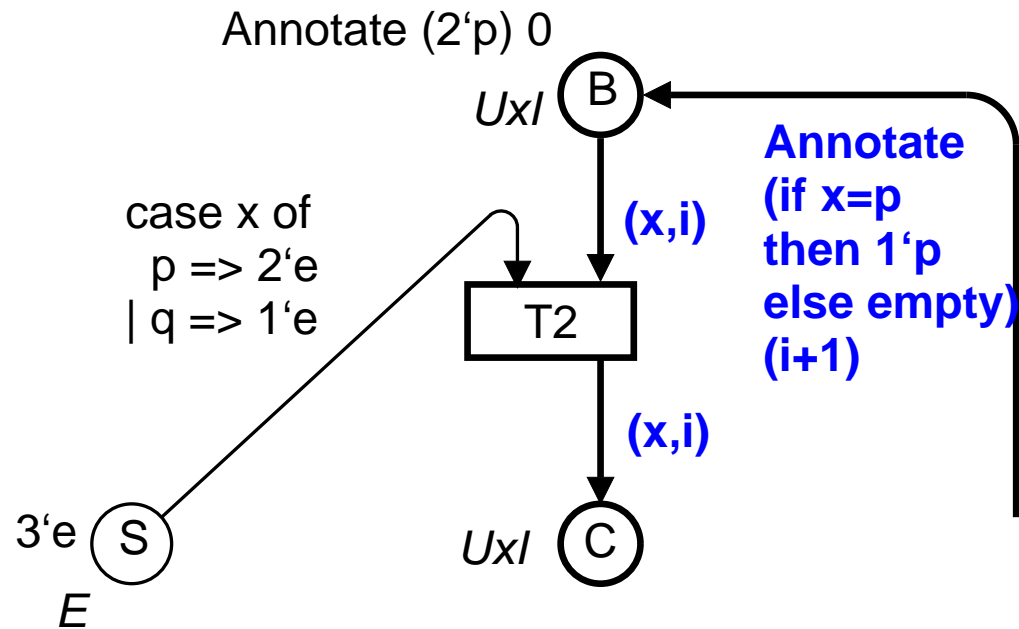
Inscriptions for **annotated places** are generated from corresponding inscriptions in the underlying CP-net and the annotation layer.

Example: Matching CP-net (5)

```

color U = with p | q;
color E = with e;
var x : U;

color I = int;
var i : I;
color UxI = product U * I;
    
```



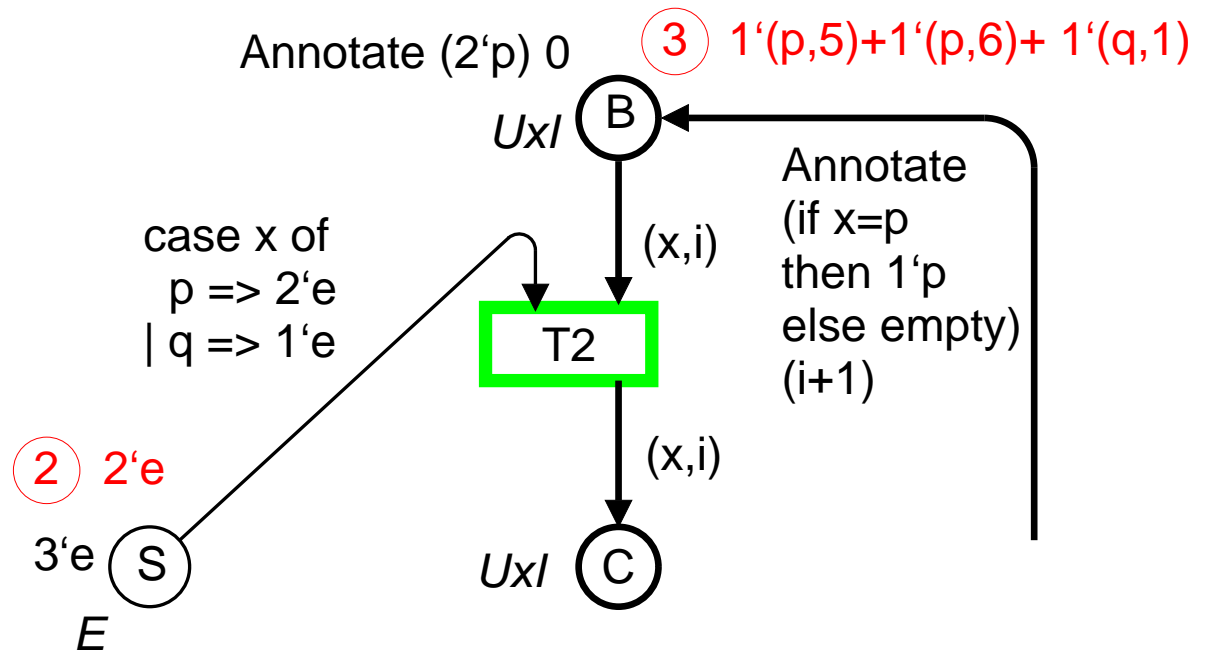
Inscriptions for **annotated arcs** are generated from corresponding inscriptions in the underlying CP-net and the annotation layer.

Example: Matching CP-net (6)

```

color U = with p | q;
color E = with e;
var x : U;

color I = int;
var i : I;
color UxI = product U * I;
    
```

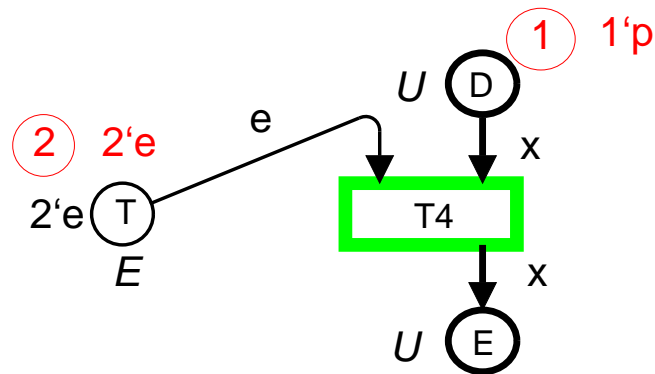


- Annotations are integrated in colours.
- Markings and bindings of a matching CP-net are said to **cover** markings and bindings in its underlying CP-net.

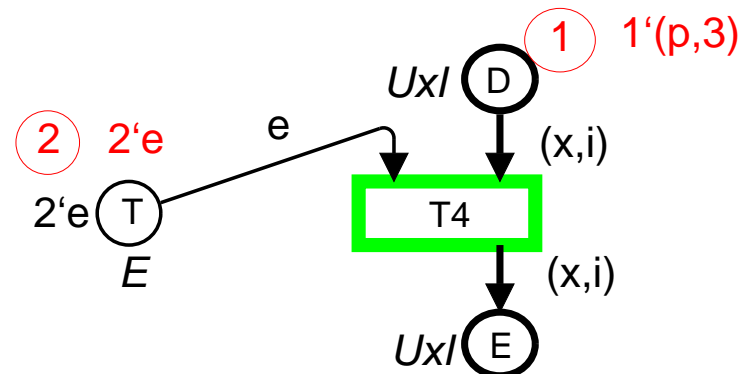
Sound Annotation Layers

If an annotation layer is **sound**, then we can guarantee that annotations will affect the behaviour of the underlying CP-net in a very limited and predictable way. Soundness property:

If a transition is enabled in marking M in the underlying CP-net, and if marking M^* from the matching CP-net covers M , then t must be enabled in M^* .

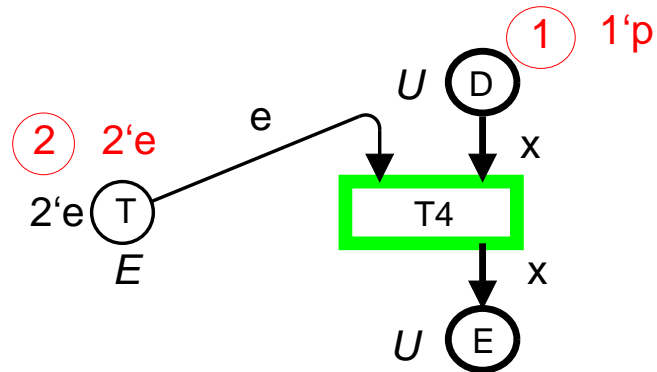


Underlying CP-net

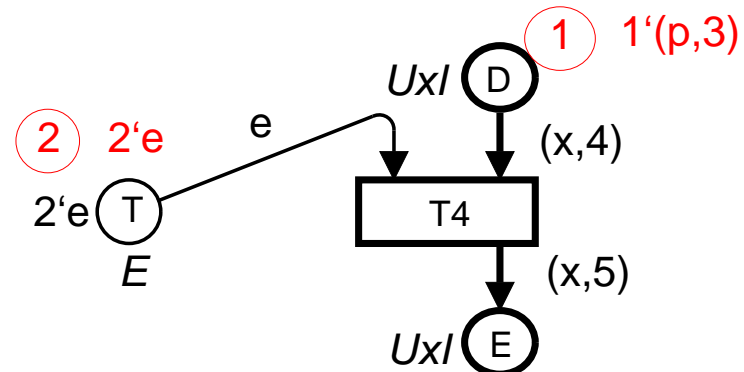


Matching CP-net

Example: Unsound Annotations



Underlying CP-net



Matching CP-net

- ❑ Only certain kinds of auxiliary arc inscriptions can be used in sound annotation layers.
- ❑ Tool implementors must determine what kinds of auxiliary arc inscriptions are allowable.

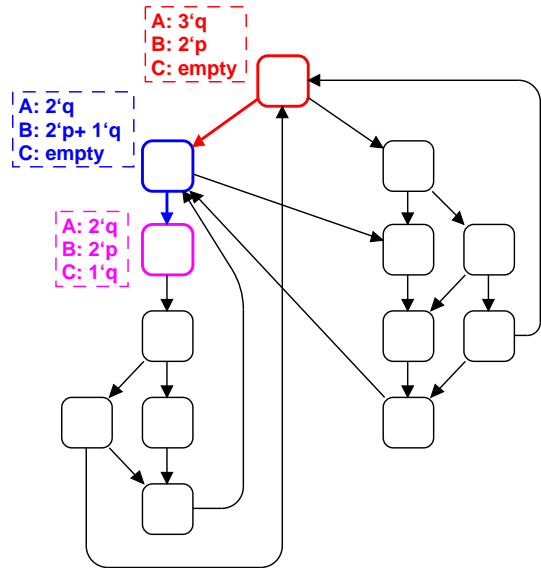
Matching Behaviour

If a matching CP-net obtained from a *sound* annotation layer and its underlying CP-net, then the behaviour of the matching CP-net is said to **match** the behaviour of the underlying CP-net.

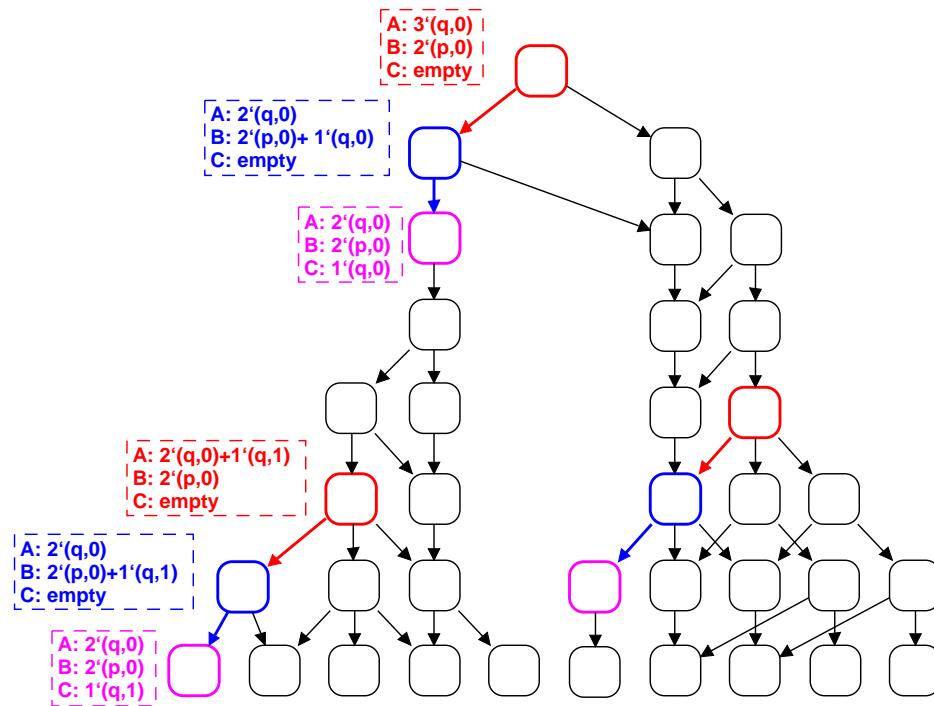
Theorem Let $(\text{CPN}, \mathcal{A})$ be an annotated CP-net with a sound annotation layer. Let CPN^* be the matching CP-net derived from $(\text{CPN}, \mathcal{A})$. Let M_0 , \mathbb{M} , and \mathbb{Y} denote the initial marking, the set of all markings, and the set of all steps, respectively, for CPN. Similarly, let M_0^* , \mathbb{M}^* , and \mathbb{Y}^* denote the same concepts for CPN^* . Then we have the following properties:

1. $\pi(\mathbb{M}^*) = \mathbb{M} \wedge \pi(M_0^*) = M_0$.
2. $\pi(\mathbb{Y}^*) = \mathbb{Y}$.
3. $\forall M_1^*, M_2^* \in \mathbb{M}^*, \forall Y^* \in \mathbb{Y}^*: M_1^* [Y^* \rangle M_2^* \Rightarrow \pi(M_1^*) [\pi(Y^*) \rangle \pi(M_2^*)$
4. $\forall M_1, M_2 \in \mathbb{M}, \forall Y \in \mathbb{Y}, \forall M_1^*, M_2^* \in \mathbb{M}^*:$
 $M_1 [Y \rangle M_2 \wedge \pi(M_1^*) = M_1 \Rightarrow \exists Y^* \in \mathbb{Y}^*: \pi(Y^*) = Y \wedge M_1^* [Y^* \rangle M_2^* \wedge \pi(M_2^*) = M_2$

Example: Matching Behaviour



Underlying CP-net full state space



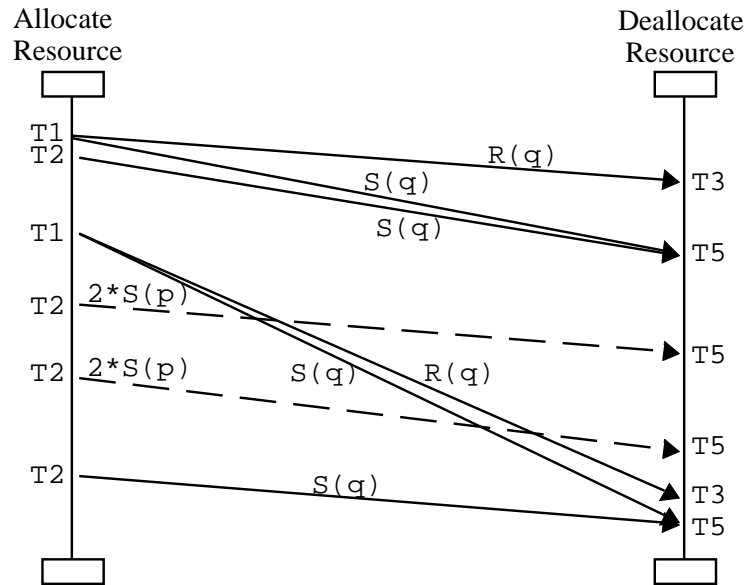
Matching CP-net partial state space

Every occurrence sequence in the matching CP-net covers an occurrence sequence in the underlying CP-net.

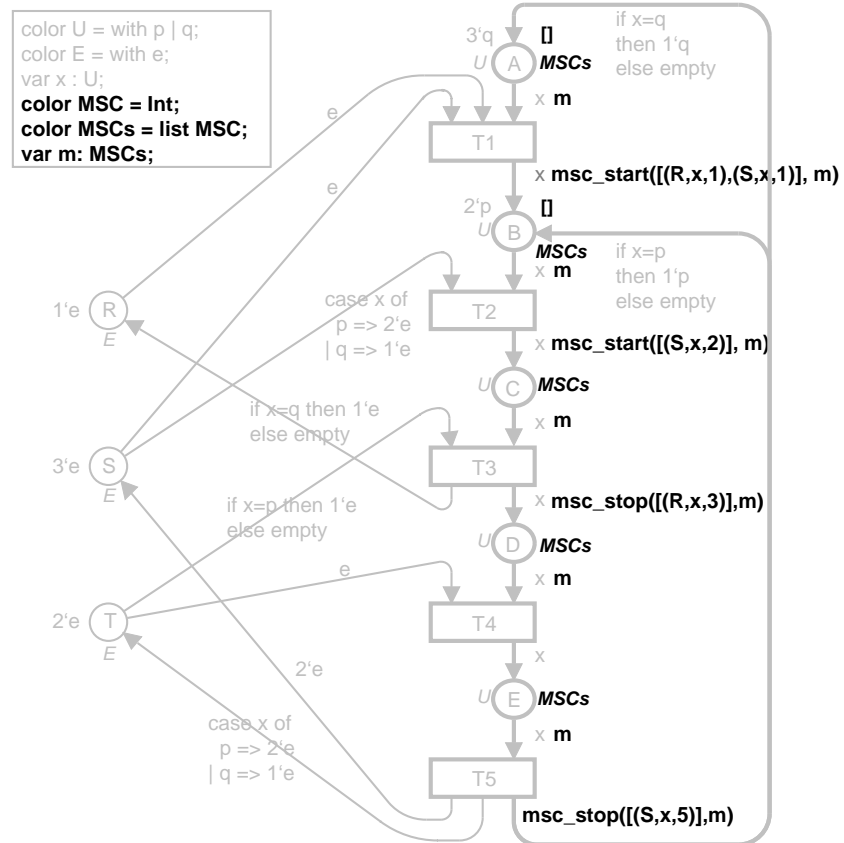
Plan

- Motivation
- Annotating Coloured Petri Nets
- **Practical Use of Annotations**
 - Message Sequence Charts
 - Multiple Annotation Layers
- Future Work and Conclusions

Practical Use of Annotations

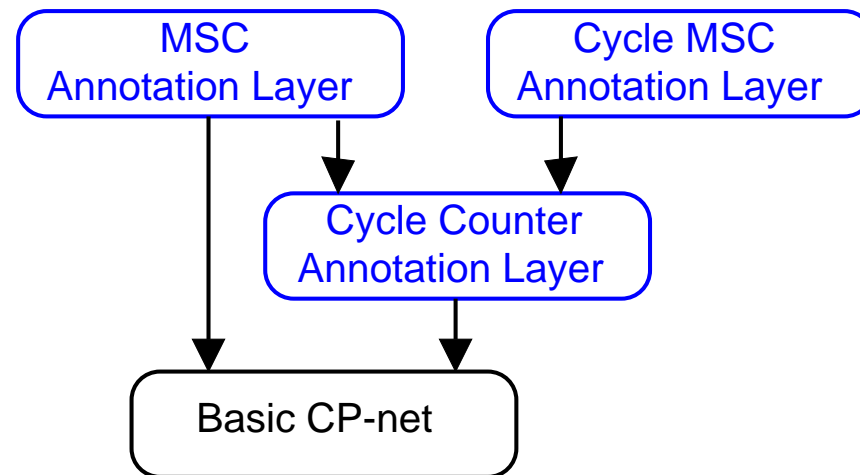


- Arrows drawn after two transitions occur.
- Annotations are arrow IDs.
- Annotations are allowed to have side effects.



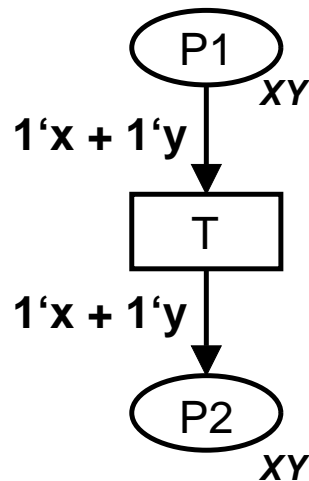
Multiple Annotation Layers

- Several “versions” of a CP-net can be maintained by creating different annotation layers.
- Different layers can be added, removed and combined for a variety of different purposes.

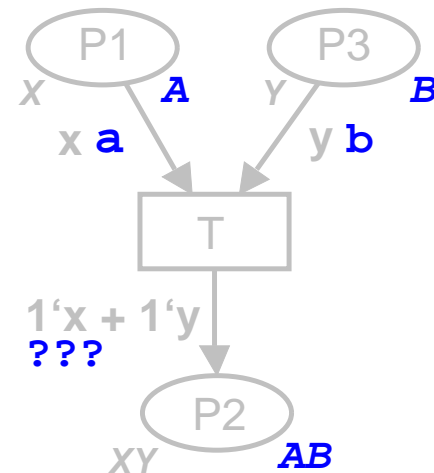


Future Work

- Implement tool support for annotating CP-nets!
- Remove or weaken some of the restrictions:
 - All arc inscriptions on arcs from annotated places to transitions in the underlying CP-net must always evaluate to a single colour.
 - Only one annotation can be added to a multi-set of colours.



Cannot annotate P1



Annotating multi-sets

Conclusions

- ❑ Method proposed for adding auxiliary information to CP-nets.
- ❑ Annotation layers support modular definitions of both CP-nets and one or more layers of auxiliary information.
- ❑ The same basic CP-net can be used for various purposes by adding, removing, and combining annotation layers.
- ❑ (Sound) Annotations affect the behaviour of a CP-net in a very predictable way.

- ❑ The paper contains formal requirements, definitions, and proofs.