



## Beschränktheit und Sicherheit

Ein Ablaufdiagramm wird als beschränkt bezeichnet, wenn in keiner einzigen Stelle des Netzes jemals mehr als eine bestimmte maximale Anzahl von Marken zusammentreffen. Ist die maximale Anzahl von Marken pro Schritt gleich 1, so wird das Netz als sicher bezeichnet. Wird ein Ablaufdiagramm zur Modellierung oder Programmierung von Steuerungen verwendet, so ist meistens nötig, dass das Netz sicher ist. (Die Schritte werden dann nur als ‚markiert‘ oder ‚nicht markiert‘ bezeichnet.)

Überlegen Sie sich mögliche Gründe!

Die Beschränktheit eines Ablaufdiagramms kann anhand der Eigenschaft ‚Ueberdeckbarkeit‘ untersucht werden.

## Ueberdeckbarkeit einer Markierung:

Beim Kompilieren eines Ablaufprogrammes können Sie mit der Fehlermeldung konfrontiert werden, dass Ihr Ablaufdiagramm überdeckbar ist. Was bedeutet dies und wie können Sie das Problem lösen? Die Ueberdeckbarkeit ist eine Eigenschaft der möglichen Markierungen eines Ablaufdiagramms. Gibt es bei der Simulation eines Ablaufdiagramms zwei unterschiedliche Markierungen M1 und M2, so überdeckt die Markierung M1 die Markierung M2, wenn alle in M2 markierten Schritte auch in M1 markiert sind. Damit dies Sinn macht, muss M2 natürlich mindestens einen markierten Schritt enthalten.

Beispiel:

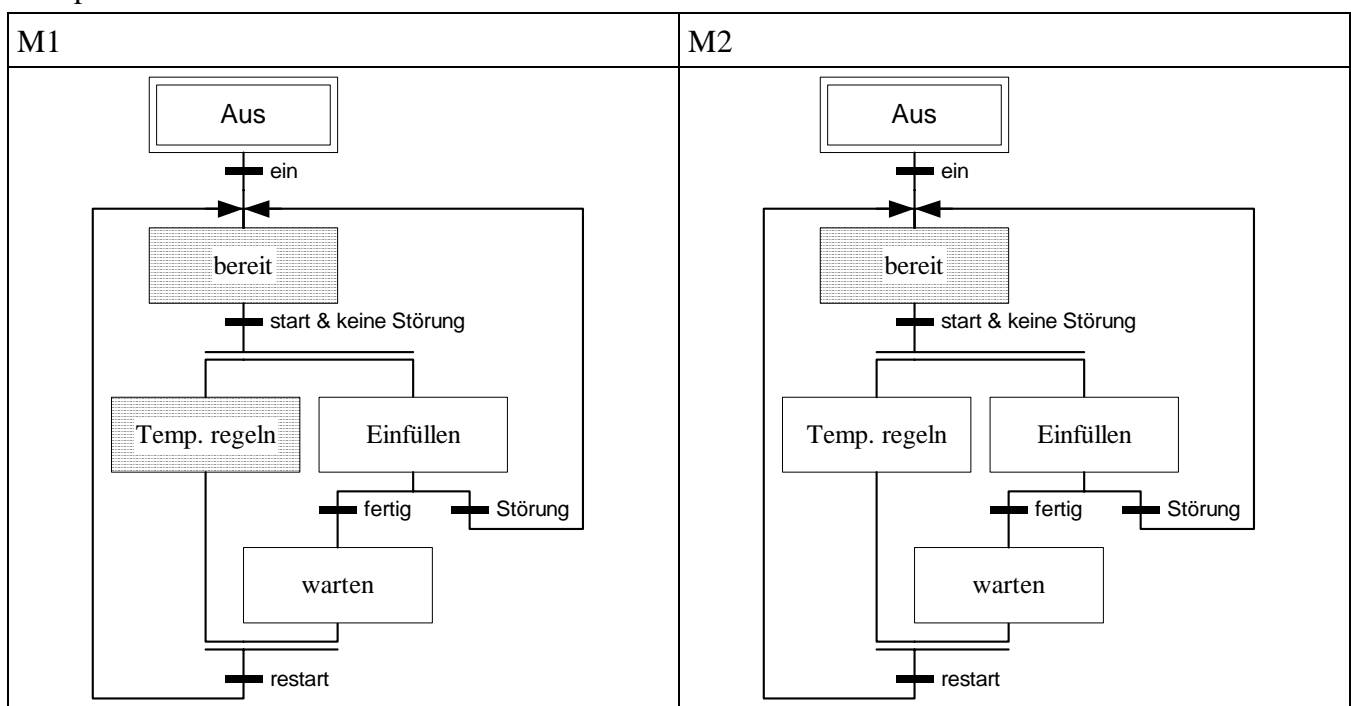
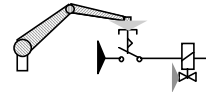


Abbildung 1: Ueberdeckende Markierungen

Im obigen Beispiel überdeckt die Markierung M1 die Markierung M2, weil

- die Markierungen M1 und M2 unterschiedlich sind
- alle markierten Schritte von M2, d.h. hier der Schritt ‚bereit‘ auch in M1 markiert ist

Die Markierung M1 entsteht aus der Markierung M2, wenn die Transitionen 'start & keine Störung' und 'Störung' schalten, d.h. wenn es nach dem Start des Einfüllens eine Störung gibt.



Sobald es bei der Simulation eines Ablaufdiagramms eine überdeckte Markierung gibt, so ist das Ablaufdiagramm nicht sicher. Analysieren Sie dazu das obige Beispiel. Obwohl es aus Einfachheitsgründen prozessmässig nicht sehr sinnvoll ist, so zeigt es, wie überdeckte Markierungen zustande kommen. Ist dies erkannt, so lassen sich auch sehr einfach nötige Korrekturen des Ablaufdiagramms ableiten. Im folgenden Ablaufdiagramm ist eine Verbesserung vorgeschlagen.

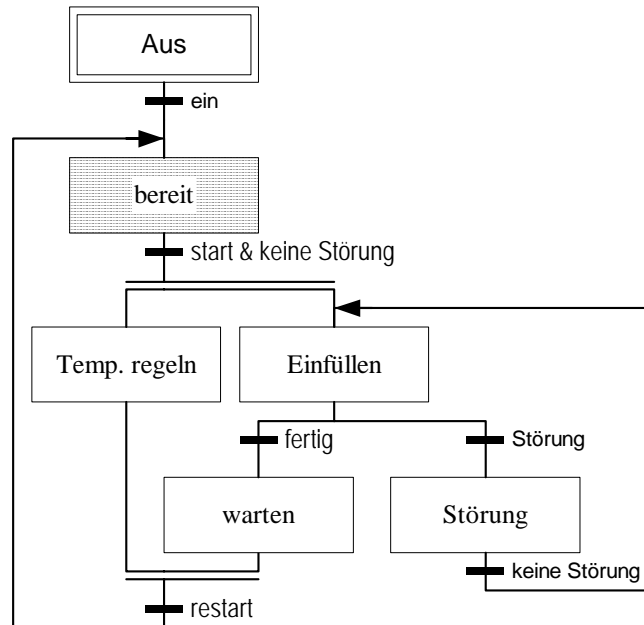


Abbildung 2: Ablauf ohne Ueberdeckung

Will man nun die Sicherheit eines Ablaufdiagramms überprüfen, so simuliert man das Ablaufdiagramm systematisch und notiert sich alle erzeugten Markierungen. Wenn eine neue Markierung eine bereits vorhandene Markierung überdeckt, so kann die Simulation abgebrochen werden, denn ein Ablaufdiagramm ist nur dann sicher, wenn es bei dessen Simulation keine sich überdeckenden Markierungen gibt. Die systematische Simulation des Ablaufdiagramms ist das Erzeugen eines Erreichbarkeitsgraphen und wird im folgenden Abschnitt erklärt.

Wie entstehen nicht sichere Ablaufdiagramme? Zwei Möglichkeiten drängen sich auf:

1. 'Rücksprung' aus parallelen Abläufen vor den Start der parallelen Abläufe. Dies ist in Abbildung 1 mit Verbesserung in Abbildung 2 gezeigt.
2. 'Entweder-Oder'-Zusammenführung von parallelen Prozessen. In Abbildung 3 ist ein solcher Ablauf dargestellt. Sind beide 'Warte'-Schritte markiert, so besteht klar ein Vorwärtskonflikt. Es entsteht eine Markierung wie in Abbildung 3. Diese überdeckt diejenige Markierung, in welcher nur der Schritt bereit markiert ist. Das Ablaufdiagramm ist in Abbildung 4 richtig gezeichnet.

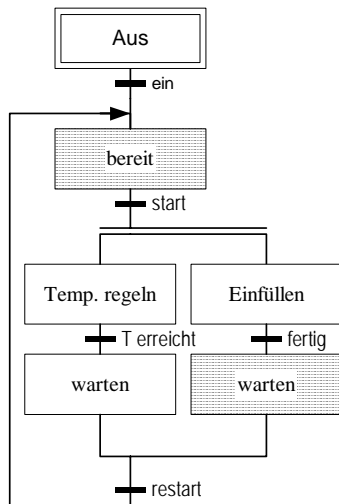
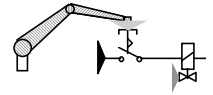


Abbildung 3: Vorwärtskonflikt

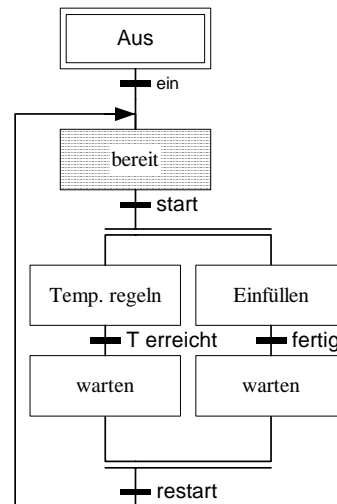
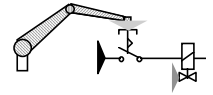


Abbildung 4: richtige Synchronisation



## Andere verwandte Darstellungen

### Petrinetz

Beispiel Roboter:

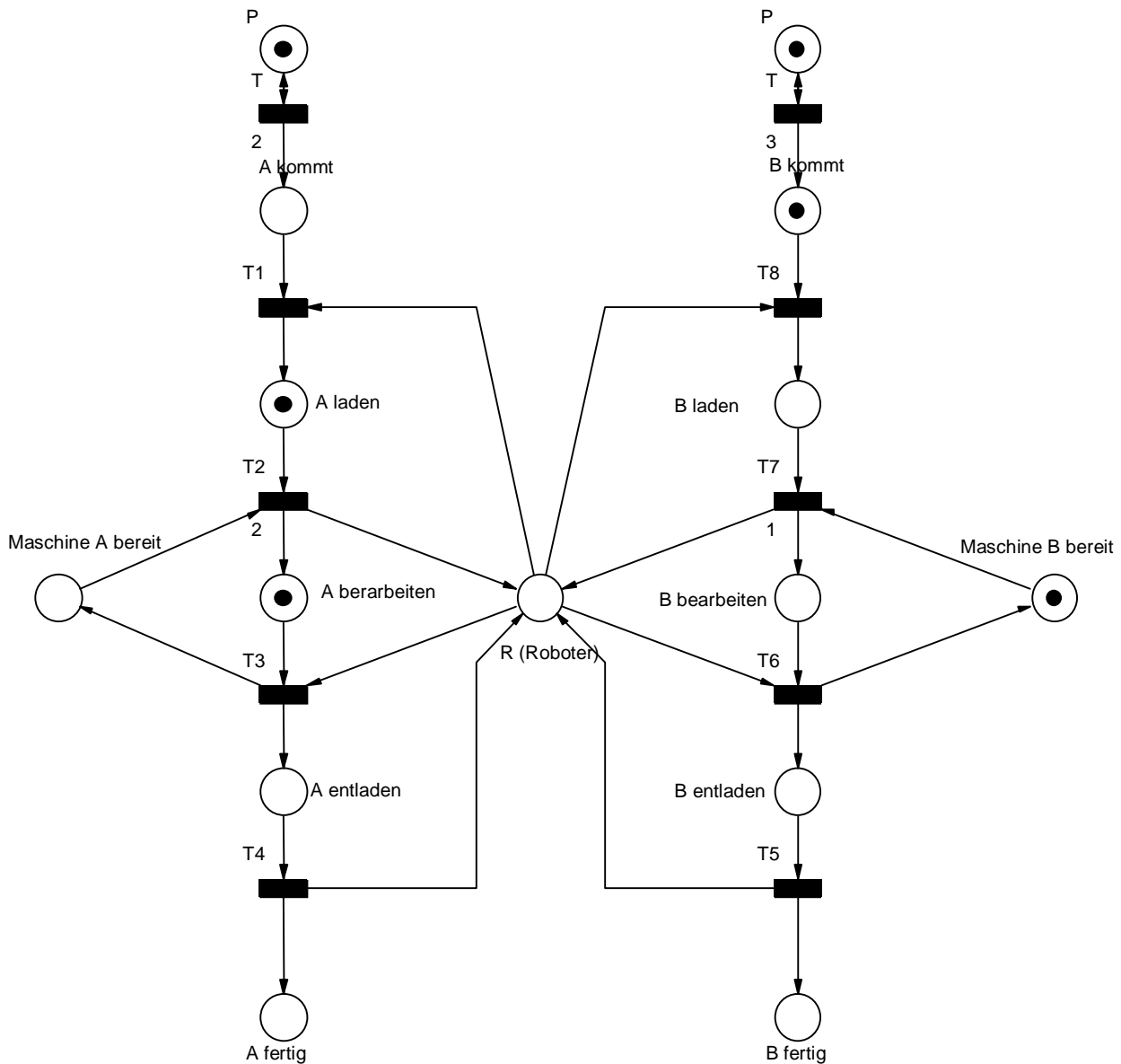
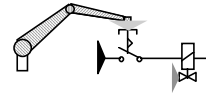


Abbildung 5: Beispiel eines Petri-Netzes

Das Petri-Netz ist dem Ablaufdiagramm von der Struktur her gleich. Zeichnerisch sind Synchronisationen und Start paralleler Aktionen im Petri-Netz versteckt und nicht durch eigene Symbole, wie in der Ablaufsprache dargestellt.

Die Zuordnung von Tätigkeiten kann sowohl zu Stellen als auch Transitionen erfolgen. Es gibt verschiedene Varianten von Petri-Netzen - im Extremfall ist eine Markierung ein Objekt, dessen Daten durch den Ablauf verändert werden.



## Zustandsautomat nach Mealy

Zustände sind über gerichtete Linien miteinander verbunden. An den Linien werden die Eingangssignale B und die Schalttätigkeiten T angegeben.

Das Motorenbeispiel im Zustandsdiagramm:

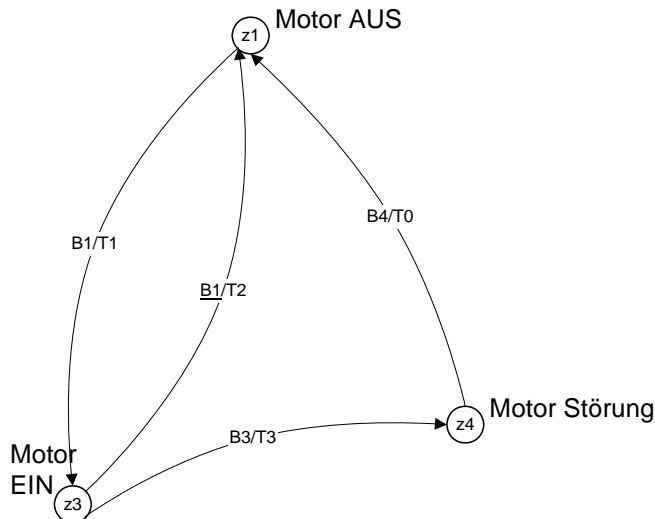


Abbildung 6: Beispiel eines Zustandsautomaten nach Mealy

Der Mealy-Automat ist sehr gut geeignet, um ereignisorientierte Bearbeitungen zu lösen. Beispiele dafür sind Benutzerführung in Automaten oder Programmen, Anwendungen in der Kommunikationstechnik und ereignisorientierte Steuerungen.

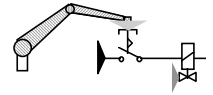
In der Implementierung enthalten solche Automaten oft einen Ereignis- oder Meldungsbuffer. Die Einträge in diesem Buffer werden entsprechend der Logik des Zustandsautomaten verarbeitet.

## Zustandsautomat nach Moore

Zustände sind über gerichtete Linien miteinander verbunden. An den Linien werden die Eingangssignale/Schaltsignal B angegeben. Die Schalttätigkeiten T werden den Zuständen zugeordnet. Diese werden durchgeführt, sobald der Zustand erreicht wird oder wenn der Automat im entsprechenden Zustand bearbeitet wird.

Typischerweise werden solche Automaten in einem zyklisch bearbeiteten Programm implementiert. Der Zustand gibt an, welcher Teil vom nachfolgenden Programm bearbeitet werden soll. Dabei kommen z.B. CASE-Switch-Strukturen zum Einsatz. Moore-Automaten sind auch nützlich, wenn mehrere Aufgaben zeitlich parallel durchgeführt werden müssen, ohne dass ein Betriebssystem diese Aufgabe für den Benutzer löst oder lösen kann. Die einzelnen Aufgaben werden mittels eines Zustandsautomaten in kleine Teilaufgaben unterteilt. Dadurch kann das Bearbeiten der Aufgabe unterbrochen werden, weil aufgrund der Zustandsinformation am richtigen Ort wieder weitergefahren werden kann. Somit können mehrere Aufgabe stückweise, dafür aber parallel gelöst werden.

Bsp: Ein Gerät, das Regelfunktionen in einem festen Zyklus durchführen muss, hat die Aufgabe, eine Menge von Parametern in ein Eeprom zu speichern. Meistens braucht es dafür viel mehr Zeit als zwischen zwei Regelzyklen zur Verfügung steht. Das Schreiben in das Eeprom kann aber mit einem Zustandsautomaten in kleine Teilaufgaben unterteilt werden.



Das Motorenbeispiel im Zustandsdiagramm:

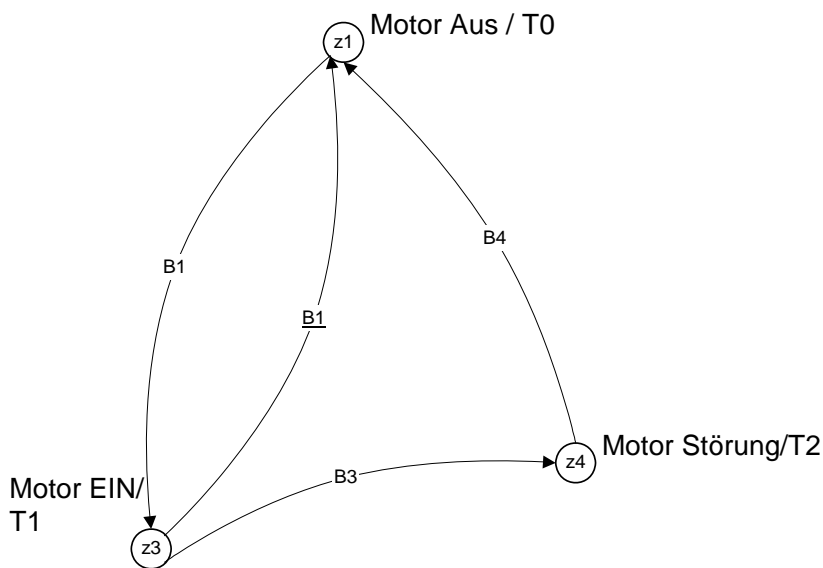


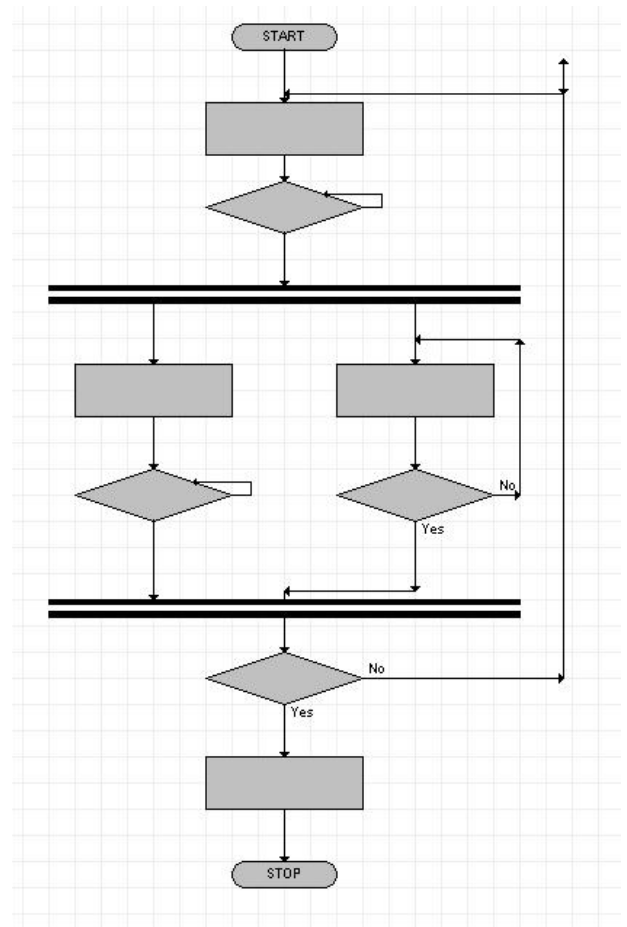
Abbildung 7: Beispiel eines Zustandsautomaten nach Moore

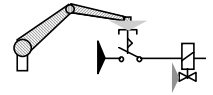
## Das Flussdiagramm

Das Flussdiagramm sei die natürlichste Art, einen Ablauf darzustellen. Dies behaupten namhafte Hersteller von solchen Engineeringtools (<http://www.entivity.com>).

Sicher gehört das Flussdiagramm zu den bekanntesten Darstellungen von Abläufen, hat aber betreffend seiner Anwendbarkeit in der Automatisierungstechnik klare Limiten. Dies führte auch dazu, dass das Flussdiagramm um Elemente erweitert werden musste. Die folgende Abbildung zeigt ein Flussdiagramm mit parallelen Abläufen. Deutlich zu erkennen sind die Synchronisationbalken, wie sie auch in der Ablaufsprache festgelegt wurden. Die Berechnung eines Flussdiagramms in einer Steuerung erfolgt nach folgender Regel: das Flussdiagramm wird verarbeitet, bis eine Linie nach oben führt. Die Berechnung des Flussdiagramms wird dann verlassen und andere Aufgaben können bearbeitet werden. Ist das Flussdiagramm wieder an der Reihe, so wird dort weitergefahren, wo der Rücksprungpfeil in einen Abwärtspfad mündet. In den Romben mit den Entscheidungskriterien sind Transitionsbedingungen dargestellt, an anderen Orten ist die Transitionsbedingung nur implizit vorhanden.

Abbildung 8: Flussdiagramm





## Vergleich der Automatendarstellungen

Diagramm	Stärken	Schwächen
Ablaufsprache	Automaten in der Steuerungstechnik mit parallelen Abläufen Schaltbedingungen explizit dargestellt	Simulation von komplexeren Prozessen Arbeiten mit Qualifiers etwas mühsam
Petri-Netz	Automaten in der Steuerungstechnik mit parallelen Abläufen Schaltbedingungen explizit dargestellt Geeignet für viele verschiedenste Problemstellungen	Synchronisationen nicht offensichtlich Aktionen in Transitionen und Stellen kann zu sehr unterschiedlichen Darstellungen des gleichen Problems führen Ueberspringen von Aktionen nicht definiert: komplexere Fehlerbehandlung
Mealy- und Moore Automat	Ereignisorientierte Automaten ohne parallel Abläufe Schaltbedingungen explizit dargestellt	Synchronisation zwischen verschiedenen Abläufen ist mühsam
Flussdiagramm	einfach Darstellung von sequentiellen Abläufen	oft nur implizite Transitionen nur binäre Verzweigungen möglich entspricht nicht dem Zustandsdenken der Steuerungstechnik