

Systematischer Entwurf eines dezentralen Multitasking-Systems zur Steuerung eines räumlich verteilten Prozesses

Eberhard Kienzle

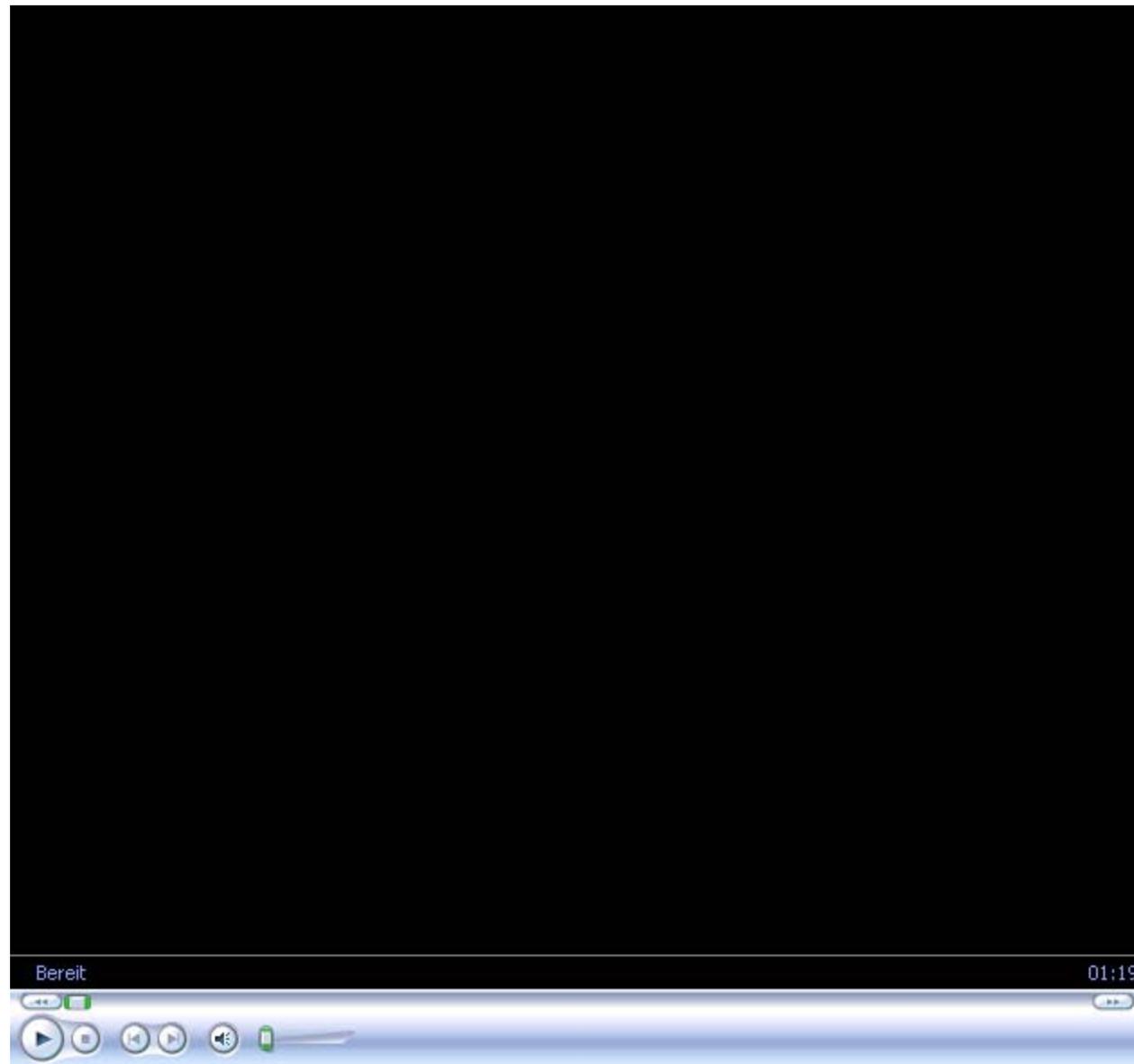
FHT Esslingen

Gliederung

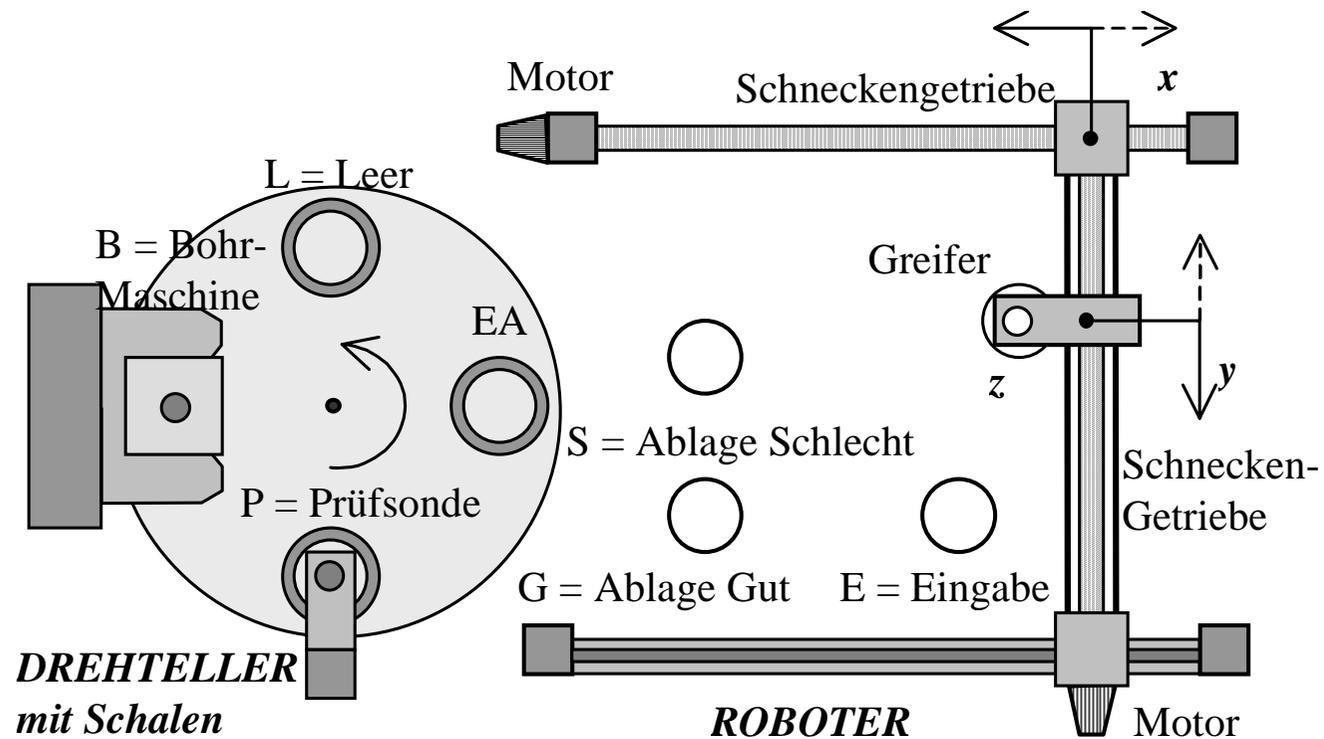
1. Vorstellung des Prozesses, Teil-Prozeß
2. Steuerfunktionen und Prozeßfluß-Graph PFG
3. Multitasking-System, vorläufige Version
4. Steuerbedingungen und Datenfluß-Graph DFG
5. Multitasking-System, ausführbare Version
6. Übertragung auf vollständigen Prozeß

Der Prozess in Aktion

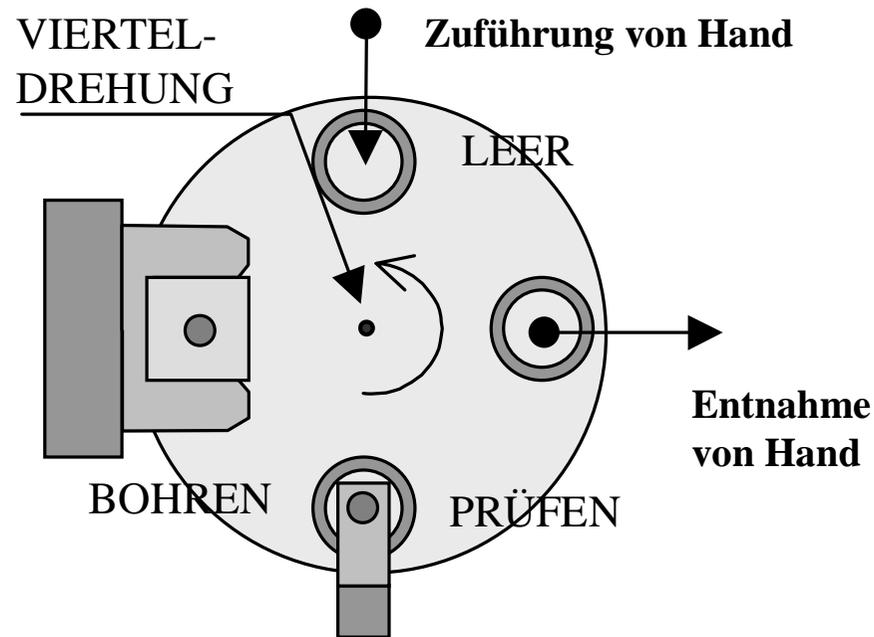
Eberhard Kienzle



Prozess „Modellanlage zur Produktion von Plastikteilen“



Teil-Prozess **Drehteller** mit Bearbeitungsmaschinen



Gliederung

1. Vorstellung des Prozesses, Teil-Prozeß
- 2. Steuerfunktionen und Prozeßfluß-Graph PFG**
3. Multitasking-System, vorläufige Version
4. Steuerbedingungen und Datenfluß-Graph DFG
5. Multitasking-System, ausführbare Version
6. Übertragung auf vollständigen Prozeß

Maschinen-Aktionen werden durch **Funktionen** gesteuert:

- **Viertel-Drehung:** "DREHEN"
- **Bohren:** "BOHREN"
- **Prüfen:** "PRÜFEN"

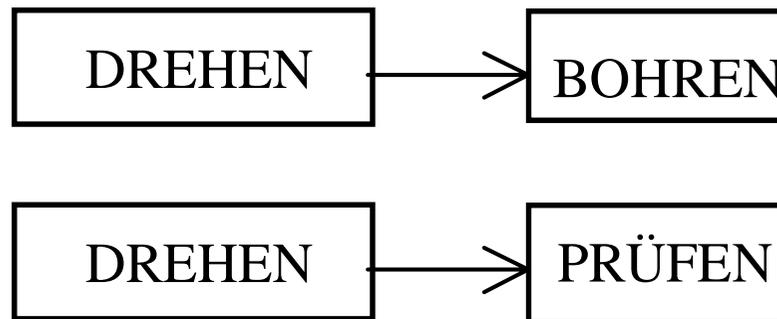
Steuerfunktion = Prozess-Ausgabe **mit Warten** (auf Interrupt)

Eingabe und **Entnahme von Hand** sind entkoppelt

Maschinen-Aktionen bzw. Funktionen {A, B}, die paarweise **kollidieren** können, müssen **sequentiell** ablaufen:

$$A \rightarrow B (\rightarrow A)$$

im Beispiel gibt es folgende **Kollisions-Paare**:

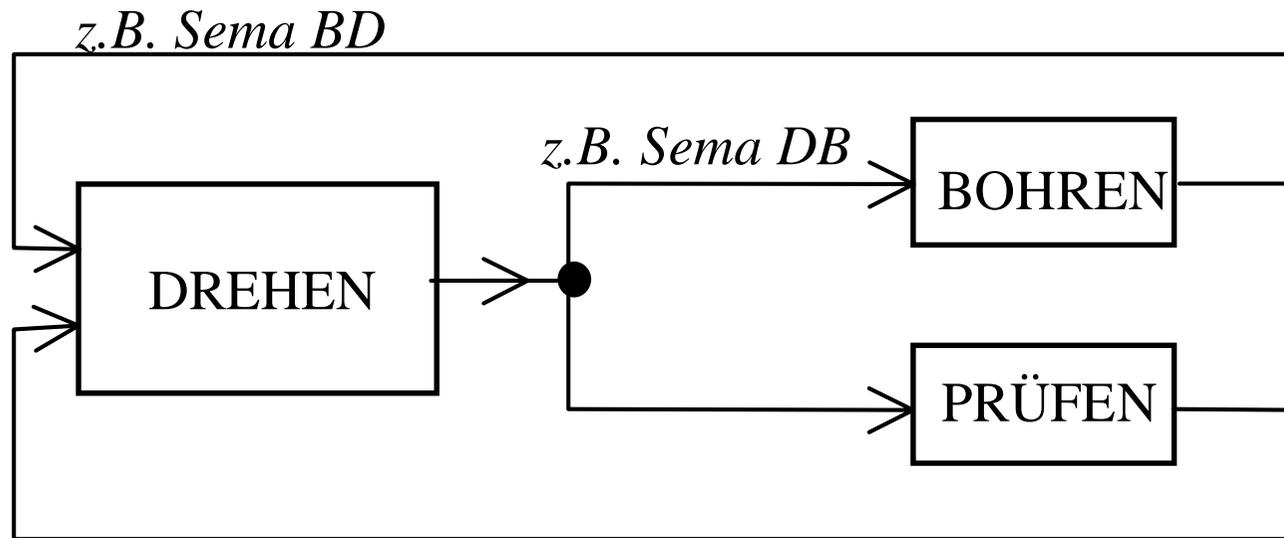


Funktions-Paar {BOHREN, PRÜFEN} kann **simultan** ablaufen

Die Zusammenfassung aller derartigen Relationen
ergibt den **Prozeßfluß-Graphen PFG**

Der PFG zeigt den möglichen Grad an **Simultan-
Arbeit**

Prozessflussgraph PFG

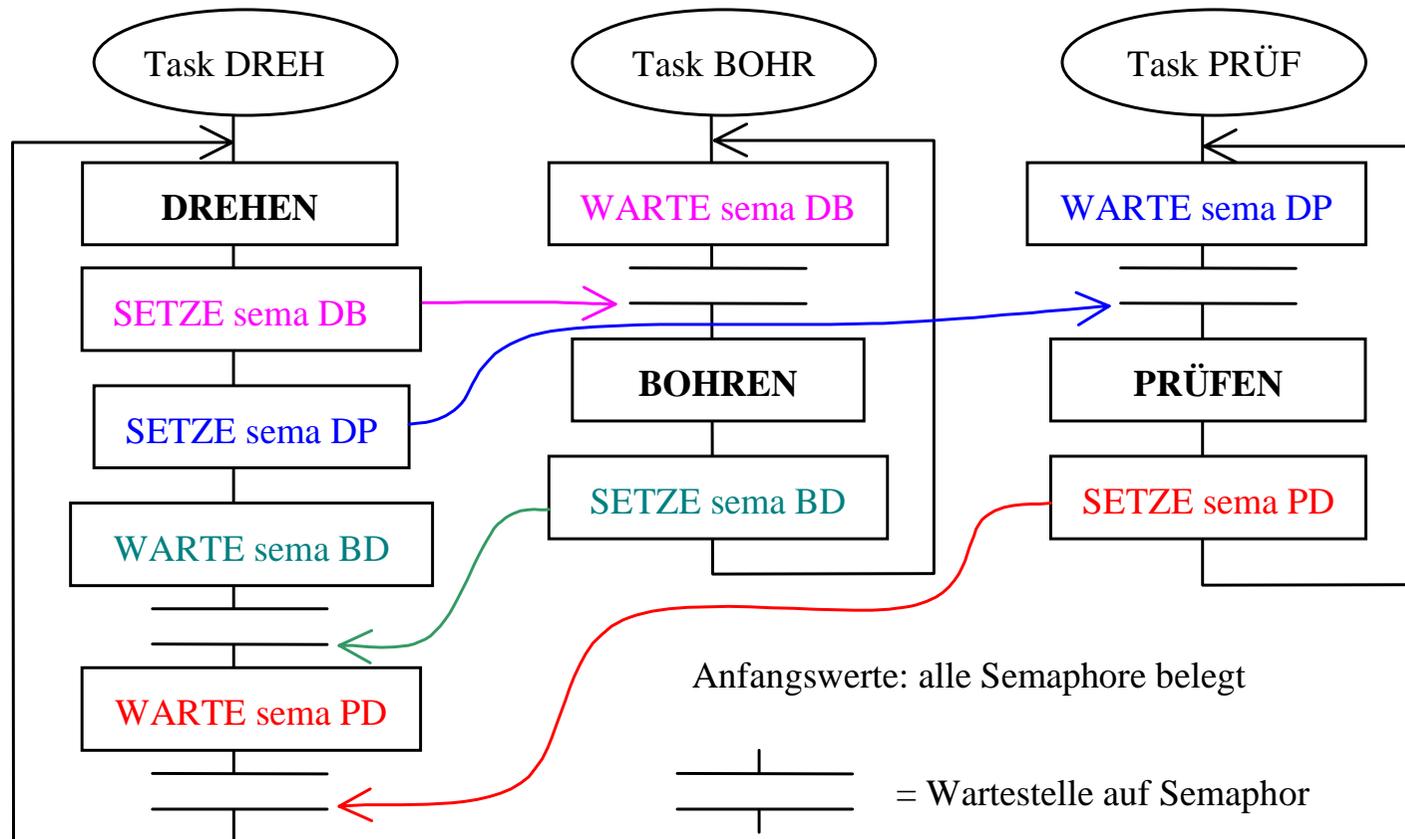


DREHEN muss auf BOHREN **UND** PRÜFEN warten

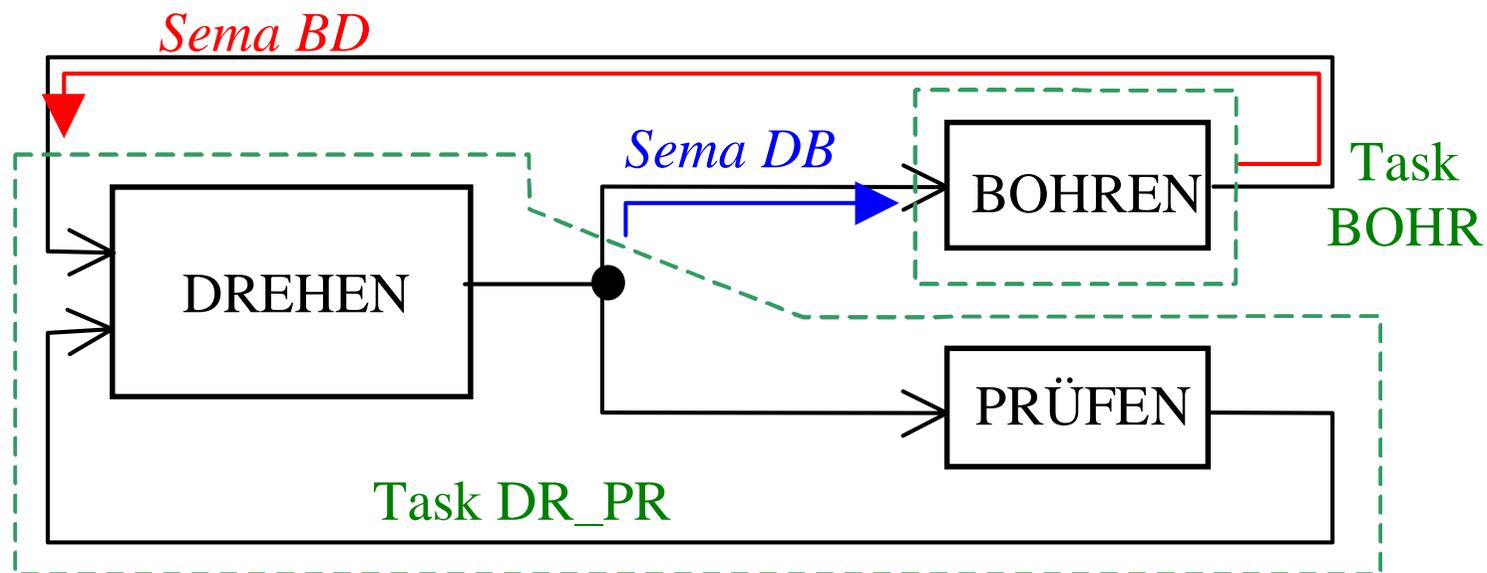
Gliederung

1. Vorstellung des Prozesses, Teil-Prozeß
2. Steuerfunktionen und Prozeßfluß-Graph PFG
- 3. Multitasking-System, vorläufige Version**
4. Steuerbedingungen und Datenfluß-Graph DFG
5. Multitasking-System, ausführbare Version
6. Übertragung auf vollständigen Prozeß

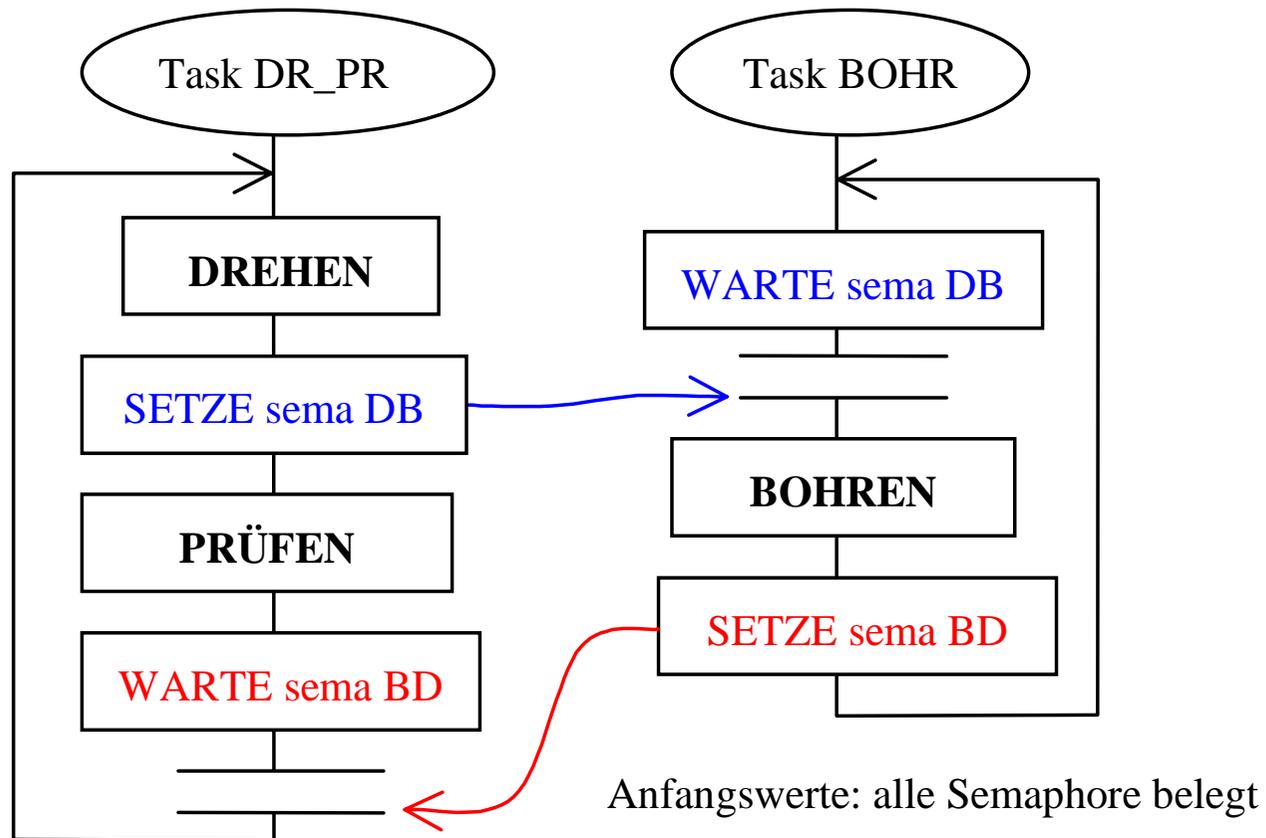
Abbildung PFG in vorläufiges Tasksystem



Zusammenfassung sequentieller Funktionen zu **einer** Task



Äquivalentes Tasksystem



Gliederung

1. Vorstellung des Prozesses, Teil-Prozeß
2. Steuerfunktionen und Prozeßfluß-Graph PFG
3. Multitasking-System, vorläufige Version
- 4. Steuerbedingungen und Datenfluß-Graph DFG**
5. Multitasking-System, ausführbare Version
6. Übertragung auf vollständigen Prozeß

Der Prozeßfluß-Graph regelt den **kollisionsfreien** und maximal **simultanen** Ablauf der Steuerfunktionen.

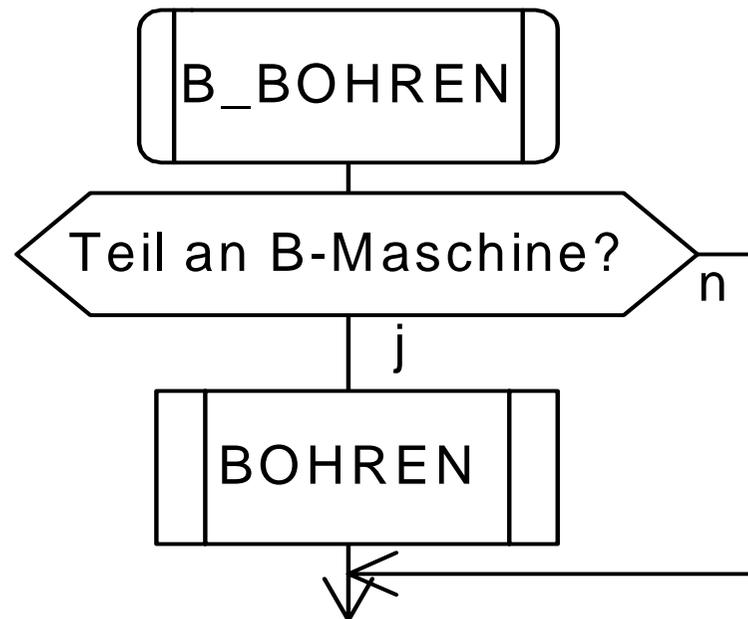
Der **zeitliche Verlauf** der Funktionen wird bestimmt vom Prozeßzustand, d. h. von der **aktuellen Position** jedes Teils

Beispiel 1: **Aufruf der Funktion BOHREN:**

gebohrt wird nur **dann, wenn** ein "**Teil an der Bohrmaschine**" liegt

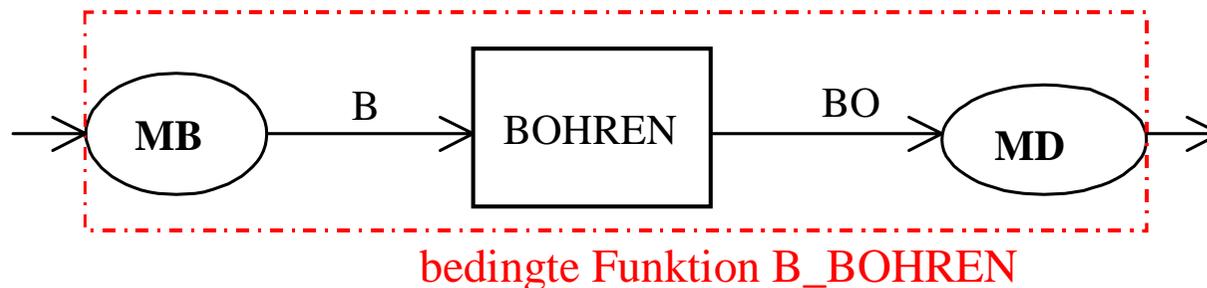
B = {"Teil an Bohrmaschine"} ist eine **Steuerbedingung**,
Werte {true, false}.

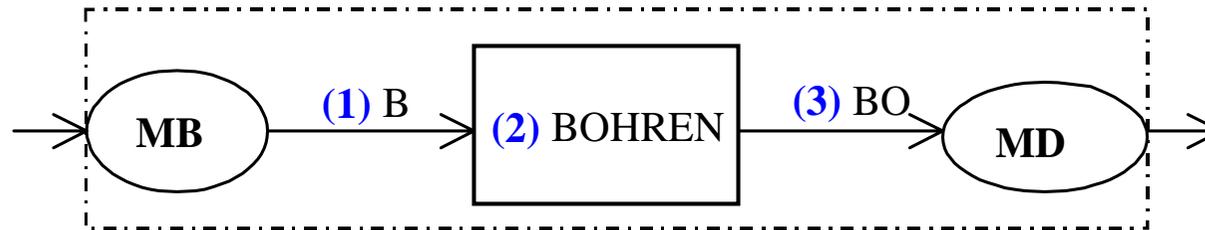
Durch die Verwendung der
Bedingung B wird die
Funktion BOHREN
erweitert zur **bedingten**
Steuerfunktion
B_BOHREN



Bedingung B wird - parallel zum **realen Teil-Transport** - von der Vorgänger-Funktion B_DREHEN in die globale **Mailbox MB** von B_BOHREN geschrieben („Send“)

B_BOHREN und Fluss der Bedingungen, **Datenfluss-Graph**:





(1) B_BOHREN prüft B und löscht B danach („RECEIVE“)

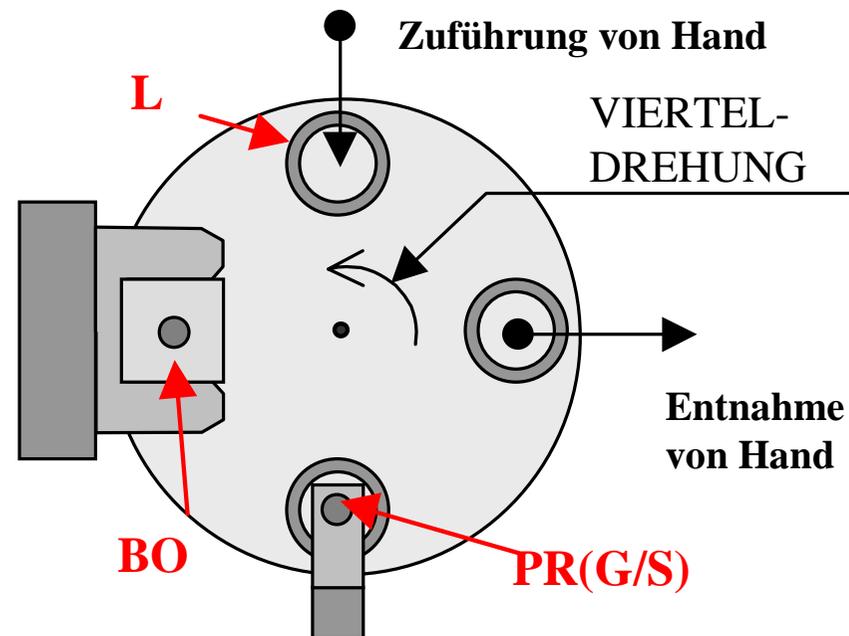
Falls B = true:

Aufruf BOHREN **(2)**, dann **Folgebedingung BO** (Teil gebohrt!)
in **Mailbox MD** der **Folgefunktion B_DREHEN** schreiben **(3)**

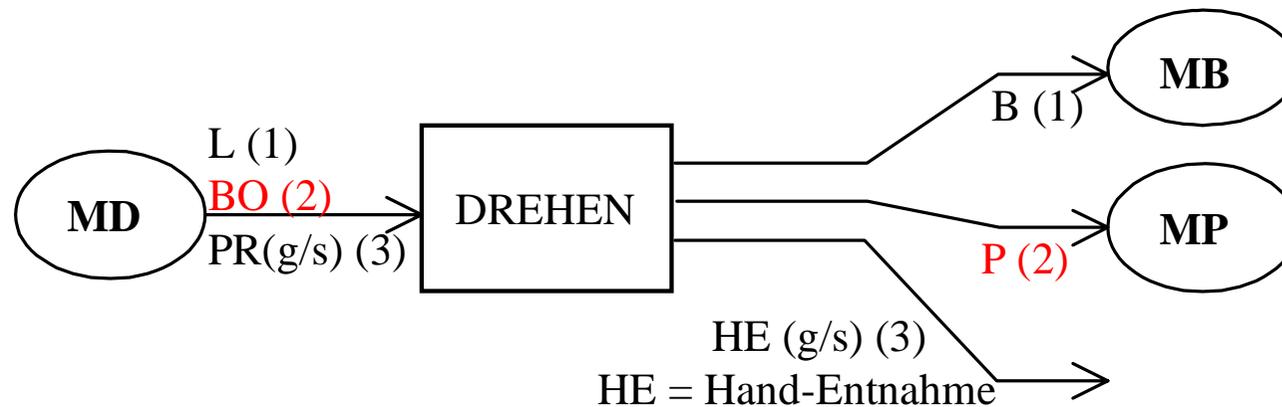
Andernfalls BOHREN und Schreiben von BO überspringen

Beispiel 2: Der Transportschritt **Viertel-Drehung**, Funktion "DREHEN", hat **3** auslösende **Bedingungen**

1. Teil in **L**EER-Position
2. Teil ge**B**Ohrt an Bohrmaschine
3. Teil ge**P**Rüft an Prüfstift (Ergebnis **G**ut/**S**chlecht)



Datenfluss der bedingten Steuerfunktion B_DREHEN :



Auch wenn ≥ 1 **Bedingungen erfüllt**, gibt es nur **1 Drehung**,
aber alle zugeordneten Folgebedingungen werden erzeugt

Umsetzung **Datenfluss** → **Pseudo-Code** von B_DREHEN,
Bedingungen **BO, P**:

```
MDreh := 0           // Merker: MDreh = „Drehung notwendig“
.....

if empfangen_mail (MD) = BO // Teil gebohrt? Falls ja
    MDreh := 1           // Drehung notwendig
    send_mail(MB, B)     // erzeuge Bedingung P = „Teil an PS“
endif

.....

if MDreh = 1         // falls Drehung notwendig
    DREHEN           // Drehen, wenn ≥1 Bedingung erfüllt
endif.
```

Gliederung

1. Vorstellung des Prozesses, Teil-Prozeß
2. Steuerfunktionen und Prozeßfluß-Graph PFG
3. Multitasking-System, vorläufige Version
4. Steuerbedingungen und Datenfluß-Graph DFG
- 5. Multitasking-System, ausführbare Version**
6. Übertragung auf vollständigen Prozeß

Vorgehen:

1. **Datenfluss** und **Pseudo-Code** sind äquivalent
2. Den kompletten **Datenfluss-Graphen DFG** gewinnt man direkt aus dem **Weg eines Teils durch die Anlage**
3. Der **(Pseudo-)Code** einer bedingten Steuerfunktion wird aus dem ihr zugehörigen **Teil des DFG** ermittelt, siehe die vorigen Beispiele

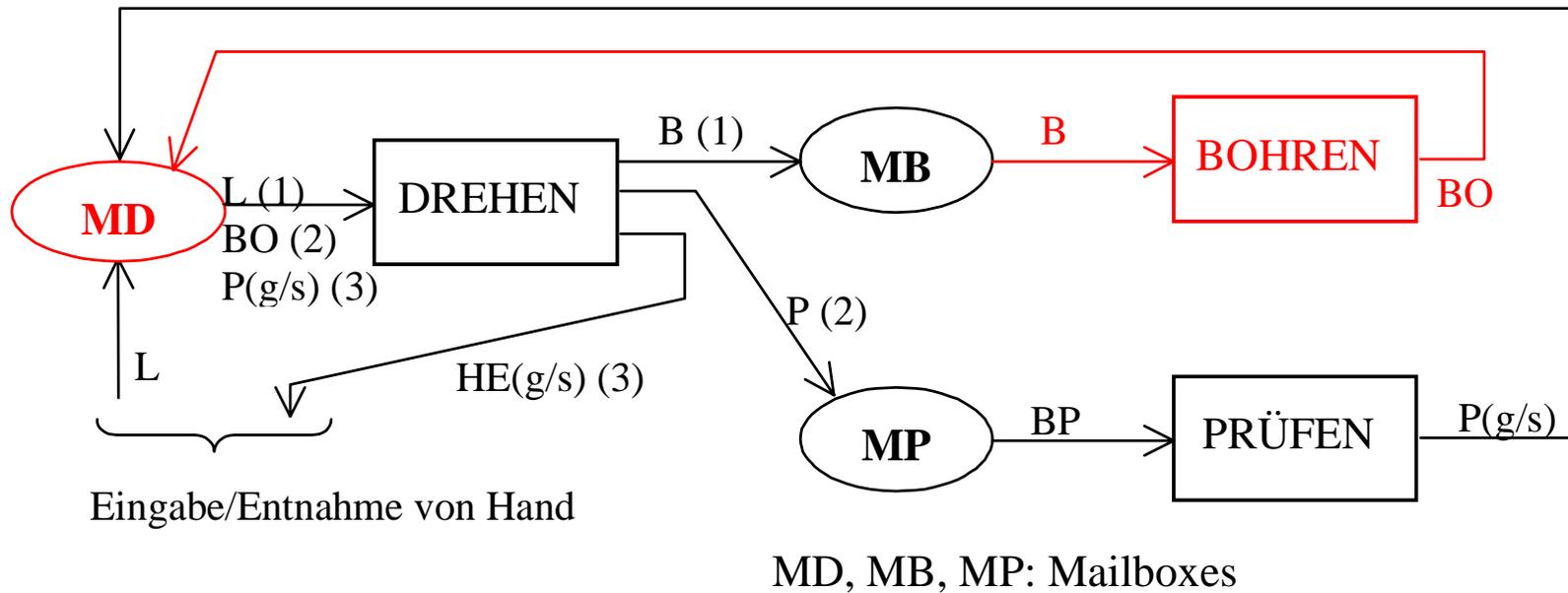
Weg eines Teils durch die Anlage, Ausschnitt

Rote Zeile = Bearbeitungs-Schritt **B_BOHREN**

B prüfen, BOHREN, Bedingung BO erzeugen

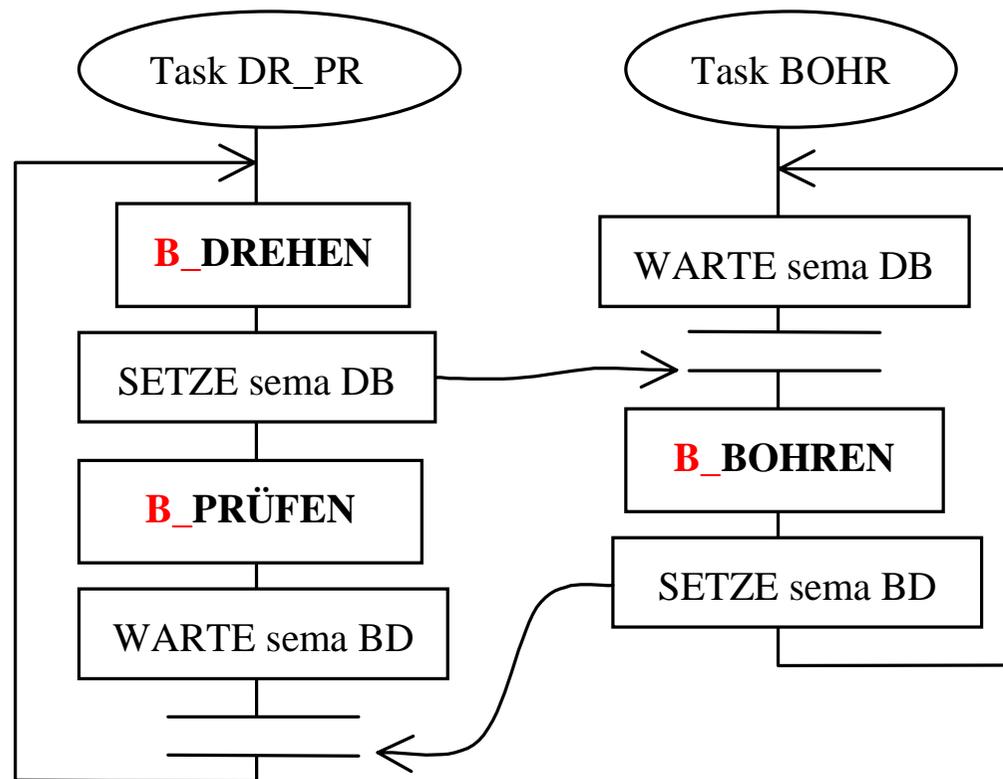
Bedingung	Unbedingte Steuerfunktion	Erzeugte Bedingung	Mailbox
.....
L = Teil in Schale Leer	DREHEN	B = Teil an Bohrmaschine	MB
B = Teil an BM	BOHREN	BO = Teil gebohrt	MD
BO = Teil gebohrt	DREHEN	P = Teil an Prüfstift	MP
.....

Vollständiger Datenfluss-Graph DFG, rot: Schritt BOHREN



Simultane Teile-Bearbeitung → simultaner Fluss von Bedingungen

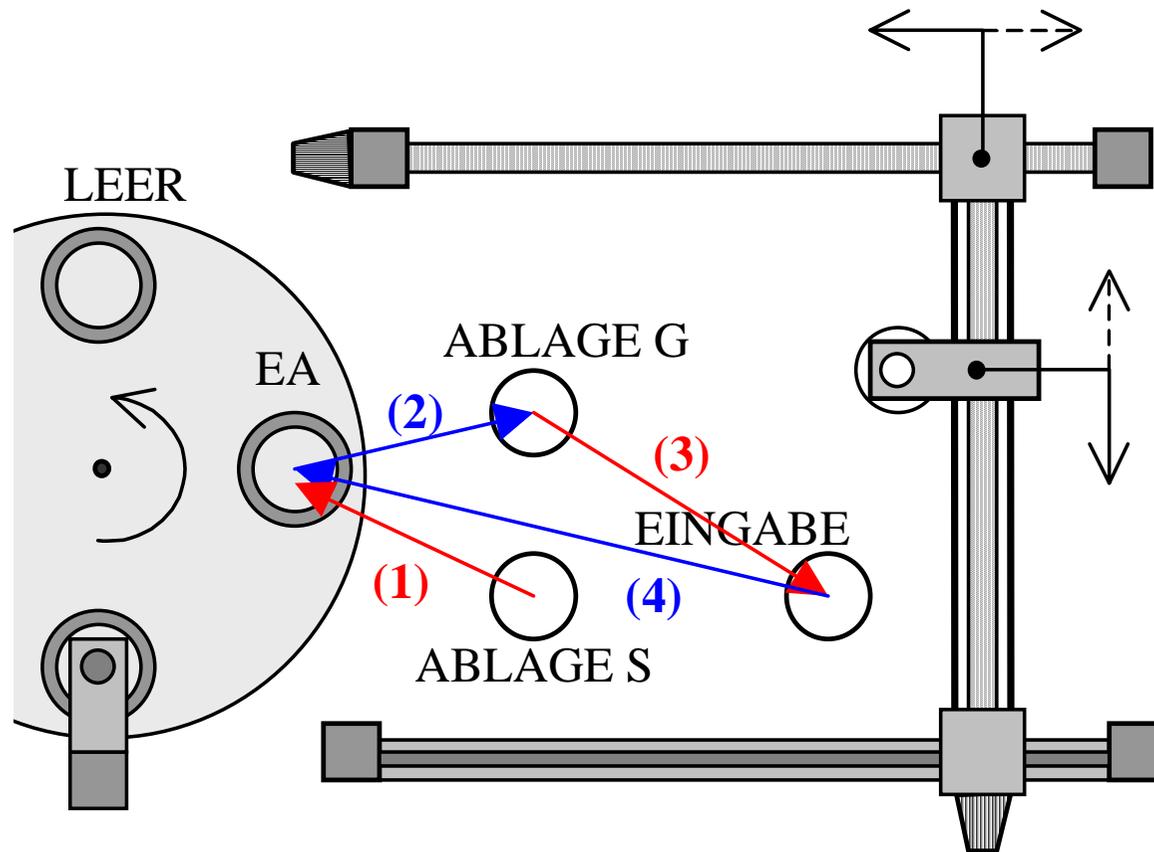
Ersetzt man im
vorläufigen
Multitasking-System die
unbedingten
Steuerfunktionen durch
die **bedingten**
Funktionen (Code!), so
erhält man das
ausführbare
Multitasking-System



Gliederung

1. Vorstellung des Prozesses, Teil-Prozeß
2. Steuerfunktionen und Prozeßfluß-Graph PFG
3. Multitasking-System, vorläufige Version
4. Steuerbedingungen und Datenfluß-Graph DFG
5. Multitasking-System, ausführbare Version
6. Übertragung auf vollständigen Prozeß

Allgemeine Anlage, längster Roboter-Fahrweg 1 -2 -3 - 4



Neue und alte Kollisions-Paare der vollständigen Anlage:

FAHRT_NKOL → FAHRT_KOLL

DREHEN → FAHRT_KOLL

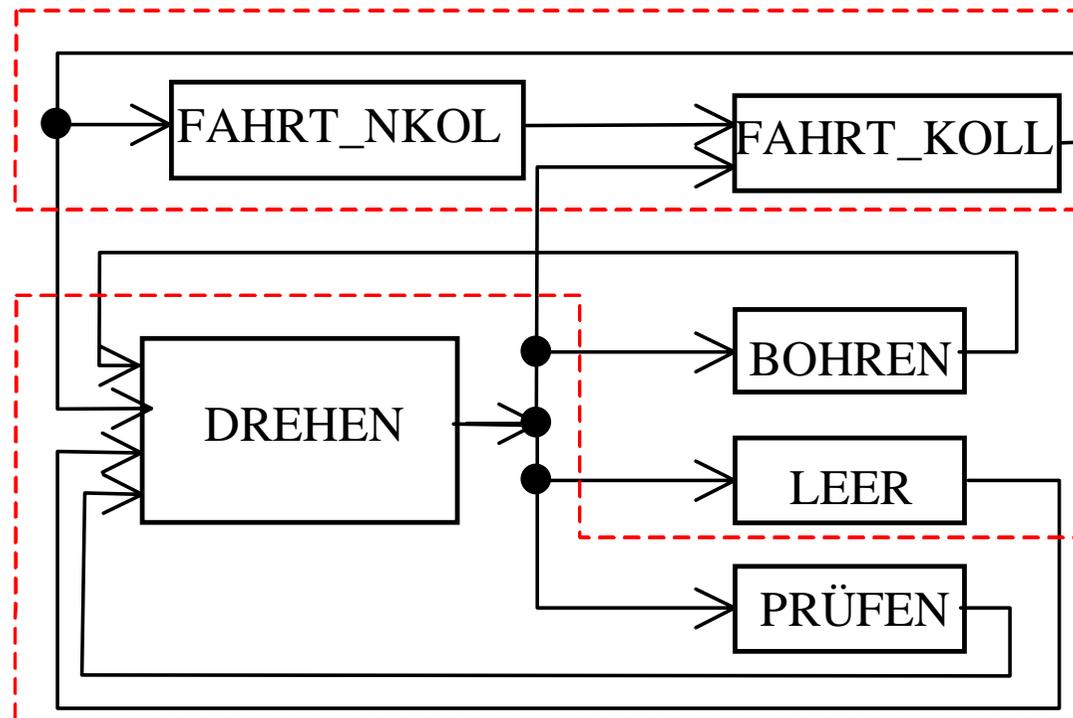
DREHEN → LEER

DREHEN → BOHREN

DREHEN → PRÜFEN

FAHRT_NKOL, FAHRT_KOLL = **kollisions-freier** bzw. mit
DREHEN kollidierender Teil der Roboter-FAHRT

Durch Zusammenfassung ergibt sich systematisch der Prozessfluss-Graph, daraus - nach Kombination sequentieller Funktionen - ein kooperierendes System von vier Tasks.



Der komplette **Datenfluss-Graph DFG** ergibt sich aus dem Weg eines Teils durch die Anlage

