Ein einfaches Modell zur Fehlerfortpflanzung

Jens Chr. Lisner

lisner@dc.uni-due.de

ICB / Universität Duisburg-Essen

Problemstellung

Üblich bei der Formalisierung von Systemen:

- Formale Definition des Systemverhaltens
- Verhalten von Komponenten im Fehlerfall (häufig informell)
- Fortpflanzung von Fehlern über Systemkomponenten hinweg (häufig implizit)

Problemstellung

Formale Beschreibung von Systemverhalten

- Automaten (DFA, NFA)
- Zeitgesteuerte Automaten
- Petri-Netze
- Markov-Ketten
- Prozessalgebren
- als Pseudo-Code
- Timed Grammar
- ...

Problemstellung

Formale Beschreibung von Fehlverhalten

- explizit:
  Modellieren des fehlerhaften Verhaltens in das Systemverhalten
- implizit:
  z.B. Modellieren von fehlerhaftem Verhalten an den Eingängen fehlerfreier Systemkomponenten

Problemstellung

Formale Beschreibung der Fehlerfortpflanzung

Was ist Fehlerfortpflanzung?

Ausbreitung eines Fehlers über benachbarte Systemkomponenten.

Problemstellung

Formale Beschreibung der Fehlerfortpflanzung

Was ist Fehlerfortpflanzung?

Ausbreitung eines Fehlers über benachbarte Systemkomponenten.

Was muß ein solches Modell leisten?

Formulierung des Zusammenhangs zwischen den Fehlern an der Eingabe zu einer Komponente und der Ausgabe aus einer Komponente.

Anforderungen

Anforderungen an eine mögliche Lösung

- Einfachheit:
  die zu untersuchenden Systeme zeigen kein komplexes dynamisches Verhalten

Anforderungen

Anforderungen an eine mögliche Lösung

- Einfachheit:
  die zu untersuchenden Systeme zeigen kein komplexes dynamisches Verhalten
- Komponentensicht im Vordergrund (im ggs. zur algorithmischen Sicht):
  leichter intuitiv erfassbar

Anforderungen

Anforderungen an eine mögliche Lösung

- Einfachheit:
  die zu untersuchenden Systeme zeigen kein komplexes dynamisches Verhalten
- Komponentensicht im Vordergrund (im ggs. zur algorithmischen Sicht):
  leichter intuitiv erfassbar
- Berücksichtigung der Zeit:
  Ausbreitung von Fehlern über Komponenten braucht Zeit

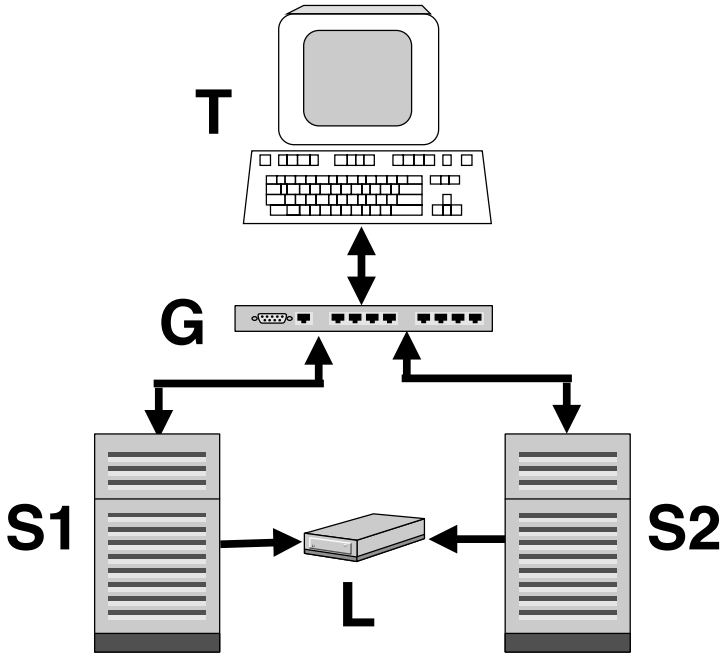
Anforderungen

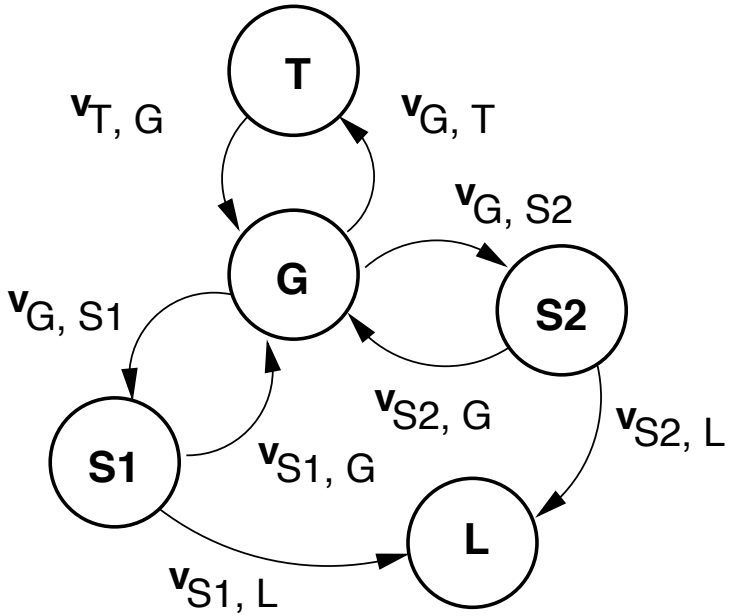
Anforderungen an eine mögliche Lösung

- Einfachheit:
  die zu untersuchenden Systeme zeigen kein komplexes dynamisches Verhalten
- Komponentensicht im Vordergrund (im ggs. zur algorithmischen Sicht):
  leichter intuitiv erfassbar
- Berücksichtigung der Zeit:
  Ausbreitung von Fehlern über Komponenten braucht Zeit
- Fehlernachverfolgung

Systemdarstellung

Darstellung erfolgt als gerichteter Graph





Systemverhalten

Reduktion des (gewünschtem) Verhaltens auf

- Einem von zwei Zuständen (*aktiv* und *inaktiv*)
- Pro Komponente gegenüber einer Nachbarkomponente
- Zu einem bestimmten Zeitpunkt t, mit $t_{start} \leq t \leq t_{end} < \infty$

Systemverhalten

Reduktion des (gewünschtem) Verhaltens auf

- Einem von zwei Zuständen (*aktiv* und *inaktiv*)
- Pro Komponente gegenüber einer Nachbarkomponente
- Zu einem bestimmten Zeitpunkt t, mit $t_{start} \leq t \leq t_{end} < \infty$

Was bedeutet aktiv und inaktiv?

- Signal liegt an / Signal liegt nicht an
- Eine Nachricht wird gesendet / Nachricht wird nicht gesendet
- Es erfolgt ein Zugriff auf einen bestimmten Adressbereich / Es erfolgt kein Zugriff

Systemverhalten

Beispiel (jeder Schritt ist aktives Verhalten):

1. Zum Zeitpunkt $t_{start} = 0$ wird eine Anfrage vom Terminal gestellt.
2. Die Abfrage wird weitergeleitet an **S1** und **S2**
3. **S1** und **S2** antworten an **G**, und senden eine Nachricht an **L**
4. **G** wählt eine Antwort aus, und sendet sie an **T**

Inaktives Verhalten:
Zwischen zwei Komponenten werden keine Nachrichten ausgetauscht.

Protokoll

Jeder Verbindung wird zu einem bestimmten Zeitpunkt ein Verhalten zugeordnet.

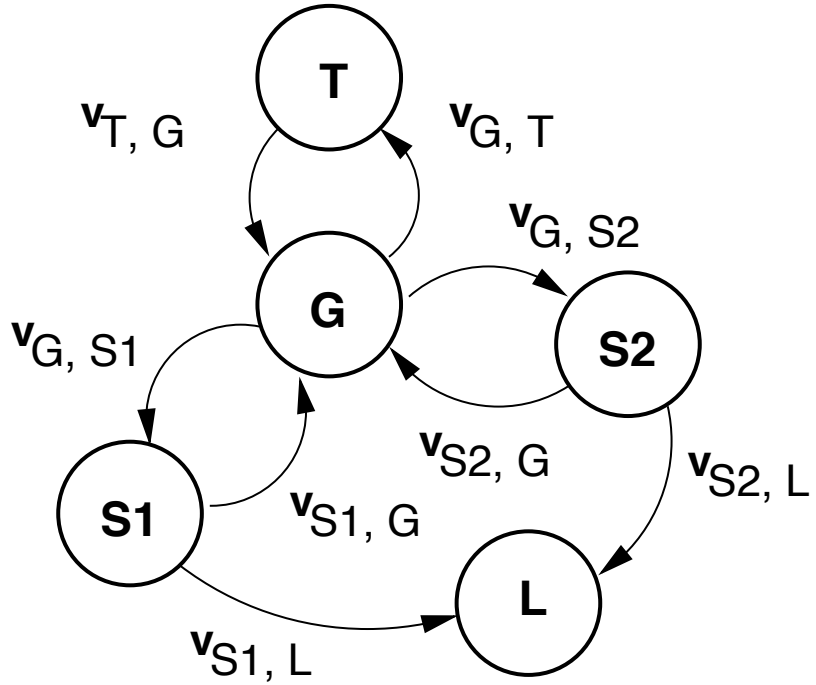
$$\sigma(v_{i,j}, t) \in \{0, 1\}$$

$v_{i,j}$ Verbindung zwischen Komponenten i und j
t Zeitpunkt
0 inaktiv
1 aktiv

Protokoll

Beispiel

Zeit	1	2	3	4
Verbindung	$v_{T,G}$	$v_{G,S1}$ $v_{G,S2}$	$v_{S1,G}$, $v_{S1,L}$ $v_{S2,G}$, $v_{S2,L}$	$v_{G,T}$



Beobachtes Verhalten

System verhält sich nicht unbedingt immer Spezifikationsgemäß ...

α liefert das beobachtete Verhalten

Das Verhalten der Komponente i gegenüber Komponente j an Kante $v_{i,j}$ zum Zeitpunkt t ist:

$$\alpha(v_{i,j},t) = \begin{cases} 1 & \text{falls Verhalten aktiv} \\ 0 & \text{falls Verhalten inaktiv} \\ a & \text{mit } 0 < a < 1 \end{cases}$$

im Fall $0 < \alpha(v_{i,j},t) < 1$ gilt das Verhalten als „schwach" aktiv.

Fehler

Eine Komponente verhält sich fehlerhaft, wenn ...

... ein Ist-Zustand von einem Soll-Zustand abweicht.

Fehler

Eine Komponente verhält sich fehlerhaft, wenn ...

... ein Ist-Zustand von einem Soll-Zustand abweicht.

Für

$$\kappa(v_{i,j},t) = \sigma(v_{i,j},t)\,\alpha(v_{i,j},t) + (1-\sigma(v_{i,j},t))\,(1-\alpha(v_{i,j},t))$$

gilt:

Das Verhalten an Verbindung $v_{i,j}$ zum Zeitpunkt t ist ...

Fehler

Eine Komponente verhält sich fehlerhaft, wenn ...

... ein Ist-Zustand von einem Soll-Zustand abweicht.

Für

$$\kappa(v_{i,j},t) = \sigma(v_{i,j},t)\,\alpha(v_{i,j},t) + (1-\sigma(v_{i,j},t))\,(1-\alpha(v_{i,j},t))$$

gilt:

Das Verhalten an Verbindung $v_{i,j}$ zum Zeitpunkt t ist ...

fehlerfrei, wenn $\kappa(v_{i,j},t) = 1$
fehlerhaft, wenn $\kappa(v_{i,j},t) < 1$

Fehlerfortpflanzung

Gesucht:

Darstellung des Einflusses des Verhaltens einer Komponente auf das Verhalten einer anderen.

- Dargestellt durch eine Funktion $\phi(v_{i,j}, t)$
- Abhängig von Verhalten der Komponente $\kappa(v_{i,j}, t)$
- Abhängig von dem Verhalten (einer Teilmenge) der Eingaben

Fehlerfortpflanzung

Einfaches Beispiel:

Über einen Broadcast-Bus kann nur ein Knoten senden ohne Kollisionen zu Verursachen.

- Jeder Knoten i ist durch ein $v_{i,c}$ mit Kanal c verbunden.
- Kanal c ist mit jedem Knoten i über ein $v_{c,i}$ verbunden.

$$\phi(v_{j,i}, t+d) = \kappa(v_{j,i}, t+d) \prod_{i=1}^{n} \phi(v_{i,j}, t)$$

wobei d die Verzögerung ist.

Eigenschaften

- Fehler können keine Fehler beheben.

Eigenschaften

- Fehler können keine Fehler beheben.
- Verbindungen dürfen nicht zyklisch voneinander abhängen (es sei denn es vergeht zwischendurch Zeit).

Eigenschaften

- Fehler können keine Fehler beheben.
- Verbindungen dürfen nicht zyklisch voneinander abhängen (es sei denn es vergeht zwischendurch Zeit).
- Rekursionsanfang: F.a. Verbindungen die von keinem Eingang abhängen gilt $\phi(v,t) = \kappa(v,t)$

Eigenschaften

- Fehler können keine Fehler beheben.
- Verbindungen dürfen nicht zyklisch voneinander abhängen (es sei denn es vergeht zwischendurch Zeit).
- Rekursionsanfang: F.a. Verbindungen die von keinem Eingang abhängen gilt $\phi(v,t) = \kappa(v,t)$
- Alle Ereignisse an den Eingaben sollten zeitlich vor dem Ereignis an den Ausgaben liegen.

Eigenschaften

- Fehler können keine Fehler beheben.
- Verbindungen dürfen nicht zyklisch voneinander abhängen (es sei denn es vergeht zwischendurch Zeit).
- Rekursionsanfang: F.a. Verbindungen die von keinem Eingang abhängen gilt $\phi(v,t) = \kappa(v,t)$
- Alle Ereignisse an den Eingaben sollten zeitlich vor dem Ereignis an den Ausgaben liegen.
- Die Funktion $\phi(v,t)$ darf beliebig phantasievoll gestaltet sein.

Fehlerfortpflanzung

Hinweg:

Komponente $\mathbf{T}$

$$\phi(v_{\mathbf{T},\mathbf{G}}, t) = \kappa(v_{\mathbf{T},\mathbf{G}}, t)$$

Komponente $\mathbf{G}$

$$\phi(v_{\mathbf{G},\mathbf{S1}}, t+1) = \kappa(v_{\mathbf{G},\mathbf{S1}}, t+1)\ \phi(v_{\mathbf{T},\mathbf{G}}, t)$$
$$\phi(v_{\mathbf{G},\mathbf{S2}}, t+1) = \kappa(v_{\mathbf{G},\mathbf{S2}}, t+1)\ \phi(v_{\mathbf{T},\mathbf{G}}, t)$$

Komponente $\mathbf{S1}$ ($\mathbf{S2}$ entsprechend)

$$\phi(v_{\mathbf{S1},\mathbf{L}}, t+1) = \kappa(v_{\mathbf{S1},\mathbf{L}}, t+1)\ \phi(v_{\mathbf{G},\mathbf{S1}}, t)$$

Komponente $\mathbf{L}$

$$\mathrm{RES}(t) = \phi(v_{\mathbf{S1},\mathbf{L}}, t)\ \phi(v_{\mathbf{S2},\mathbf{L}}, t)$$

Fehlerfortpflanzung

Rückweg:

Komponente **S1** (**S2** entsprechend)

$$\phi(v_{\mathbf{S1},\mathbf{G}}, t+1) = \kappa(v_{\mathbf{S1},\mathbf{G}}, t+1)\ \phi(v_{\mathbf{G},\mathbf{S1}}, t)$$

Komponente **G**

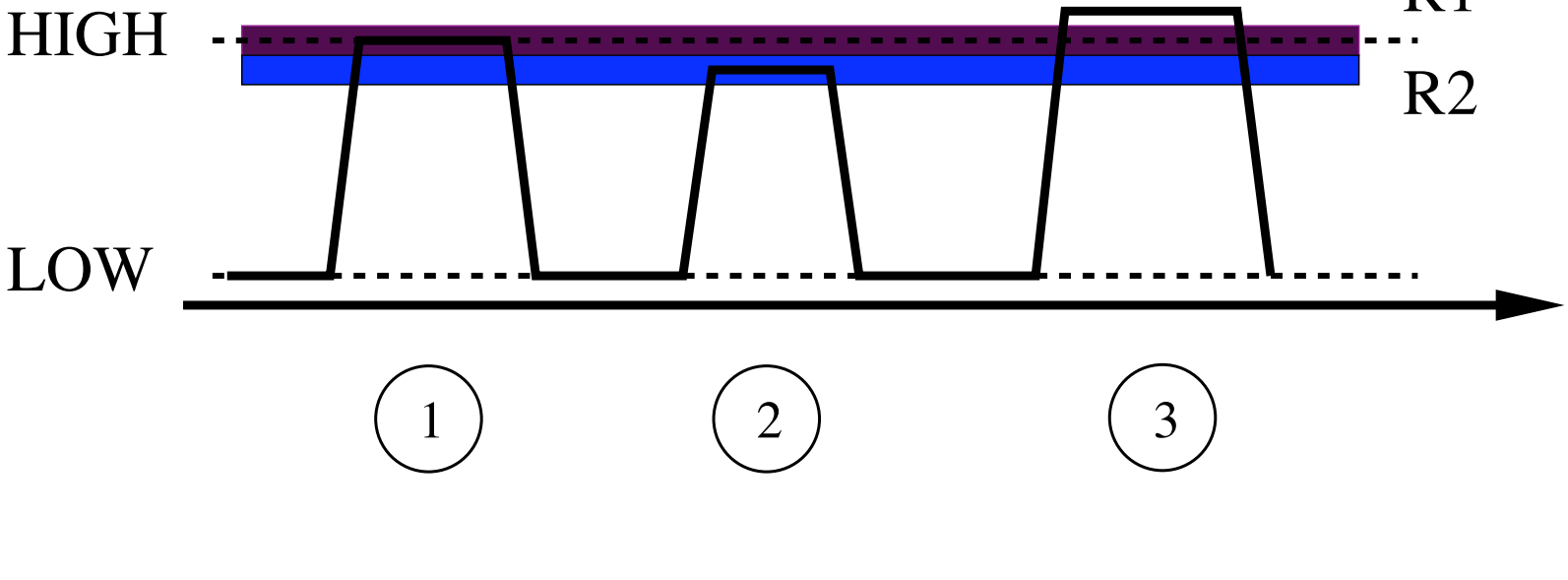
$$\phi(v_{\mathbf{G},\mathbf{T}}, t+1) = \kappa(v_{\mathbf{G},\mathbf{T}}, t+1)\ (1-(1-\phi(v_{\mathbf{S1},\mathbf{G}}, t))\,(1-\phi(v_{\mathbf{S2},\mathbf{G}}, t))$$

Komponente **T**

$$\mathrm{RES}(t) = \phi(v_{\mathbf{G},\mathbf{T}}, t)$$

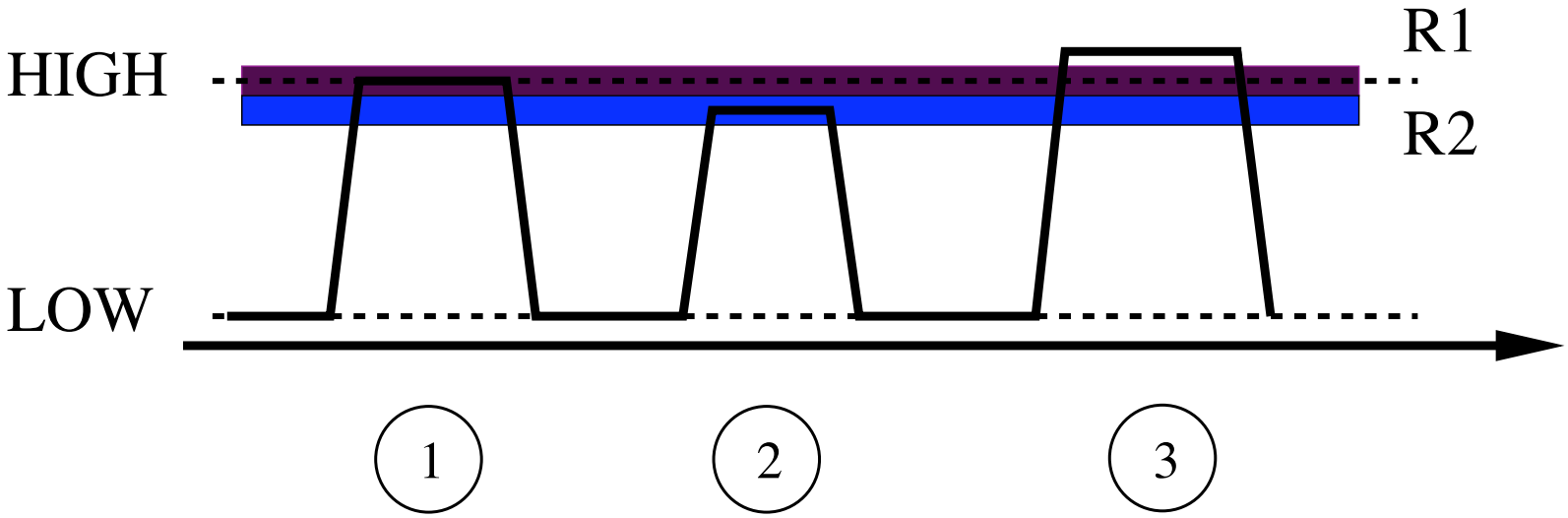
Schwache Aktivität

Empfänger R1 und R2 tasten ein Signal ab:



Schwache Aktivität

Empfänger R1 und R2 tasten ein Signal ab:

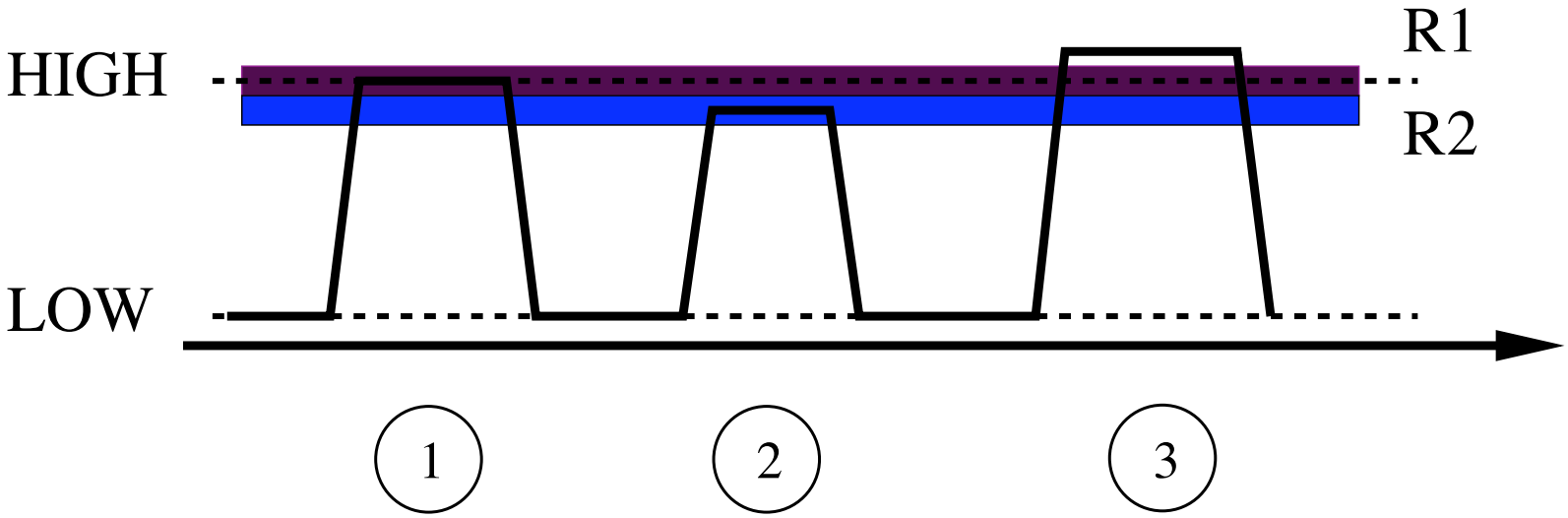


Bedingt durch unvermeidbarer Meßungenauigkeiten gilt:

- Punkt 1: Verhalten ist fehlerfrei für R1 und R2

Schwache Aktivität

Empfänger R1 und R2 tasten ein Signal ab:

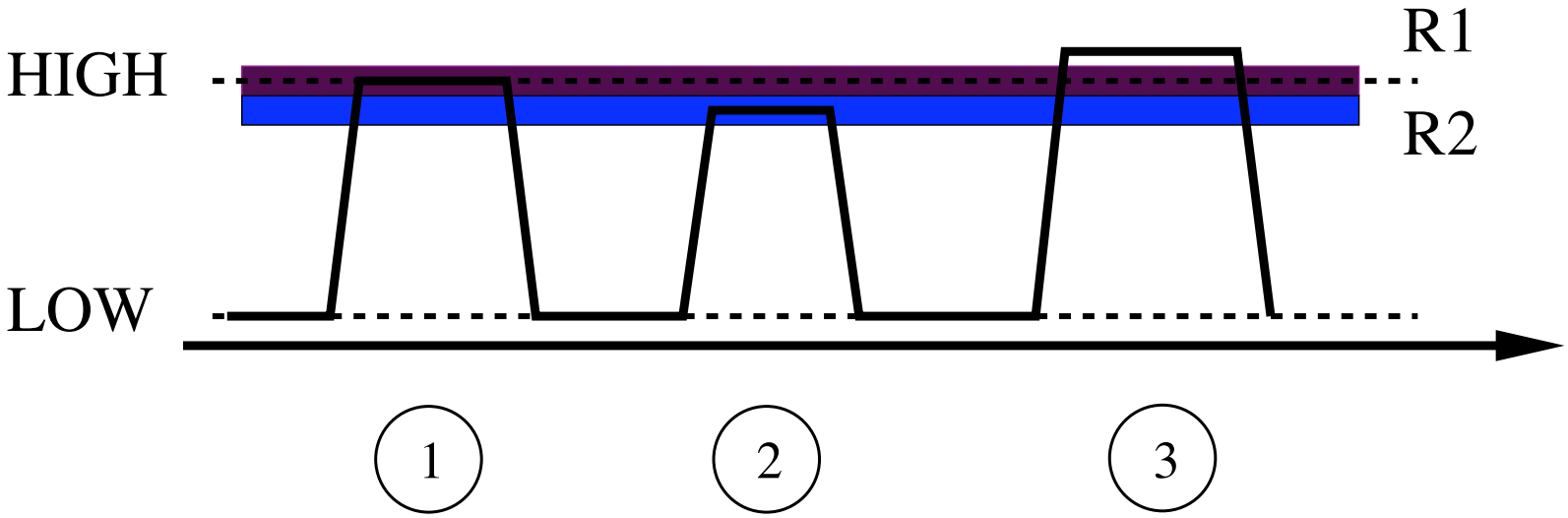


Bedingt durch unvermeidbarer Meßungenauigkeiten gilt:

- Punkt 1: Verhalten ist fehlerfrei für R1 und R2
- Punkt 2: Verhalten ist für R1 fehlerfrei, für R2 fehlerhaft

Schwache Aktivität

Empfänger R1 und R2 tasten ein Signal ab:

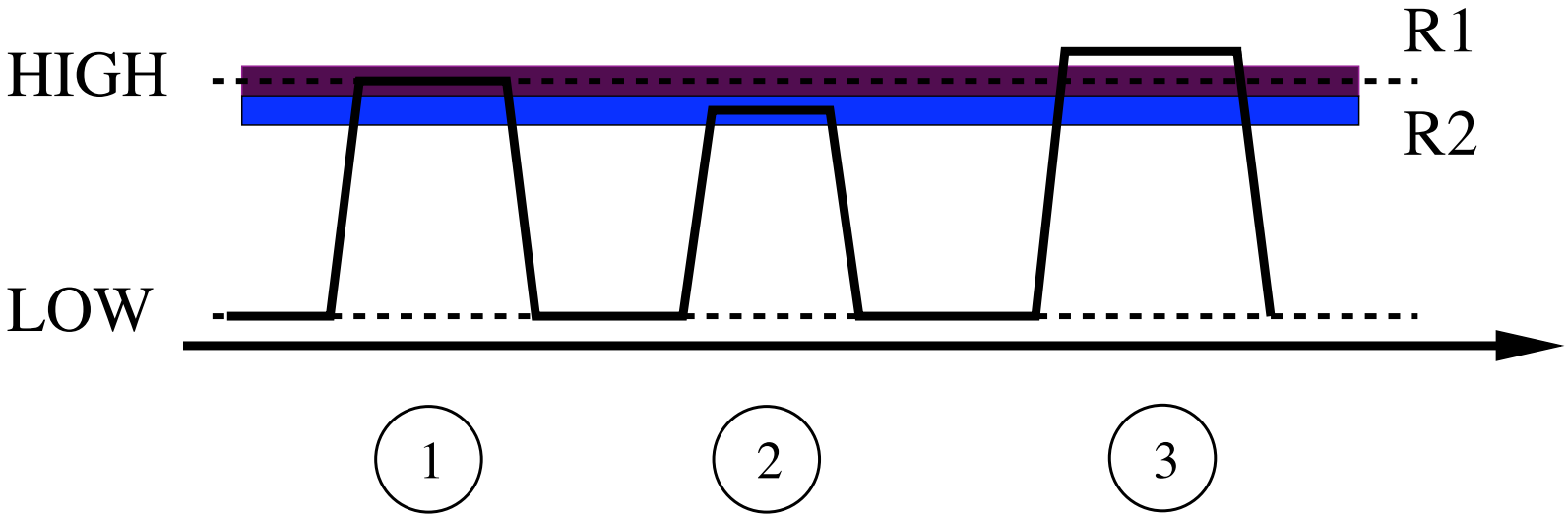


Bedingt durch unvermeidbarer Meßungenauigkeiten gilt:

- Punkt 1: Verhalten ist fehlerfrei für R1 und R2
- Punkt 2: Verhalten ist für R1 fehlerfrei, für R2 fehlerhaft
- Punkt 3: Verhalten ist für R1 und R2 fehlerhaft

Schwache Aktivität

Empfänger R1 und R2 tasten ein Signal ab:



Bedingt durch unvermeidbarer Meßungenauigkeiten gilt:

- Punkt 1: Verhalten ist fehlerfrei für R1 und R2
- Punkt 2: Verhalten ist für R1 fehlerfrei, für R2 fehlerhaft
- Punkt 3: Verhalten ist für R1 und R2 fehlerhaft
- Punkt 2+3: Verhalten ist fehlerhaft (objektiv betrachtet)!

Schwache Aktivität

Einige Komponenten müssen Entscheidungen treffen:

Ist die Eingabe fehlerfrei oder fehlerhaft?

Problem:

Manchmal gibt es einen nicht vermeidbaren Fehler, z.B.

- Uhrenabweichungen: Ein Signal wird zu einem bestimmten Zeitpunkt erwartet.
- Meßungenauigkeit bei der Messung der Signalstärke eines Signals.

Beides kann zu byzantinischen Fehlern führen!

Schwellenfunktion

Das Verhalten auf den Eingaben wird als fehlerfrei oder fehlerhaft bewertet

$$\theta(v,t,a_0,a_1) = \begin{cases} 0 & \text{falls } \sigma(v,t) = 0 \text{ und } \phi(v,t) \geq a_0 \\ 1 & \text{falls } \sigma(v,t) = 1 \text{ und } \phi(v,t) \geq a_1 \end{cases}$$

Beispiel:

R1: $RES(t) = \theta(v,t,0.9,0.95)$
R2: $RES(t) = \theta(v,t,0.8,0.9)$

Zusammenfassung

- Systeme werden als gerichtete Graphen dargestellt

Zusammenfassung

- Systeme werden als gerichtete Graphen dargestellt
- Verhalten der Komponenten wird auf aktiv und inaktiv reduziert.

Zusammenfassung

- Systeme werden als gerichtete Graphen dargestellt
- Verhalten der Komponenten wird auf aktiv und inaktiv reduziert.
- Komponenten verhalten sich (weitgehend) statisch

Zusammenfassung

- Systeme werden als gerichtete Graphen dargestellt
- Verhalten der Komponenten wird auf aktiv und inaktiv reduziert.
- Komponenten verhalten sich (weitgehend) statisch
- Verzögerungen können explizit formuliert werden

Zusammenfassung

- Systeme werden als gerichtete Graphen dargestellt
- Verhalten der Komponenten wird auf aktiv und inaktiv reduziert.
- Komponenten verhalten sich (weitgehend) statisch
- Verzögerungen können explizit formuliert werden
- Fehler ist die Differenz zwischen beobachtetem und gewünschtem Verhalten

Zusammenfassung

- Systeme werden als gerichtete Graphen dargestellt
- Verhalten der Komponenten wird auf aktiv und inaktiv reduziert.
- Komponenten verhalten sich (weitgehend) statisch
- Verzögerungen können explizit formuliert werden
- Fehler ist die Differenz zwischen beobachtetem und gewünschtem Verhalten
- Zu dem Fehlverhalten der Komponente kommt der Einfluß der Eingaben

Zusammenfassung

- Systeme werden als gerichtete Graphen dargestellt
- Verhalten der Komponenten wird auf aktiv und inaktiv reduziert.
- Komponenten verhalten sich (weitgehend) statisch
- Verzögerungen können explizit formuliert werden
- Fehler ist die Differenz zwischen beobachtetem und gewünschtem Verhalten
- Zu dem Fehlverhalten der Komponente kommt der Einfluß der Eingaben
- Es werden zwei Arten von byzantinischem Verhalten berücksichtigt