# Semantik der Aktivitätsdiagramme

Tong Tong

Betreuerin: Dr. María Victoria Cengarle

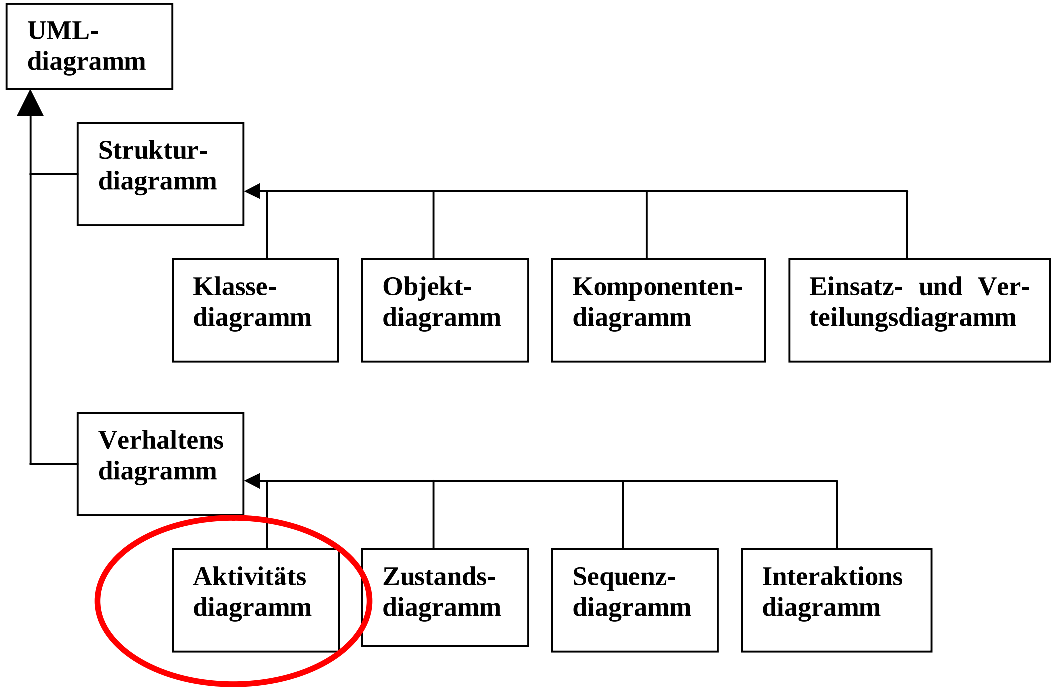
# Gliederung

- Einleitung
  - Motivation
  - Definition
- Die Syntax der Aktivitätsdiagramme
  - Abstrakte Syntax
  - Konkrete Syntax
- Semantik der Aktvitätsdiagramme
  - Grundlegende Definitionen
  - Semantik der Kontrollflüsse
  - Semantik der Aufruf Aktivitäten
- Konklusion

# Einleitung

- Die Unified Modeling Language (UML)
- Von OMG (Object Management Group)

1997 Version 1.1 → 1999 Version 1.3 → demnächst Version 2.0



# Motivation

- grafische Darstellung
- Darstellung komplexer Abläufe

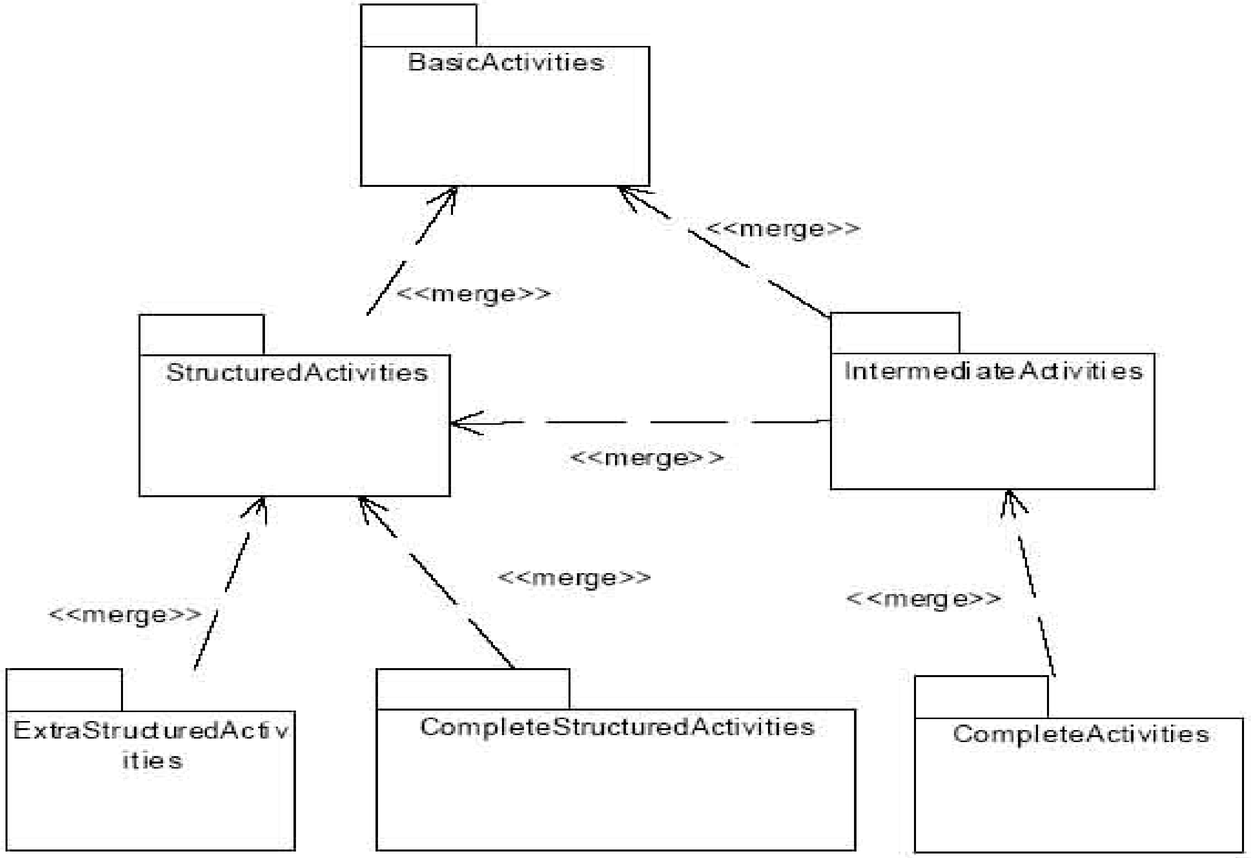
Deshalb werden Aktivitätsdiagramme als Sprache zur Definition von Abläufer genannt.

# Definition

- Ein Aktivitätsdiagramm besteht aus
  - Knoten
    - Kontrollknoten (Start- und Endknoten, u.s.w.)
    - Objektknoten
    - Aktionen
  - Kanten
    - Kontrollflüsse
    - Objektflüsse
- <Name, <Aktivitätsknoten, Aktivitätskanten>>

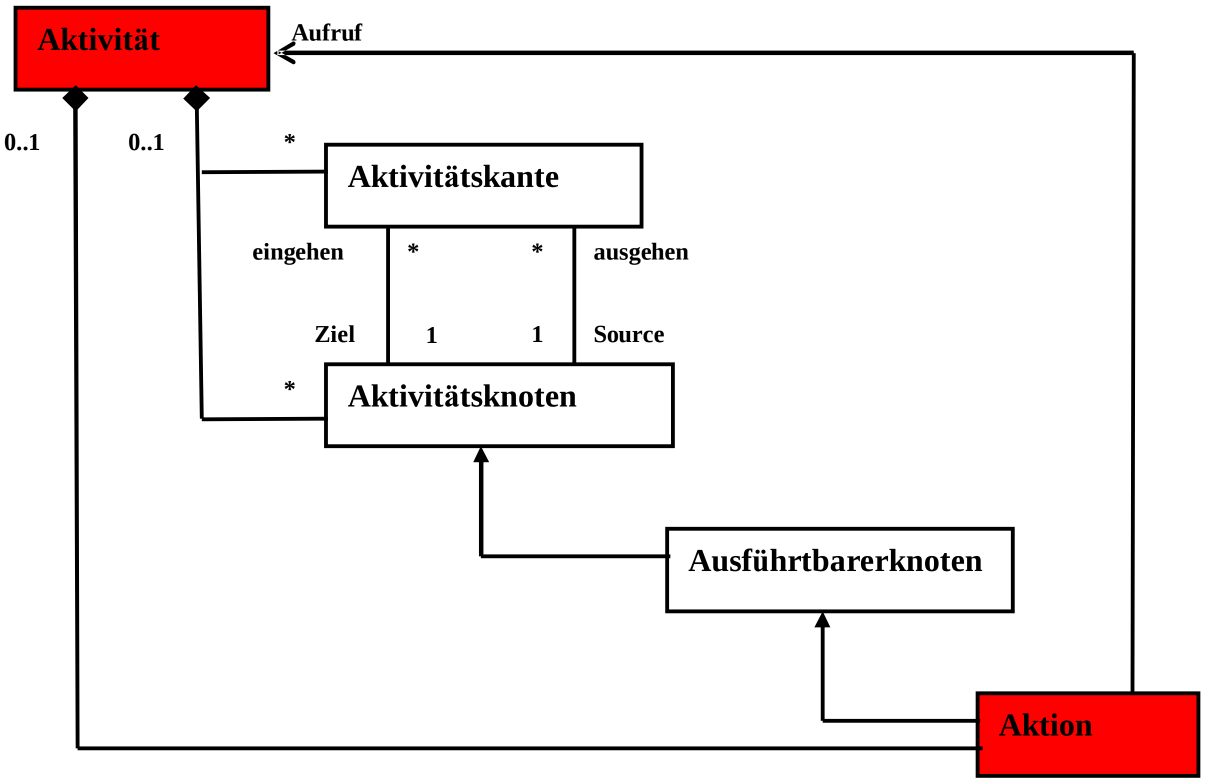
# Abstrakte Syntax

- Die Struktur von sechs Ebenen



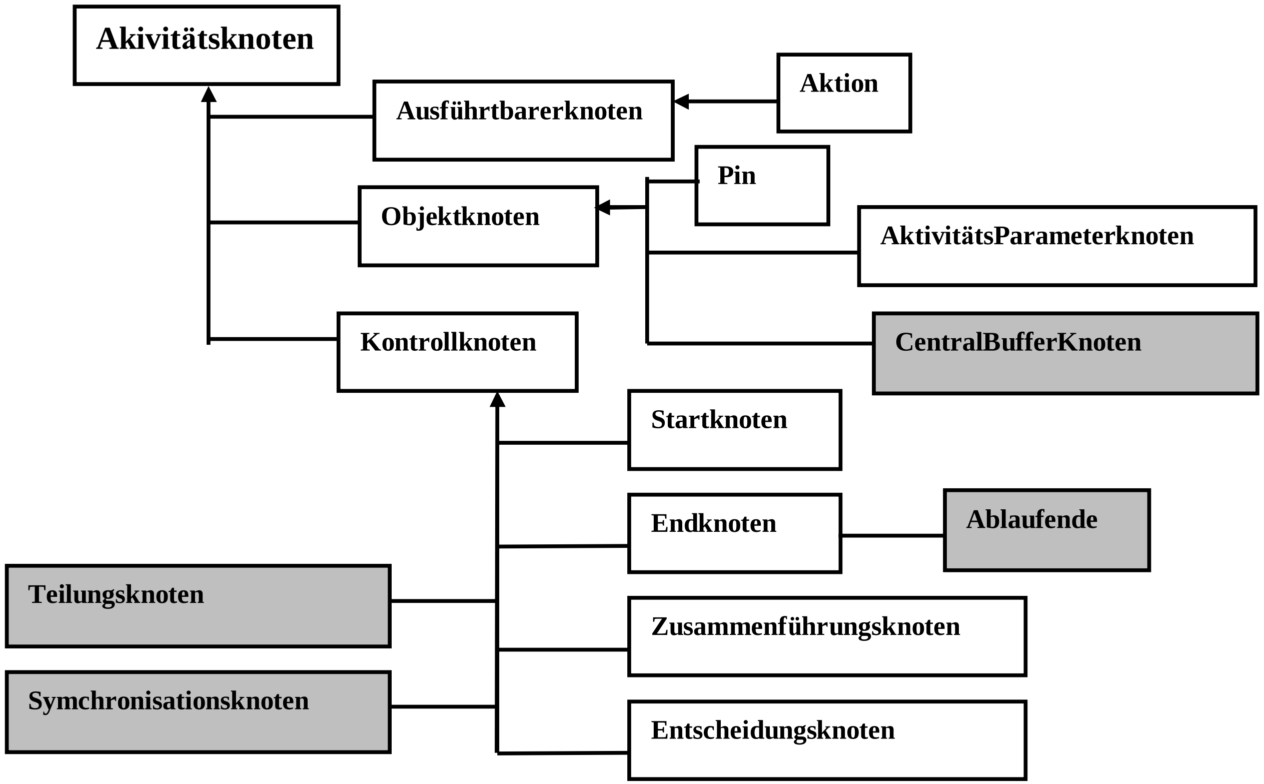
# Abstrakte Syntax

- Zwei grundlegende Entities



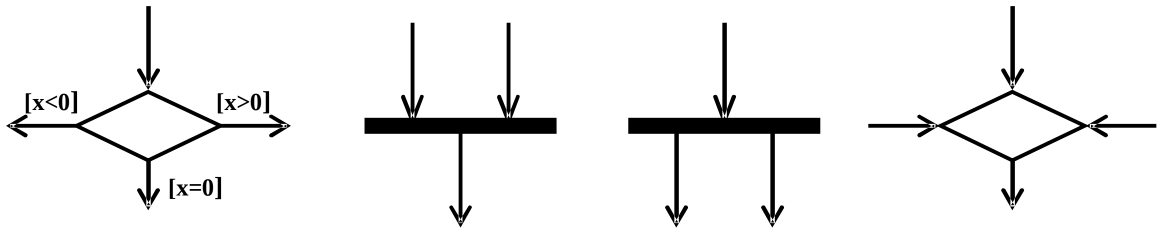
# Abstrakte Syntax

- BasicAcitivities und IntermediateAcitivities



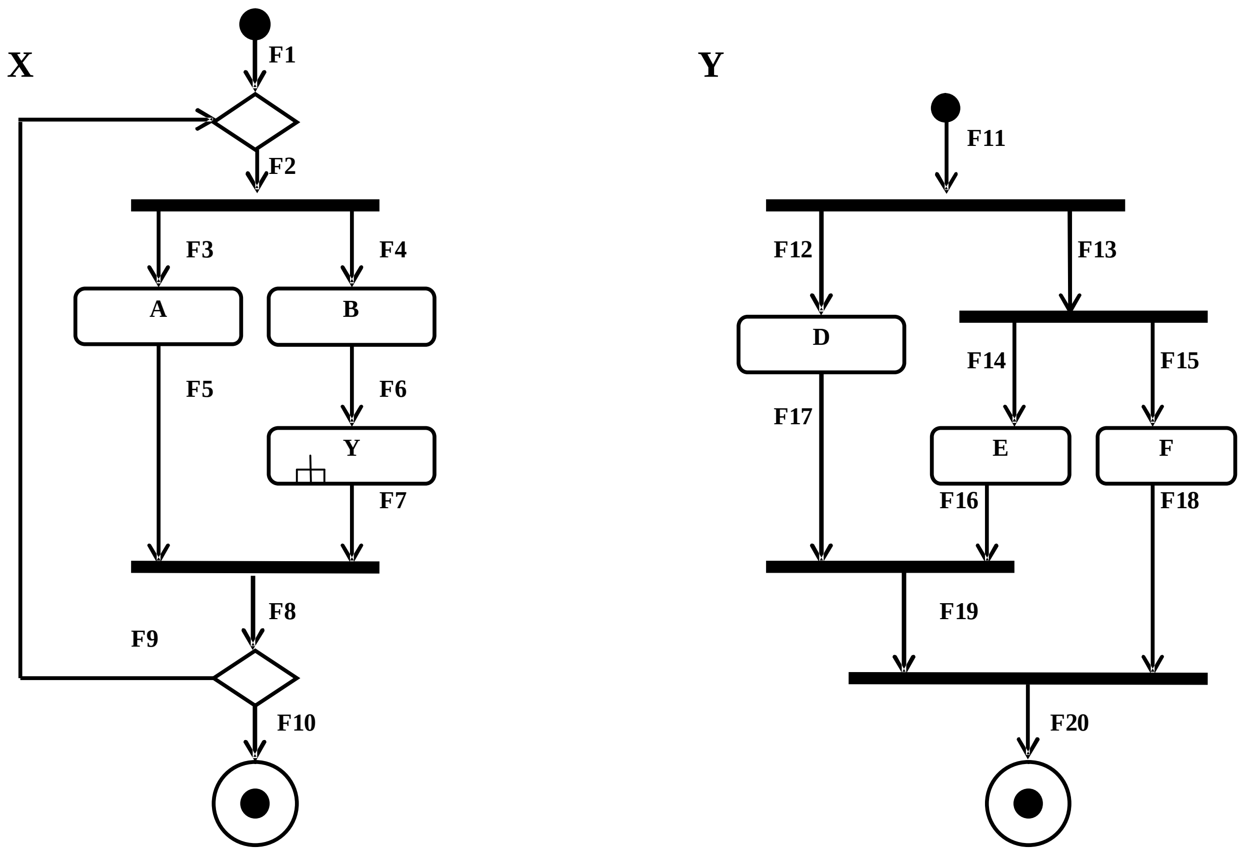
# Konkrete Syntax

- Aktivitätskanten
- Aktivitätsknoten
  - Ausführtbarerknoten: Aktion 1
  - CallBehaviorAktion: Name der aufgeruften Aktivität
  - Objektknoten: Aktion 1 Objekt [Zustand] oder Aktion 1 Objekt [Zustand]
  - Kontrollknoten: Startknoten Endknoten




Entscheidungsknoten Synchronisationssknoten Teilungssknoten Zusammenführungsknoten

# Ein Beispiel



# Petri-Netze

- 1962 von C.A. Petri,
- ermöglichen eine Graphen-orientierte Beschreibung verteilter Systeme und deren Abläufe,
- führen zu einer Theorie zur Analyse und Entwicklung nebenläufiger System.

# Petri-Netze

- Struktur Definition

Ein Petri-Netz ist ein Tripel <P, T, A> mit:

- $P \cap T = \emptyset$, P ist eine endliche Menge von Stellen,

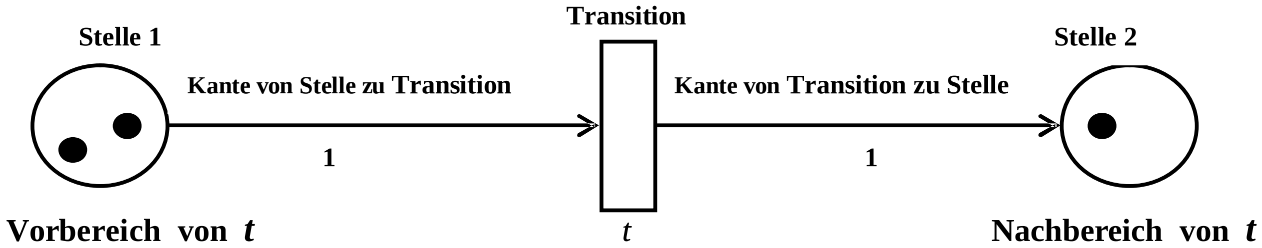
  T ist eine endliche Menge von Transitionen.
- $A \subseteq P \times T \cup T \times P$ ist eine endliche Menge von Kanten.
- Eine Kapazität c: P à $\mathbb{N} \cup \{\infty\}$

  Eine Gewichtung w: A à $\mathbb{N} \cup \{0\}$

  Eine Markierung m: P à IN mit $m(p) \leqslant c(p)$ für alle $p \in P$
- Für Netz-Elemente $x \in X = P \cup T$

  $\cdot x = \{ y \mid \langle y, x \rangle \in A \}$

  $x\cdot = \{ y \mid \langle x, y \rangle \in A \}$



# Vollständige Petri-Netze

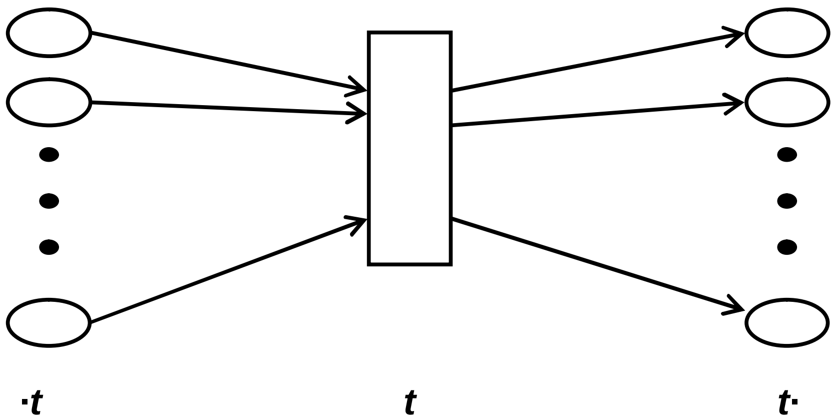
- Struktur Definition

  - Ein Quintupel $N = < P, T, A, \overline{m}, \underline{m} >$ ist ein Vollständiges Petri-Netz, wenn <P, T, A> ein Petri-Netz ist, $\overline{m}$ die Startmarkierung ist, und $\underline{m}$ die Endmarkierung ist.

# Vollständige Petri-Netze

- Verhalten Definition

  $N=\langle P, T, A, \overline{m}, \underline{m} \rangle$




*t* kann schalten, wenn:

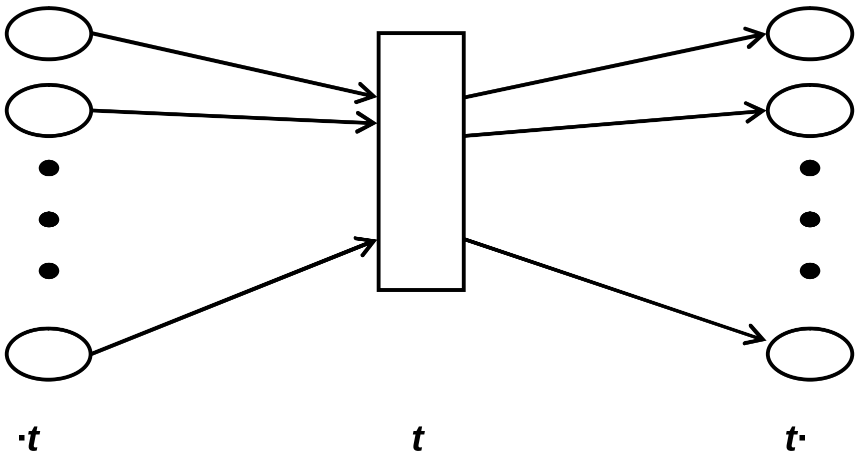
m($p$) $\geqslant$ w(<$p$, $t$>) für alle $p \in {\cdot}t$

c(q) $\geqslant$ w(<$t$, $q$>) + m(q) für alle $q \in t{\cdot}$

# Vollständige Petri-Netze

- Verhalten Definition

$N=\langle P, T, A, \overline{m}, \underline{m} \rangle$, $m \xrightarrow{t} m'$




$m'(p)=m(p)-w(\langle p, t\rangle)$ für alle $p \in {}^{\bullet}t$

$m'(q)=m(q)+w(\langle t,q\rangle)$ für alle $q \in t^{\bullet}$

$N$: $\overline{m} = m_0 \rightarrow m_0' = m_1 \rightarrow m_1' = \ldots\ldots \rightarrow m_n = \underline{m}$

# Prozedurale Petri-Netze

- Struktur Definition: NS=<N, $\rho$ >
  - N endl.Menge Vollständiger Petri-Netze jeweils mit zusätzlichen partiellen Funktion $act_N: T_N \rightarrow A$
  - $\rho : T_N \rightarrow N$ partieller Funktion für Aufruf. $(T_N = \bigcup_{N \in \mathrm{N}} T_N)$
  - Zustände von NS sind Elemente von $C \times I \times M \times F$ :
    - C enthält die Aufruf-Transitionen
    - I ist die Menge der eindeutigen Instanzidentifikatoren
    - M ist die Menge der Netzmarkierungen
    - F ist die Menge der aktuellen Aufrufe.

# Prozedurale Petri-Netze

- Verhaltens Definition
  - Definition für normale Transitionen:
    - Transition t ist nicht gehört zu dom( $\rho$ ) und schaltbereich in Zustand s.
    - wenn <c, i, m, f> zu s gehört, und entweder t zu $T_{\rho(c)}$ gehört order $c=\perp$, wird der neue Zustand s' durch t erzeugt mit s'=s-<c, i, m, f>+<c, i, m', f>.

# Prozedurale Petri-Netze

- Verhaltens Definition
  - Definition für prozedurale Aufruf-Transitionen:
    - Eine Transition t ($t \in dom(\rho)$) in Zustand s kann eine prozedurale Aufruf-Transition sein: $s \xrightarrow{t_{call}}$
    - Wenn t schaltet, werden eine neue Instanz i' und ein neuer Zustand s' erzeugt: $s \xrightarrow{t^{i'}_{call}} s'$
    - $s' = s - \langle c, i, m, f \rangle + \langle c, i, m - \cdot t, f \cup \{i'\} \rangle + \langle t, i', \overline{m}_{\rho(t)}, \emptyset \rangle$

# Prozedurale Petri-Netze

- Verhaltens Definition
  - Definition für prozedurale Rückkehr-Transitionen:
    - Eine Transition t in Instanz i' kann eine Rückkehr-Transition zu s sein: $s \xrightarrow{t^{i'}_{return}} s'$
    - Durch Schalten von t , wird ein neuer Zutand s' erzeugt, und die Instanz i' entfernt.
    - $s' = s - \langle c,i,m,f \cup \{i'\}\rangle + \langle c,i,m+t^{\cdot},f\rangle - \langle t, i', \underline{m}_{\rho(t)}, \varnothing\rangle$

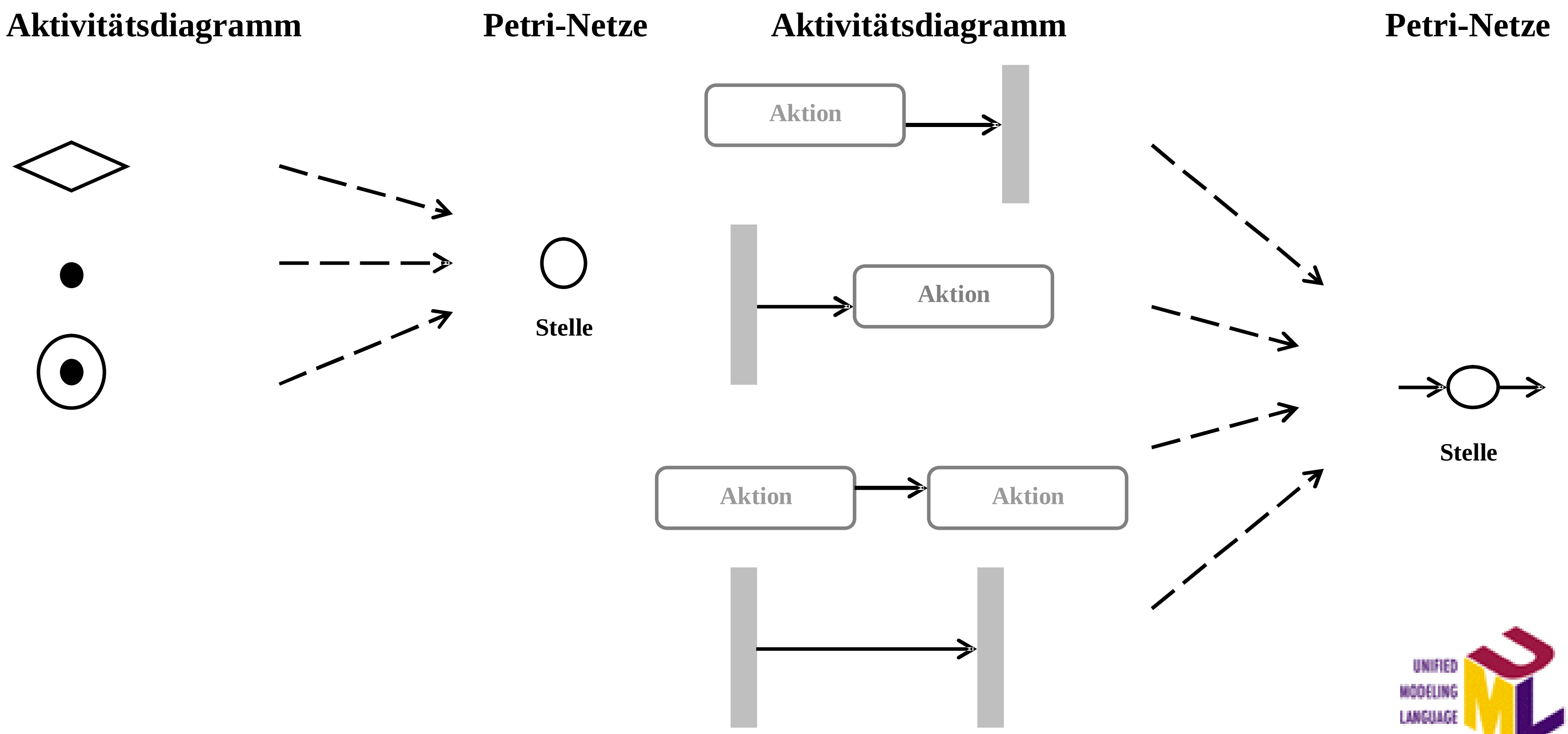
# Semantik der Kontrollflüsse

- $[\![ \langle \text{Aktivitätsknoten, Aktivitätskanten} \rangle ]\!] = \langle P, T, A, \overline{m}, \underline{m} \rangle$
  (ohne Objektknoten, ohne Objektflüsse)

- Aktivitätsknoten: <EN, iN, fN, BN, CN>
  - EN ist eine Menge von der Ausführtbarerknoten (Aktionen).
  - iN, fN sind die Startknoten und Endknoten.
  - BN ist eine Menge von der Zusammenführungsknoten und Entscheidungsknoten.
  - CN ist eine Menge von der Teilungsknoten und Synchronisationsknoten.

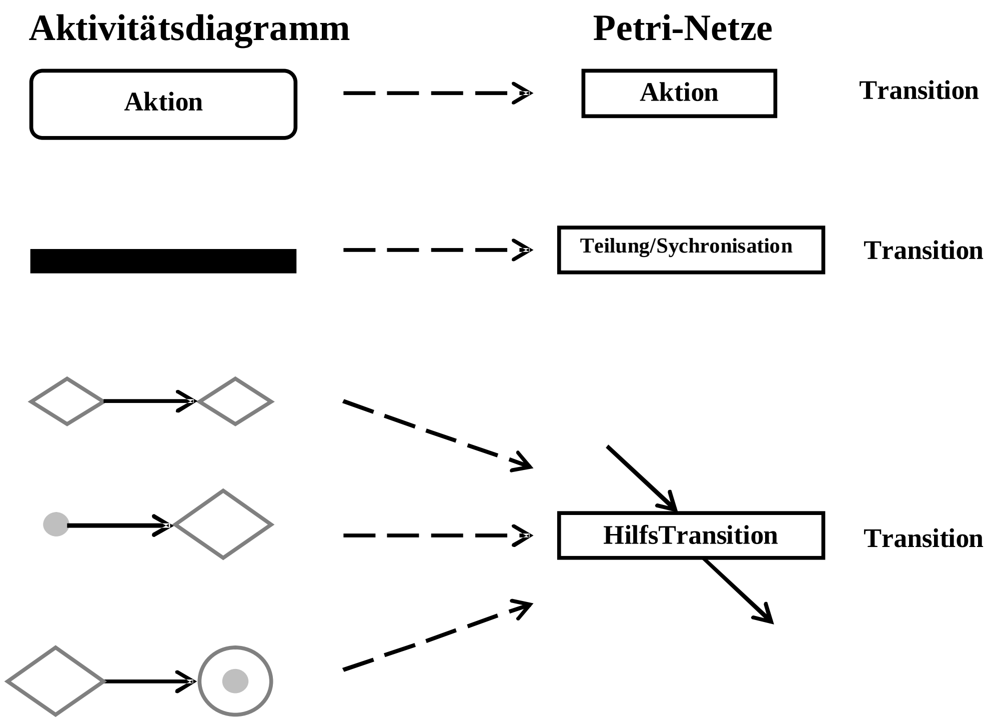
# Semantik der Kontrollflüsse

- $P = \{iN, fN\} \cup BN \cup \{ p_a \mid a \in \text{Kontrollflüsse}, \{a1, a2\} \cap (EN \cup CN) \neq \emptyset \}$



# Semantik der Kontrollflüsse

- $T = EN \cup CN \cup \{ t_a \mid a \in$ Kontrollflüsse, $\{a1, a2\}$ $BN \cup \{iN, fN\}\}$



# Semantik der Kontrollflüsse

- $A = \{<x_{<von,\ nach>}, nach>, <von, x_{<von,\ nach>}> \mid$

  $<von, nach> \in$ Kontrollkanten},

  bei der eine Seite ist

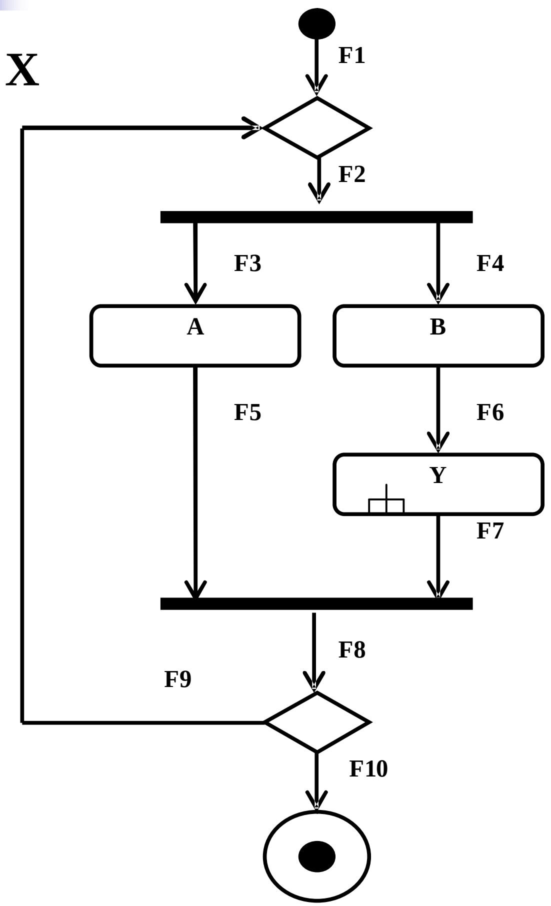
  $p_a$ ($p_a \in P$ und $a \in$ Kontrollflüsse) oder

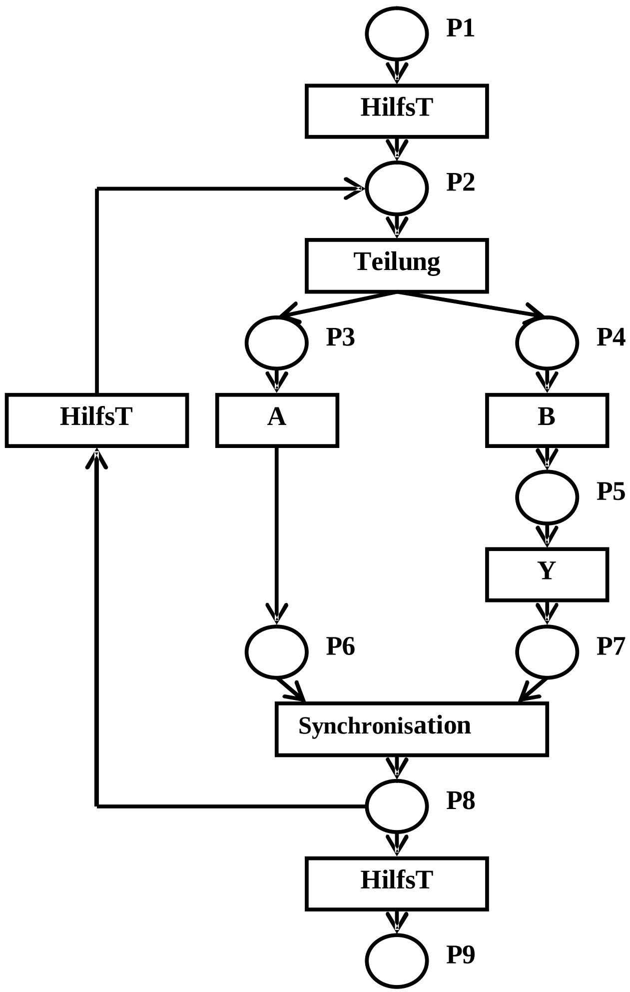
  $t_a$ ($t_a \in T$ und $a \in$ Kontrollflüsse).

- $\overline{m} = iN$

- $\underline{m} = fN$

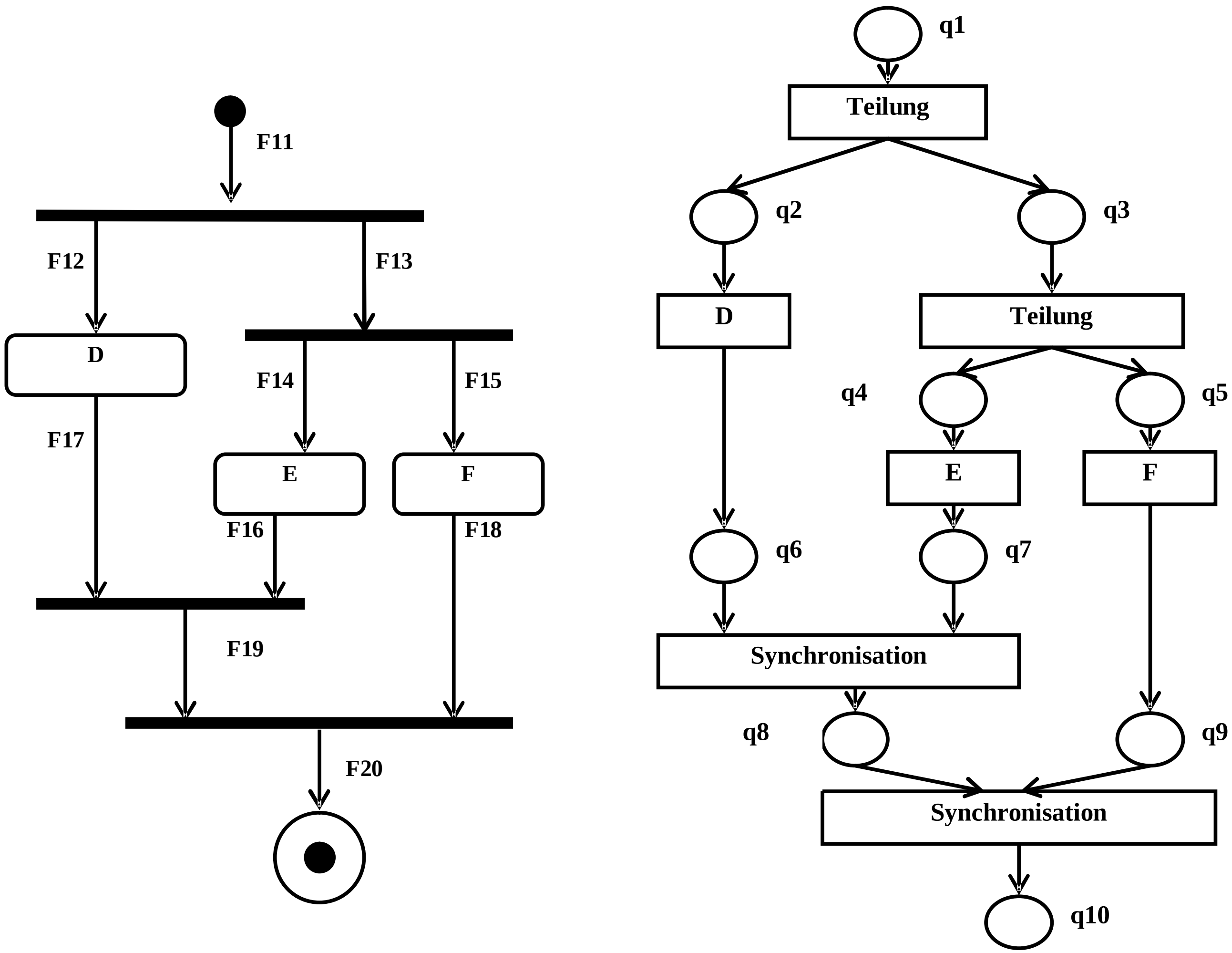
# Semantik des Beispiels



# Semantik des Beispiels

Y

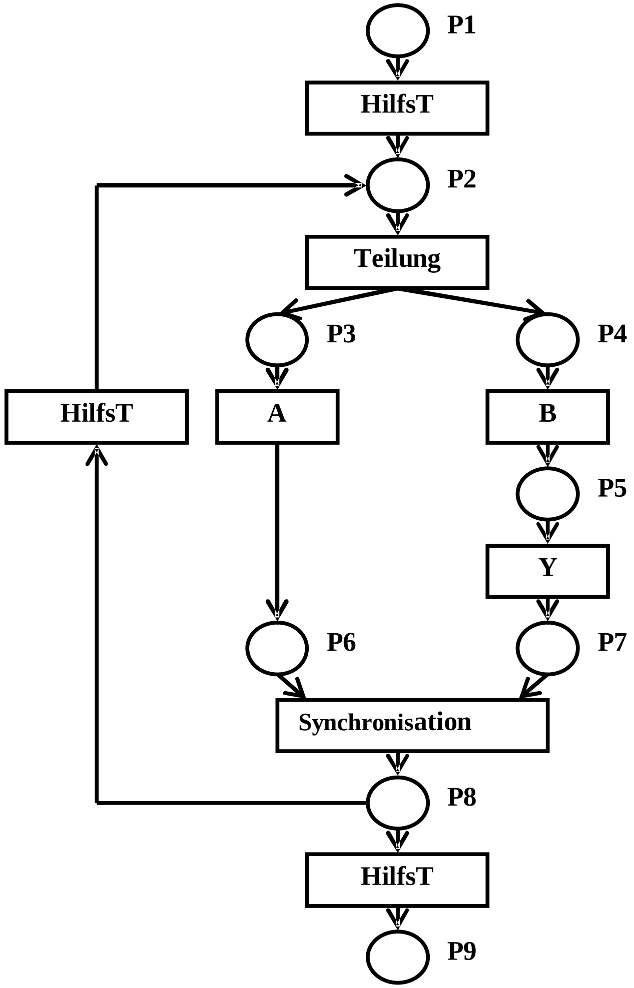


# Semantik der Aufruf Aktivitäten

- $[\![ \mathit{Spec} ]\!] = \langle N, \rho \rangle$
  - *Spec* ist eine Menge von Aktivitäten.
  - Es gibt Spitzen-Aktivität, die durch *top(Spec)* ausgewählt wird.
  - N ist {[Diagramm(A)] | A$\in$*Spec*}
  - $\rho$ ist eine Reihe von Aufruf-Beziehungen in *Spec*: {Name(A) $\mapsto$ [*act*(A)] | A$\in$*Spec*, *act*(A) ist definiert}
  - $\overline{m}_{Spec} = \langle \perp, 0, \overline{m}_{Spec}, \emptyset \rangle$
  - $\underline{m}_{Spec} = \langle \perp, 0, \underline{m}_{Spec}, \emptyset \rangle$

# Semantik des Beispiels

$[Spec]=<\{[X], [Y]\}, \{ Y \mapsto [Y] \}>$




$\overline{m}_{Spec}=<\perp, 0, p1, Ø>$

$\downarrow$ Hilfstransition

$<\perp, 0, p2, Ø>$

$\downarrow$ Teilungstransition

$<\perp, 0, p3, Ø><\perp, 0, p4, Ø>$

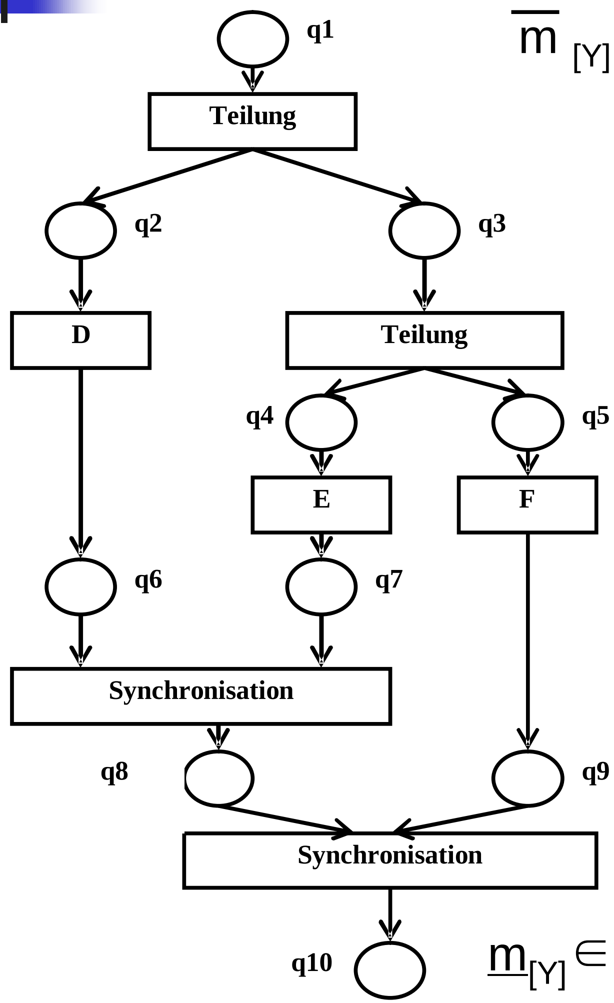
$\downarrow$ Transitionen A, B

$<\perp, 0, p6, Ø><\perp, 0, p5, Ø>$

$\downarrow$ Transition Y( $Y^{1}_{call}$ )

$\overline{m}_{[Y]}$

# Semantik des Beispiels




$\overline{m}_{[Y]}$ ∈ <⊥, 0, p6, {1}><Y, 1, q1, Ø>

Teilungstransition

<⊥, 0, p6, {1}><Y, 1, q2, Ø><Y, 1, q3, Ø>

Transition D,Teilungstransition

<⊥,0,p6,{1}><Y,1,q6,Ø><Y,1,q4,Ø><Y,1,q5,Ø>

Transitionen E, F

<⊥,0,p6,{1}><Y,1,q6,Ø><Y,1,q7,Ø><Y,1,q9,Ø>

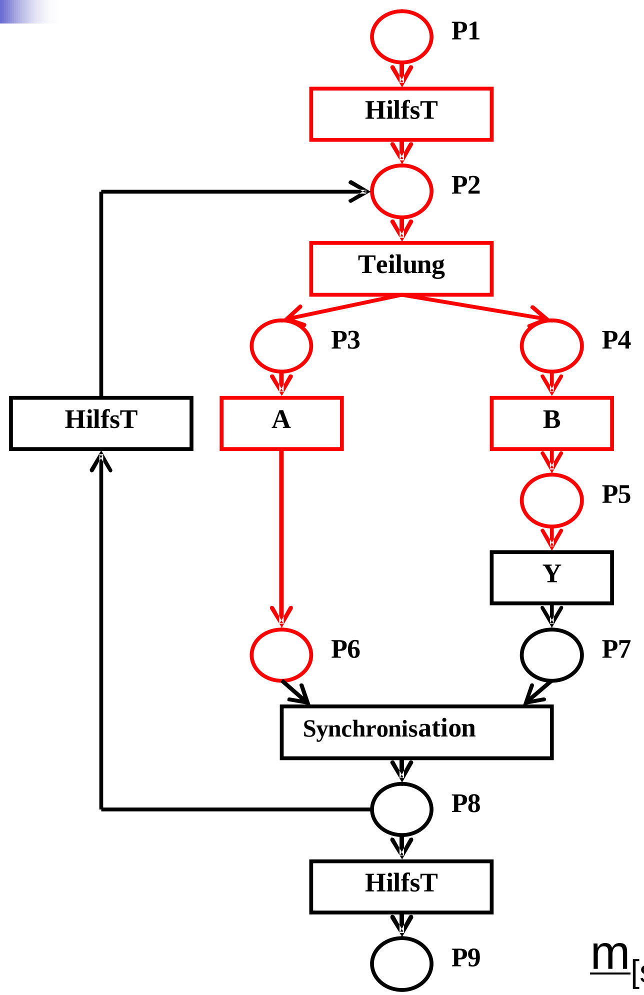
Sychronisationstransition

<⊥, 0, p6, {1}><Y, 1, q8, Ø><Y, 1, q9, Ø>

Sychronisationstransition

$\underline{m}_{[Y]}$ ∈ <⊥, 0, p6, {1}><Y, 1, q10, Ø>

# Semantik des Beispiels




$\underline{m}[Y]$

↓ Transition Y ($Y^{1}_{return}$)

$<\perp, 0, p6, \emptyset><\perp, 0, p7, \emptyset>$

↓ Synchronisationstransition

$<\perp, 0, p8, \emptyset>$

↓ Hilfstransition

$\underline{m}_{[spec]} = <\perp, 0, p9, \emptyset>$

# Konklusion

- Syntax der Aktivitätsdiagramme
- Semantik der Aktvitätsdiagramme
  - In UML V2.0 → Petri-Netze
  - In UML V1.X → Zustandsmaschinen
- Semantik der Kontollflüsse
  - komplette Petri-Netze
- Semantik der Aufruf-Aktivitäten
  - prozedurale Petri-Nezte

# Vielen Dank für Ihre Aufmerksamkeit!