2.5.3. Konflikte, Konfliktfreiheit

Fragen?

- sind alle Alternativen determiniert im PN gelöst?
- zu steuerndes System: nicht determinierte Alternativen sin Angriffspunkte für die Steuerung, Steuerung: nichtdet. Alternativen sind i.a. ein Fehler

Konflikt enspricht in etwa dem Widerspruch in Automaten.

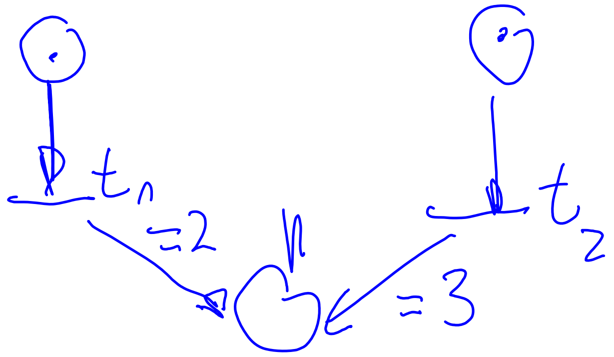
Def. Zwei oder mehrere Transitionen stehen in Konflikt, wenn sie gleichzeitg schaltfähig sind, das Schalten einer der oder den anderen wechselseitig die Schaltfähigkeit entzieht.

Bsp. PN 20 links Vorkonflikt, rechts ist für K(p9)=1 ein Nachkonflikt

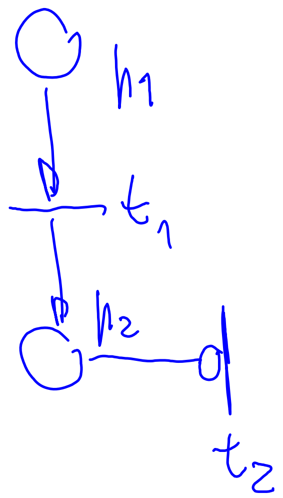
statischer Konflikt: es könnte evtl. eine m geben, bei der ein Konflikt auftritt

dynamischer Konflikt; es ist auch eine m von m0 aus erreichbar, bei der der Konflikt auftritt

Konflikt und Sonderkanten?

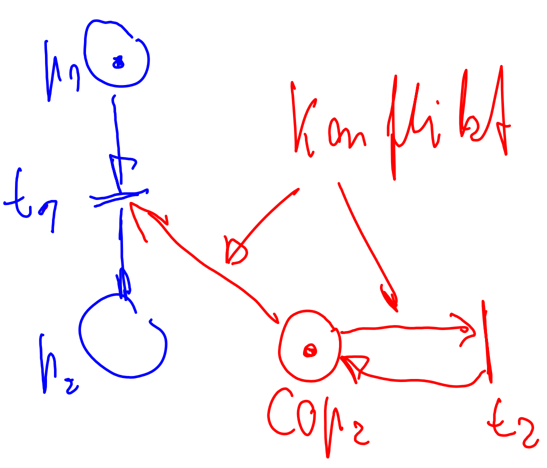


was ist nach gleichzeitigem Schalten von t_1 und t_2 in p?



Schalten von t_1 entzieht t_2 die Schaltfähigkeit t_2 aber t_1 nicht

Lösung über Ersatzkonstruktion



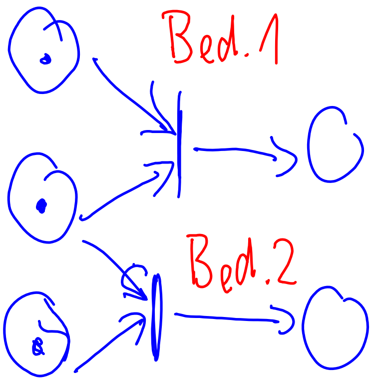
Lösung von Konflikten: Netz so ergänzen, dass im Konfliktfall entschieden ist welche Alternative gewählt wird.

Möglichkeiten:

- Erweiterung der Netzstruktur so, dass der Konflikt dynamisch nicht mehr auftritt.

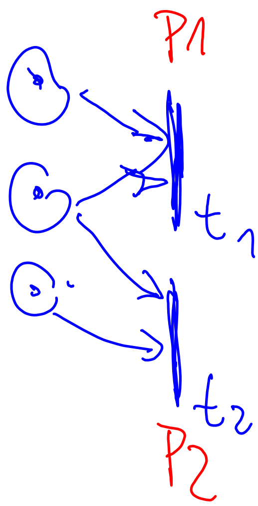
Bsp. PN22 links, Ergebnis im Konfliktfall schalten t1 und t2 immer wechselseitig (Einschränkung des Verhaltens)

- Bedingungen an Kanten (genauer in Kap. 3)



Bed 1 $\wedge$ Bed 2 = 0
(≙ Widerspruchsfreiheit)

- stochastische Bedingung -> stochastische Konfliktlösung
- Bedingungen aus Priorität



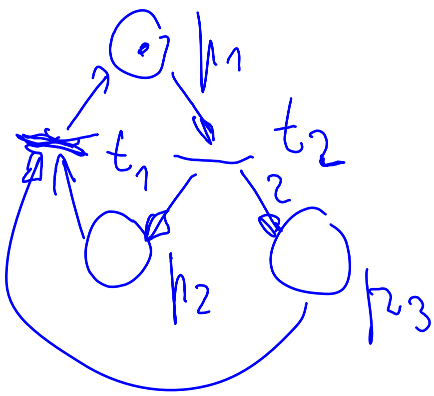
P1 ist höhere Priorität,
d.h. im Konfliktfall schaltet t_1

2.5.4. Beschränktheit und Sicherheit

Fragen:?

- Existiert eine Möglichkeit, benötigte Kapazitäten zu realisieren?
- Reichen die entworfenen Kapazitäten und kann mann evtl. auf den Test auf ausreichende Kapazität zur Laufzeit verzichten?

Def.: Ein PN ist beschränkt, dass für den Fall, wenn alle K(p) auf unendlich gesetzt werden im PN nur eine endliche maximale m(p) in allen p entstehen kann.



$m(p_3) \rightarrow \infty$

PN ist nicht beschränkt

Def.: Ein PN ist kapazitätsbeschränkt (K-beschränkt), dass für den Fall, wenn alle K(p) auf unendlich gesetzt werden im PN in allen erreichbaren m gilt m(P) ist kleiner gleich der urspünglichen K(p).

Ergebnis: in K-beschränkten Netzen ist der Test auf Nachbedingung nicht notwendig.

Def.: Ein PN ist sicher (1-beschränkt), dass für den Fall, wenn alle K(p) auf unendlich gesetzt werden in allen erreichbaren m gilt: m(p) ist kleiner gleich 1.

Bsp PN24 oben sicher

unten (bei K(p5)=1 wird das exklusive Drucken (d.h. in p2 und p4 nicht gleichzeitig eine Marke) realisiert), für K(p5) = unendlich ensteht in p5 aber auch die Markierung 2, d. h. nicht sicher und auch nicht K-beschränkt.

2.6. Spezielle Netzklassen

Pl-T-Netze mit einigen Einschränkungen

2.6.1. Zustandsmaschinen (state machines)

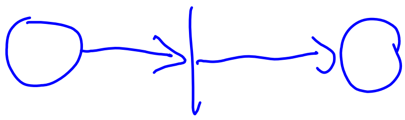
entsprechen dem endlichen dig. Automaten

ZM=(P,T,F,m0)

P: keine Einschränkung gegnüber allgemeinen PN

T: „

F: $\forall t : \exists!\, p \mid (p,t) \in F \wedge \exists!\, p \mid (t,p) \in F$



,d.h. jede Transition hat genau einen Vor- und einen Nachplatz

K und V können entfallen, da K und V immer gleich 1 sind.

m0: $m_0(p_i) = 1 \;;\; m_0(p_j) = 0 \;\forall p_j \mid j = i$

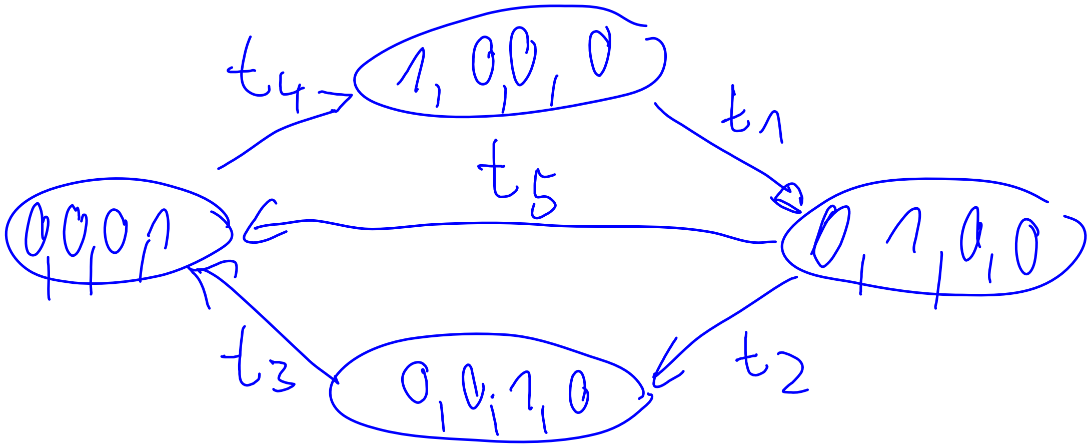
Es gibt genau einen anfangsmarkierten Platz.

Aufgrund der strukturellen Eigenschaften einer ZM gilt die Eigenschaft von m0 auch für alle erreichbaren mi.

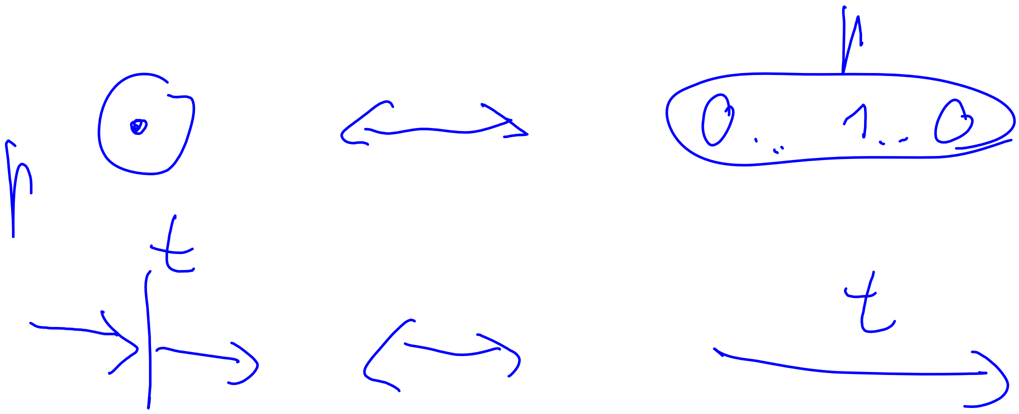
Bsp.: PN25

ZM sind immer sicher.

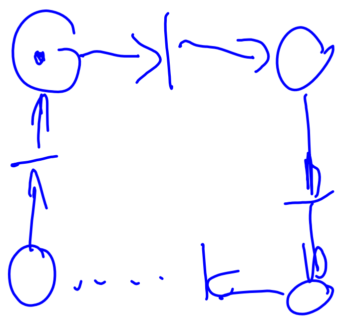
Erreichbarkeitsgraph (Bsp.)

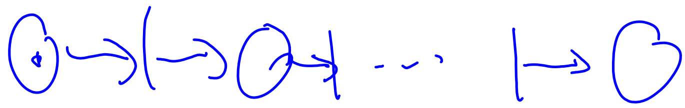


EG und ZM sind „strukturäquivalent“.



ZM können lebendig, schwach lebendig und tot sein.
ZM haben typischerweise Konflikte: diese modellieren die Alternativen, konfliktfrei sind nur der Ring und die Linie:





ZM modellieren sequentielle Algorithmen (ohne Parallelität).

2.6.2. Systeme von Zustandsmaschinen (SZM, buffered state machines)

SZM: Menge von ZM und ein Koppelnetz KN

KN=(P,T,F,K,V,m0)
T: alle t sind in den ZM
keine weitern Einschränkungen
Bsp. PN26

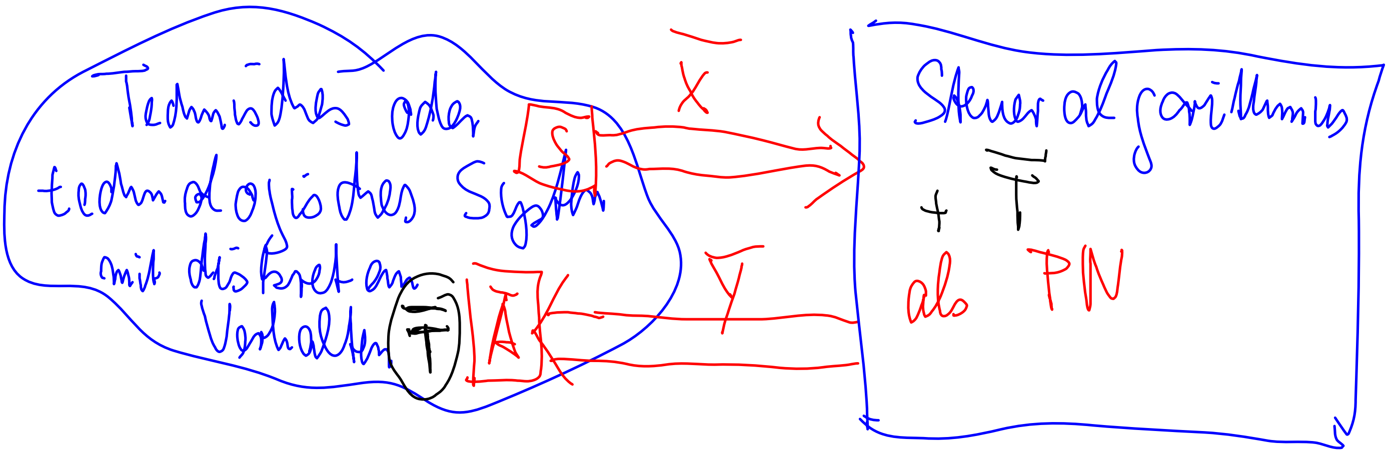
KN führt zur Lösung der sonst (ohne KN) in den ZM entstehenden Konflikten, wobei eine ZM wechselseitig auf den Konflikt der anderen ZM einwirkt.

SZM modellieren parallele sequentielle Algorithmen, die Abhängigkeiten untereinander haben.

- ➔ Softwaremodelle dazu in Kap. 6

3. Steuerungsentwurf

Problem:



S: Sensoren

A: Aktoren

$\overline{T}$: Zeitwerte

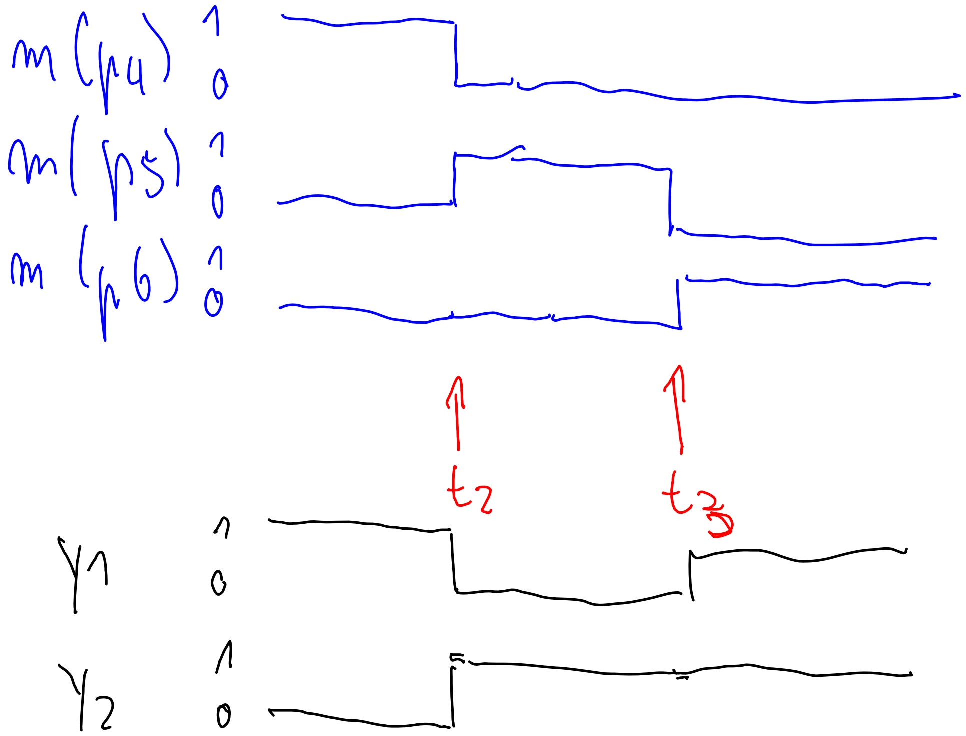
$\overline{X}$: Binäre Eingangsvariablen

$\overline{Y}$: Binäre Ausgangsvariablen

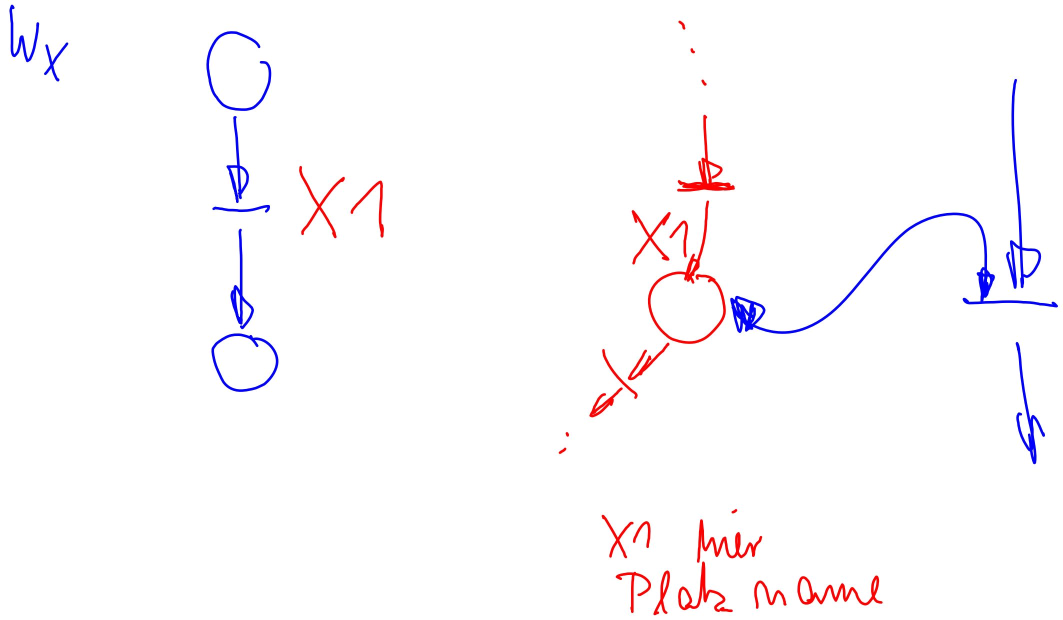
Notwendig: Erweiterungen für Berücksichtigung von $\overline{X}$ und $\overline{T}$ und Ausgabe von $\overline{Y}$

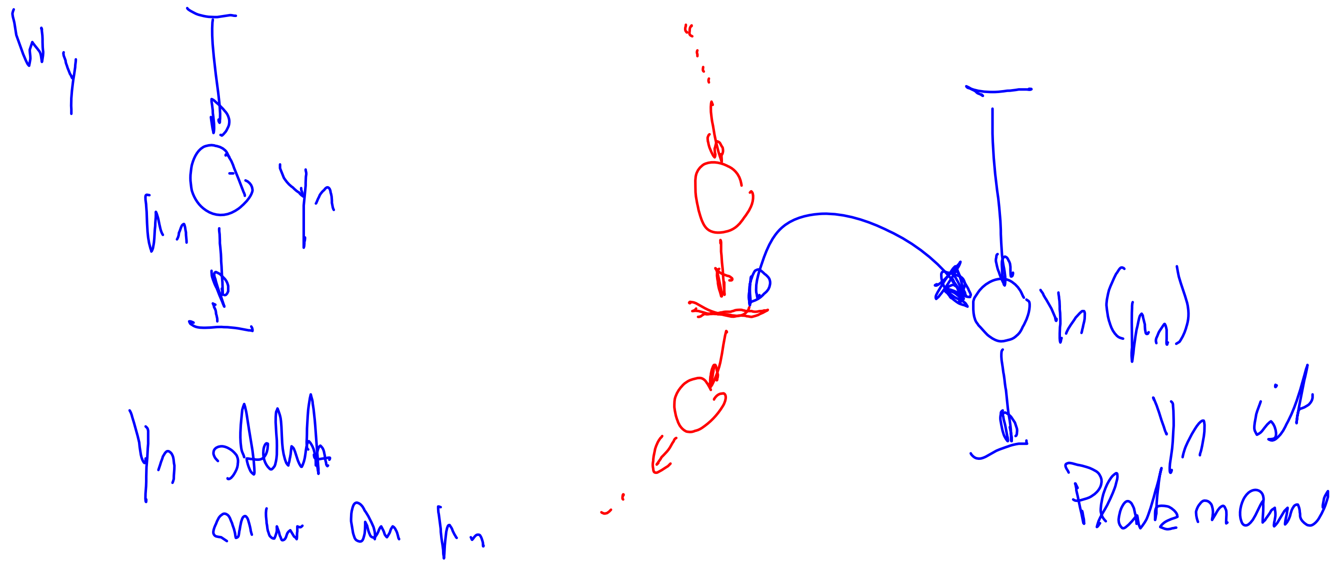
Def. Bewertungsfunktion wx:
Menge von Booleschen Ausdrücken -> T:
Wahrheitwert des Ausdruckens ist zusätzlich notwndige Schalbedingung (=1).
Bsp. PN 27 oben:
t1 ist sf für m(p1)=m(p2)=1 und m(p3)=0 und X1=1 und X2=0
Das Schalten einer t ändert den Wert der Variablen im zugeordneten Ausdruck nicht direkt.

Def. Ausgabefunktion wy:
Menge von Aufzählungen von Booleschen Variablen -> P
Steht eine Variable an mindestens einem Platz mit m(p)=1 ist ihr Wert=1 und 0 sonst.
Bsp. PN27 unten:
Zeitverlauf:



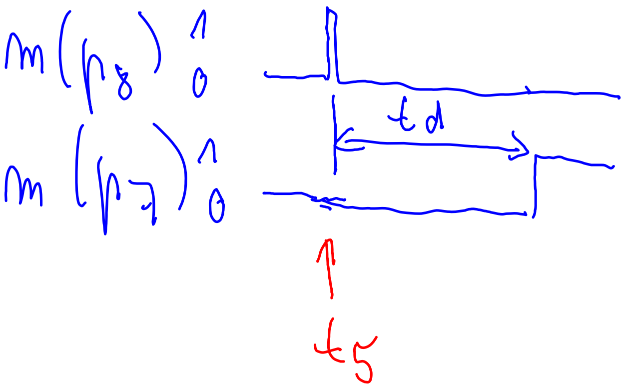
Falls ein PN-Modell vom Prozess existiert, kann wx und wy auch über Sonderkanten modelliert werden:
Bsp.





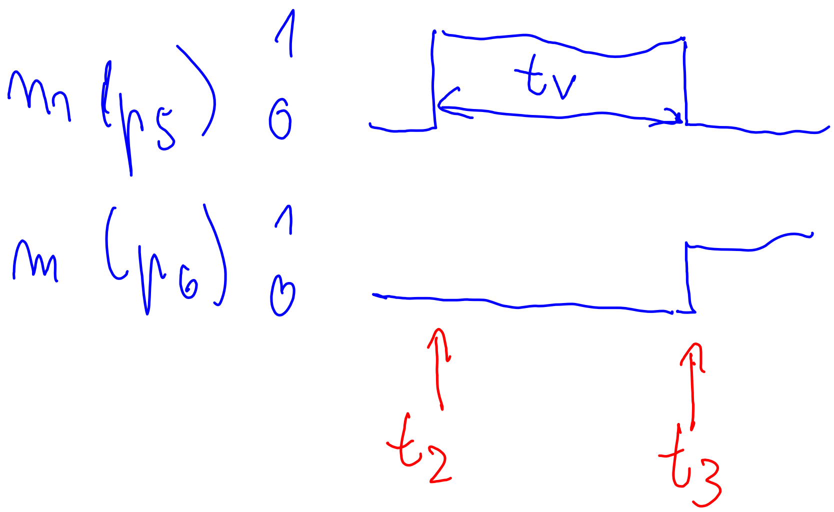
Zeitbewertunsfunktionen wt
wtv, wtd, wtvi, wtdi
v: Verzögerung
d: Dauer
i: Intervall

Def. wtd: T -> Menge von Zeitwerten (Festwerte):
t mit wt(t)=tdi schaltet ihre VB (Vorbedingung) bei sf sofort, die Nachbedingung nach tdi, falls noch möglich, sonst zum nächst möglichen Zeitpunkt nach tdi.
Bsp. PN 28 oben:
Zeitverlauf



wtv (Zeitverzögerungsfunktion)
Abbildung von T in eine Menge von Zeitverzögerungswerten (Festwerte): Eine t muss die angegbene Zeitverzögerung ununterbrochen schaltfähig sein, um schalten zu können. Dann schaltet sie unmittelbar. (unendlich kurz)

Zeitverlauf:

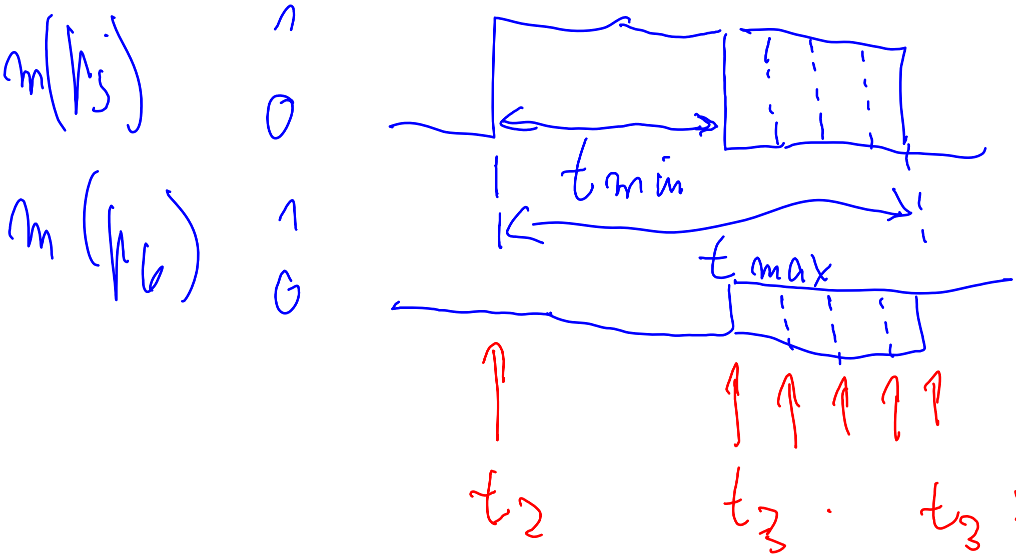


Evtl. ist das Zeitverhalten einer t nicht genau bekannt bzw. es ist variabel. Dann sind eventuell Intervalle sinnvoll:

(tmin, tmax)
tmin. ist der kleinste mögliche Zeitwert
tmax ist der größte mögliche Zeitwert

Bsp. tvi (Zeitverzögerungsfunktion mit Intervall)

tv=(tmin, tmax)

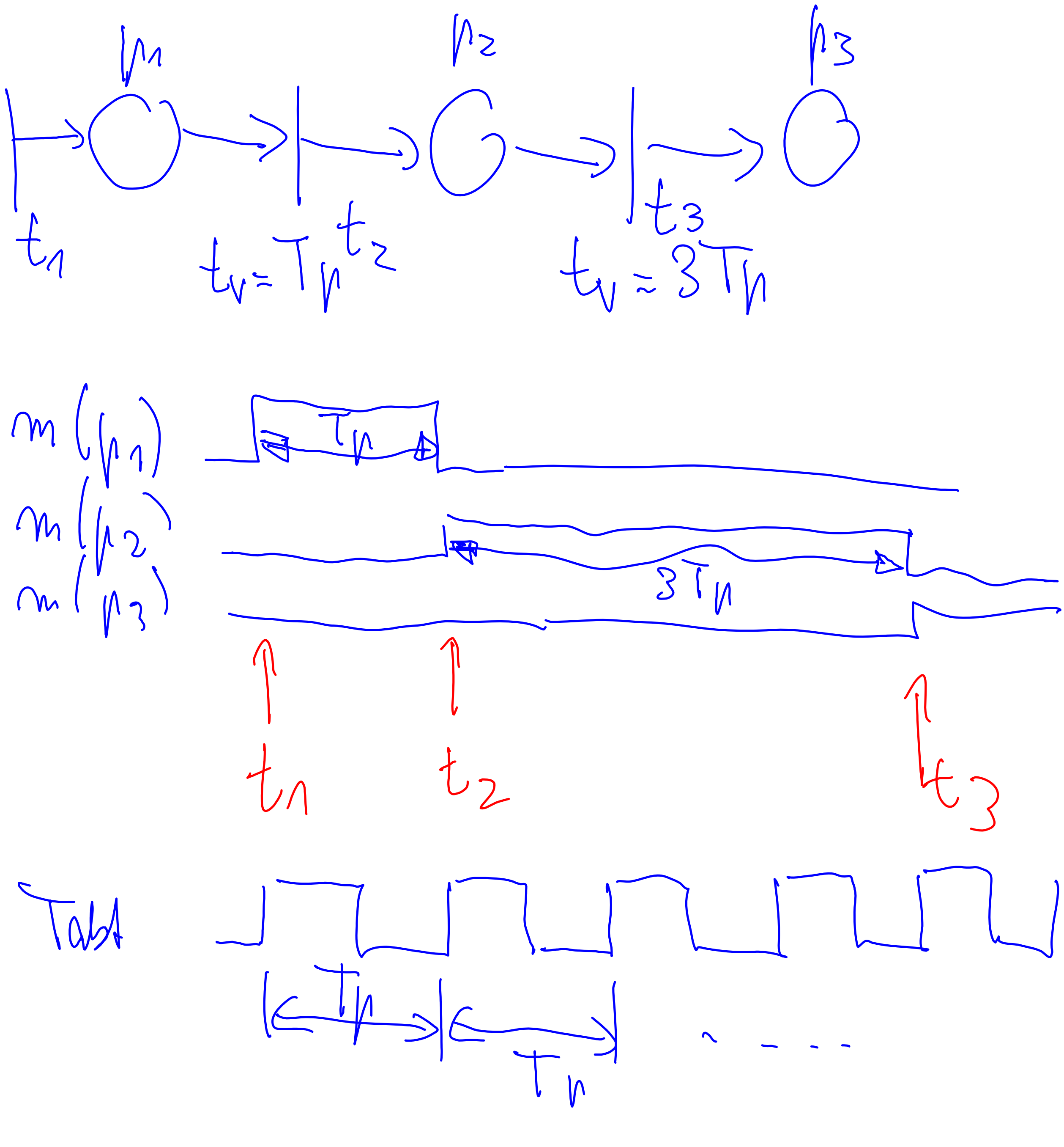


t_3: im konkreten Fall nur zu einem Zeitpunkt

Auswahl des tatsächlichen Zeitwertes im Intervall z.B. zufällig.

Spezialfälle:
tmin=tmax=0 : maximale Schaltregel
tmin=0, tmax= ∞ : stochastisches Schalten
tv für alle Transitionen ist Zeitwert Tp bzw. posives ganzahliges Vielfaches davon : getaktetes Netz

Alle t schalten immer am Ende von Tp:



Dauertransitionen lassen sich auf Verzögerungstransitionen zurück führen: (Pn 29)