

Für über den Prüfungsstoff interessierte:

Grundprinzip zur softwaretechnischen Implementierung von PN

1. Erzeugen eine nichtgraphische Darstellung mit allen relevanten Informationen (graphische Informationen sind nicht notwendig)

Platzliste:

je Platz:

- symbolischer Bezeichner, optional (z.B. für Debugfunktion)
- Zeiger auf Variable (unsigned integer): Physische Realisierung des Platzes mit seiner Markierung
- Konstante für Anfangsmarkierung
- Konstante für die Kapazität
- Zeiger auf evtl. wy-Funktion
- Zeiger auf evtl. wa-Funktion (siehe Kap. 6, Softwareaktion)

Transitionsliste:

je Transition:

- symbolischer Bezeichner (optional)
- Zeiger für evtl. wx-Funktion
- Zeiger für evtl. wt-Funktion
- Zeiger für evtl. wa-Funktion (siehe Kap. 6)
- Konfliktlösungsangaben (z.B. Priorität)

n mal (für alle Vorplätze,  $0 \leq n \leq$  endliche nat. Zahl)

- Zeiger in die Platzliste auf entsprechenden Vorplatz
- Konstante für die Vielfachheit
- Konstante für Sondereigenschaft der Kante (z.B. Testkante)

m mal (für alle Nachplätze,  $0 \leq m \leq$  endl. nat. Zahl)

- wie bei den Vorplätzen

2. Lauffähiges Programm



+ Erzeugen  $m_0$

+  $w_x$ -Variable einlesen

+ 1. Transition ( $t$ )

+  $t$  sf? ( $VB, NB, w_x, w_t$ )

nächste  
 $t$

+ 1.  $t$   
Kernz. sf? setzen  
Kernz. sf?

+ letzte  $t$ ?

Konfliktlösung  $\rightarrow$  KZ sf  
wird erst zurück  
gesetzt

nächste  
 $t$

+ Kernz. sf?

+  $t$  schalten ( $VB, NB, w_y$   
erk. wa stärken)

+ letzte  $t$ ?

+  $w_y$  ausgeben

+ Abbruch?  
 $m$



➔ Testen des generierten Programms:

- Anschluss an den Prozess über wx (Sensoren, über EA-Schnittstelle) und über wy (Aktoren, über EA-Schnittstelle)
- Betriebsarten für Test:

Anzeigen: aktuelle m, wx, wy

1\* Durchlauf für alle t, evtl. geteilt für Test sf und Schalten

Lauf bis angegebene mi oder Teilmarkierung mit evtl. Trace der Zwischenmarkierung

Lauf bis tote m

...