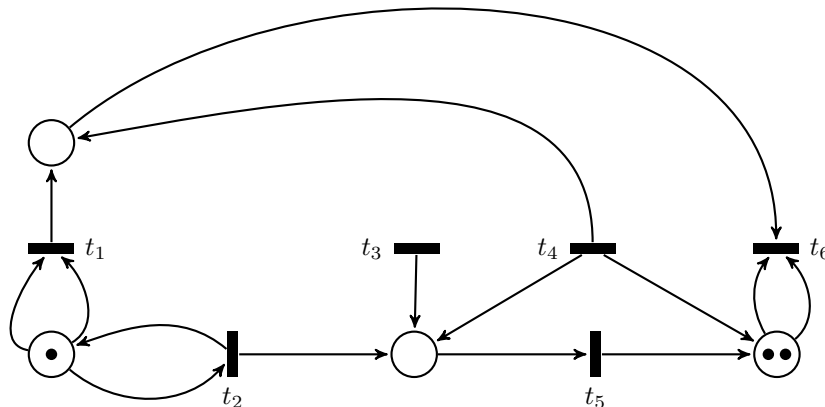


## Aufgabe 1 Fragen

- Welche formale Spezifikationssprache eignet sich besonders zur Spezifikation von nebenläufigen Systemen?
- Benenne die Kopplungsart: *Modul A übermittelt Daten an Modul B um die interne Logik von Modul B zu steuern*
- Benenne die Kohäsionsart: *Modul führt keine Aktionen auf gemeinsamen Daten aus, wobei die Reihenfolge relevant ist*
- Definiere folgende Begriffe:
  - Vollständigkeit (Einer Menge von Anforderungen)
  - Datenexterne Kopplung
  - Sequenzielle Kohäsion

## Aufgabe 2 Petri Netze

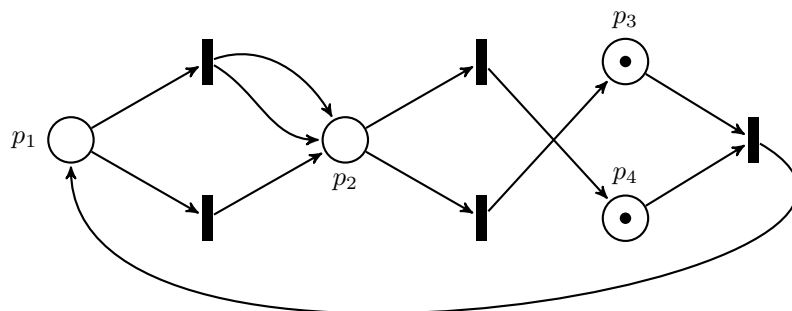
- a) Gegeben:



Geben Sie für alle Transitionen die Schaltbereitschaft bzgl. der aktuellen Markierung an.

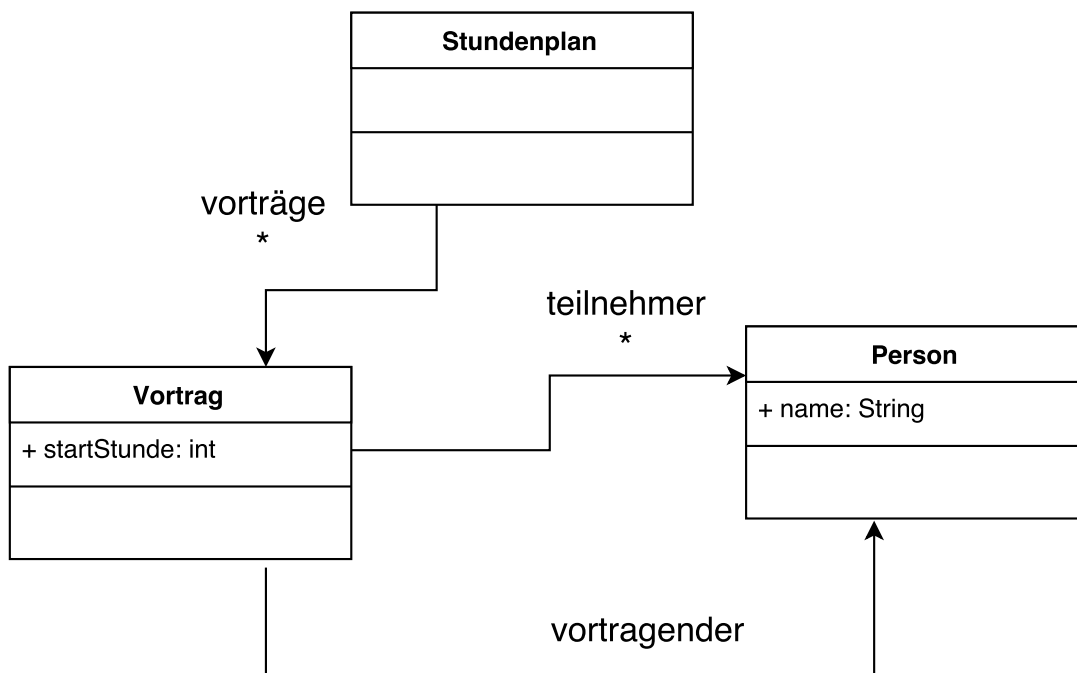
- |           |           |           |
|-----------|-----------|-----------|
| • $t_1$ : | • $t_3$ : | • $t_5$ : |
| • $t_2$ : | • $t_4$ : | • $t_6$ : |

b) Gegeben:



- a) Geben Sie alle Markierungen, die von der aktuellen Markierung erreicht werden können.
- b) Geben Sie alle Deadlock-Markierungen an, die von der aktuellen Markierung erreicht werden können.

### Aufgabe 3 OCL



Formulieren Sie folgende Aussagen in OCL

- a) Jeder Vortrag hat mindestens 2 und höchstens 16 Teilnehmer
- b) Für jeden Vortrag ist der Vortragende auch gleichzeitig Teilnehmer
- c) Kein Stundenplan enthält Vorträge, die zur gleichen Stunde beginnen
- d) Jeder Stundenplan enthält höchstens 3 Vorträge des gleichen Vortragenden

Formulieren Sie die folgenden OCL Ausdrücke in natürlicher Sprache

- a) context Stundenplan  
inv: not vorträge->exists(v | v.startStunde = 12 or v.startStunde = 13)
- b) context Person  
inv: Vortrag.allInstances()->select(v | v.teilnehmer->includes(self))->size()=1
- c) context Stundenplan  
inv: vorträge->forall(v | v.startStunde <> 8) implies  
vorträge->exists(v | v.startStunde = 18)
- d) context Stundenplan  
inv: vorträge->forall(v | not vorträge->exists(w | (v<>w)  
and (w.vortragender = v.vortragender) and  
((v.startStunde - w.startStunde <= 1) or  
(w.startStunde - v.startStunde <= 1))))