

Vorlesung
***Methodische Grundlagen des
Software-Engineering***
im Sommersemester 2014

Prof. Dr. Jan Jürjens

TU Dortmund, Fakultät Informatik, Lehrstuhl XIV

Teil 2.5: Konformanzanalyse

v. 07.07.2014

2.5 Konformanzanalyse

[mit freundlicher Genehmigung basierend
auf einem englischen Foliensatz von
Prof. Dr. Wil van der Aalst (TU Eindhoven)]

Literatur:

[vdA11] Wil van der Aalst: **Process Mining: Discovery, Conformance and Enhancement of Business Processes**, Springer-Verlag, 2011.

Unibibliothek (6 Exemplare): <http://www.ub.tu-dortmund.de/katalog/titel/1332248>
(Bei Engpässen kann eine **Kopiervorlage** der relevanten Ausschnitte zur Verfügung gestellt werden.)

- **Kapitel 7**

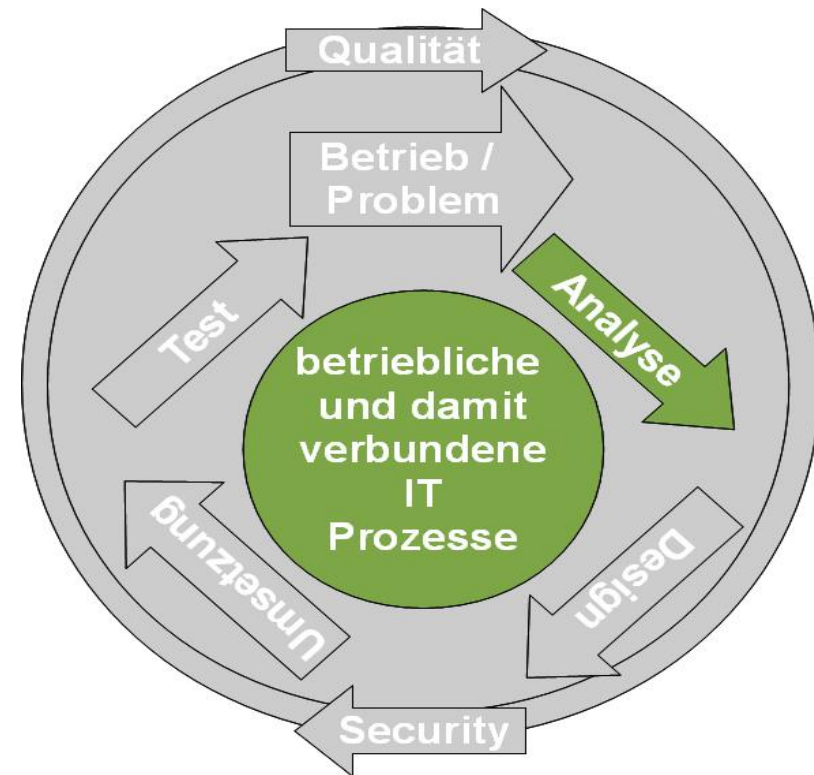
Einordnung Konformanzanalyse

- Geschäftsprozessmodellierung

- **Process-Mining**

- Einführung: Process-Mining
- Petrinetze
- Data-Mining
- Datenbeschaffung
- Prozessextraktion
- **Konformanzanalyse**
- Mining: Zusätzliche Perspektiven
- Betriebsunterstützung
- Werkzeugunterstützung
- Analysiere „Lasagne Prozesse“
- Analysiere „Spaghetti Prozesse“
- Kartographie und Navigation
- Epilog

- Modellbasierte Entwicklung sicherer Software



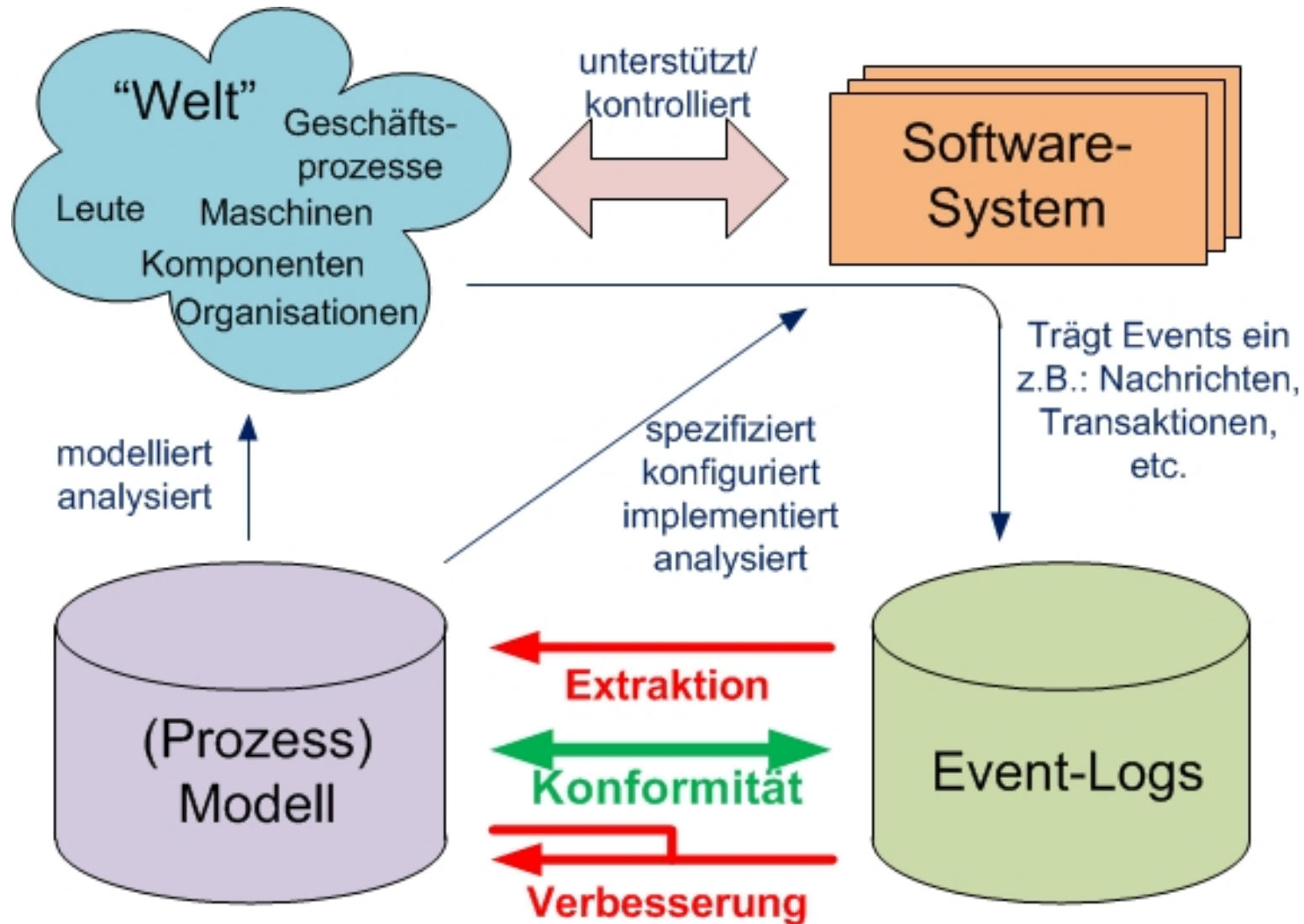
Einleitung

Konformanzanalyse

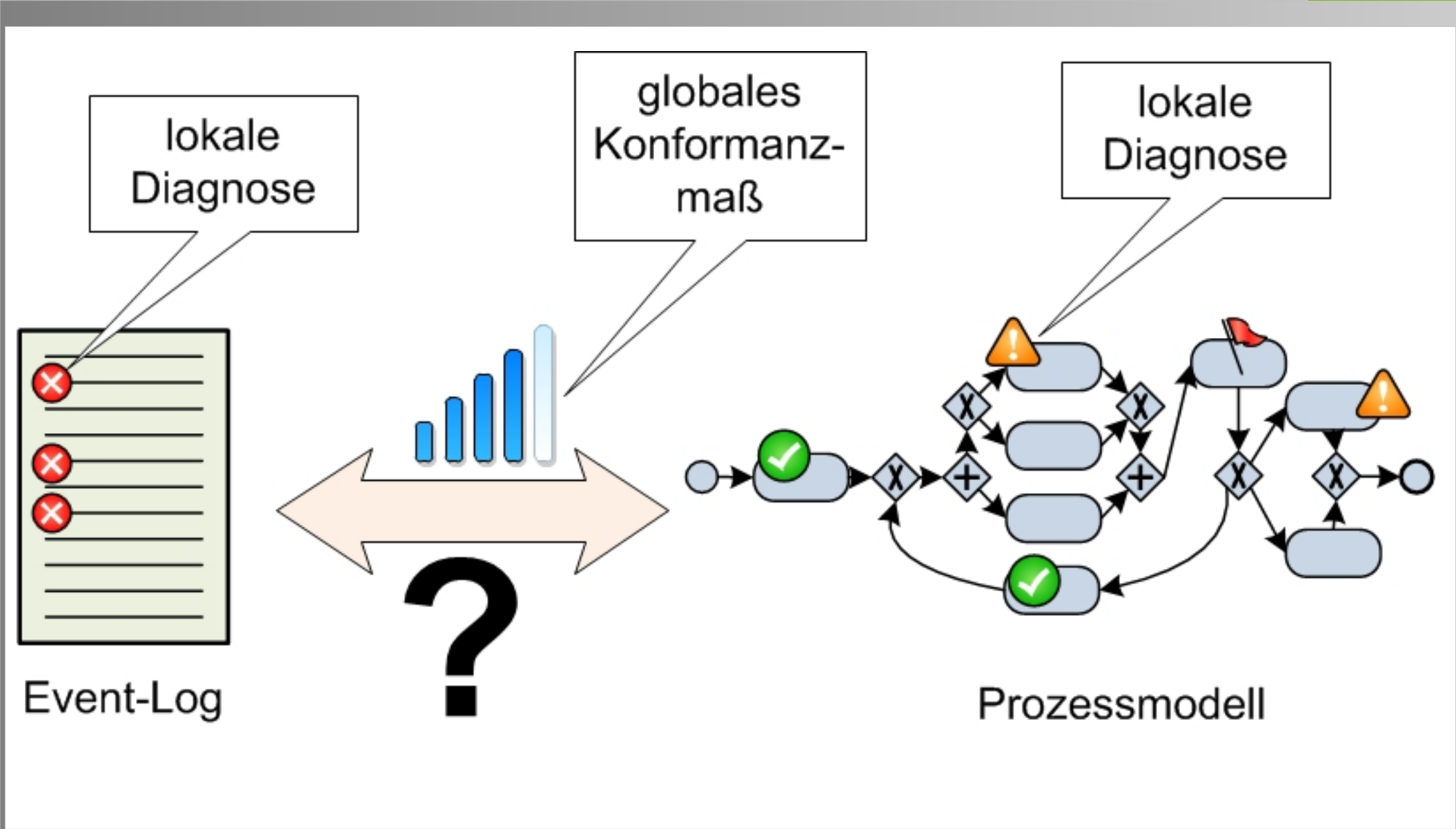
- **Letzter Abschnitt:** Prozessextraktion
- **Dieser Abschnitt:** „Konformanzanalyse“
 - Konkretes Beispiel: **Konformanzanalyse** durch Replay
 - Workflow-Diagnose:
 - Kommen Pfade häufig vor?
 - Wie ist die Laufzeit?

Überblick Konformanzanalyse

- **Einführung**
- Konformanzanalyse basierend auf Replay
 - Motivation
 - Beispiel
 - Konformanzmaße
- Diagnose



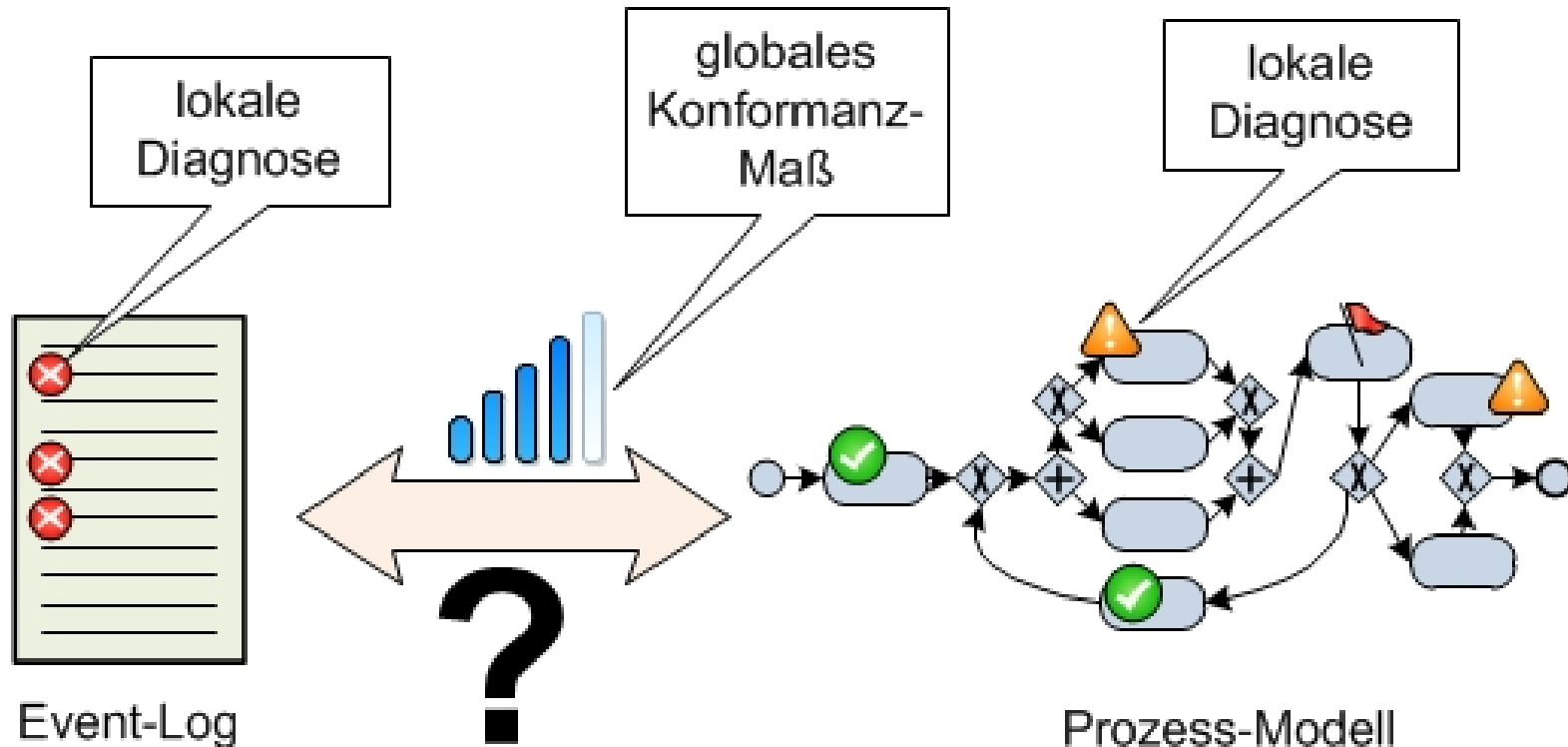
Konformanzanalyse



- **Corporate Governance, Risk, Compliance, und Gesetzgebung:**
 - z.B.: Sarbanes-Oxley (US), Basel II/III (EU), J-SOX (Japan), C-SOX (Kanada), 8th EU Directive (EURO-SOX), Bilanzrechts-modernisierungsgesetz (BilMoG) (Deutschland), Markets in Financial Instruments Directive (MiFID) (EU), Law 262/05 (Italien), Code Lippens (Belgien), und Code Tabaksblad (Niederlande).
- **ISO 9001:2008:** Unternehmen müssen Betriebsprozesse modellieren.
- **Business alignment:** Sicherstellen → Informationssystem und realer GP gut aufeinander abgestimmt.

- **Auditierung:** Evaluation von Unternehmen und ihren Prozessen.
- **Audits** stellen **Validität** und **Zuverlässigkeit** von Informationen über Unternehmen und entsprechende Prozesse sicher.
- Test von Ausführung der GP in bestimmten Grenzen (von Managern, der Politik und anderen Interessenvertretern gesetzt).
- **Process-Mining:** bei Aufdecken von **Betrug**, **Fehlverhalten**, **Risiken** und **Ineffizienzen** hilfreich.
- Alle Events eines GP evaluierbar, auch während Prozess noch läuft.

- **Modell** oder **Log** “falsch” ?
- “**Gewünschte**” oder “**unerwünschte**” Abweichungen ?
- “Glas zerbrechen” könnte Leben retten !



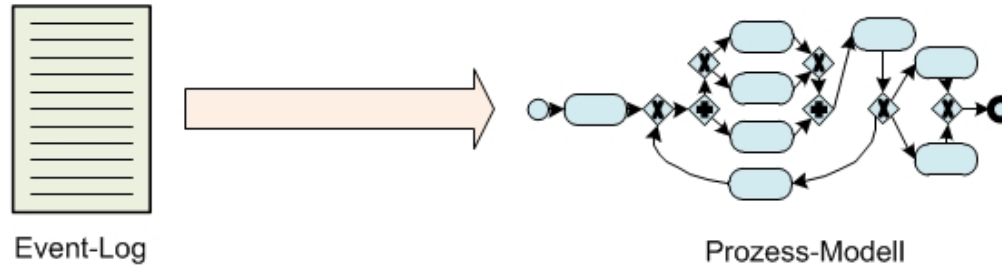
Überblick

Konformanzanalyse

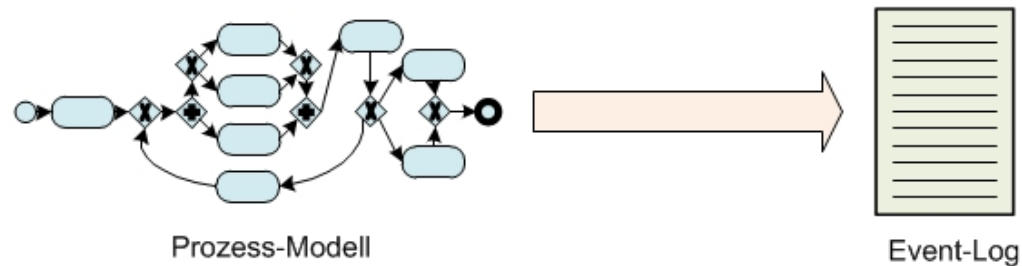
- Einführung
- **Konformanzanalyse basierend auf Replay**
 - Motivation
 - **Beispiel**
 - Konformanzmaße
- Diagnose

Wiederholung: Play-In, Play-Out, Replay

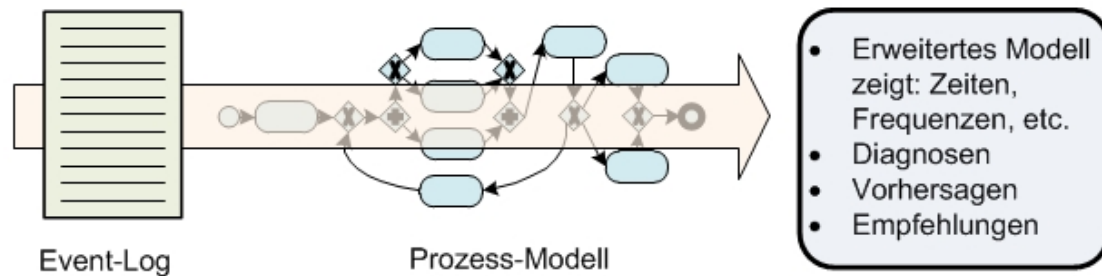
Play-In

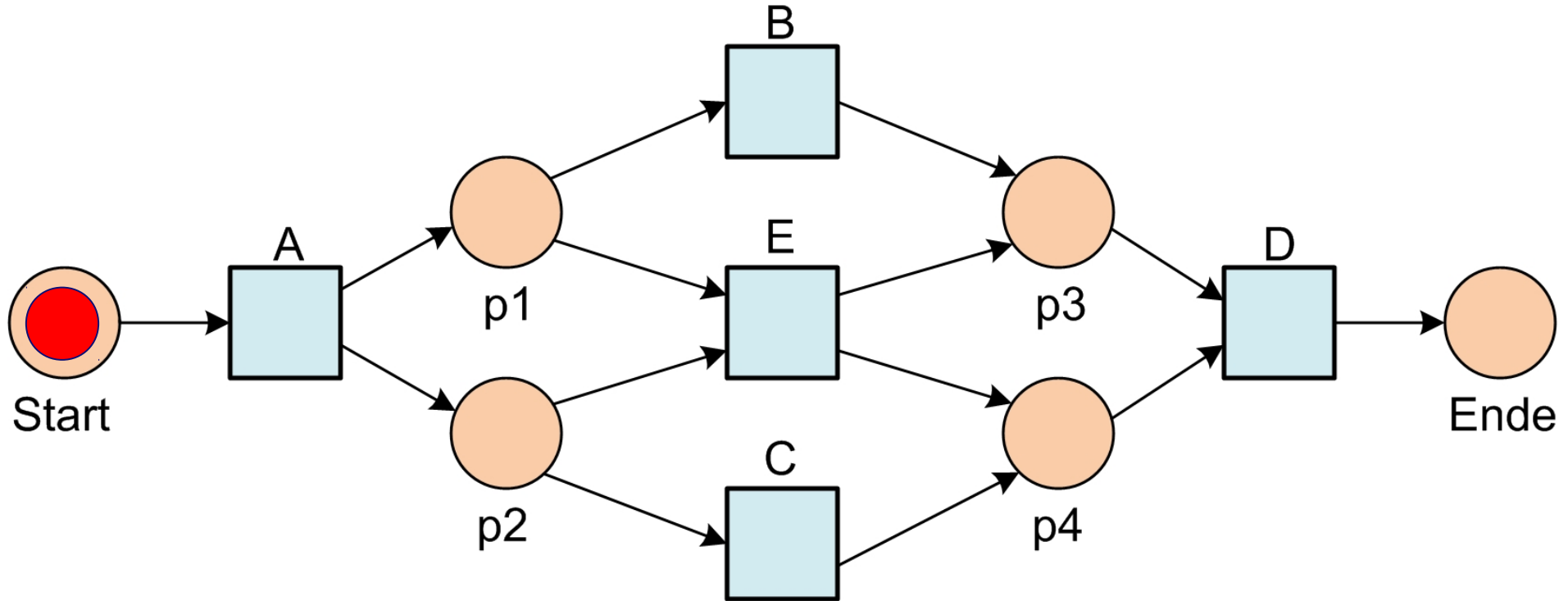


Play-Out



Replay

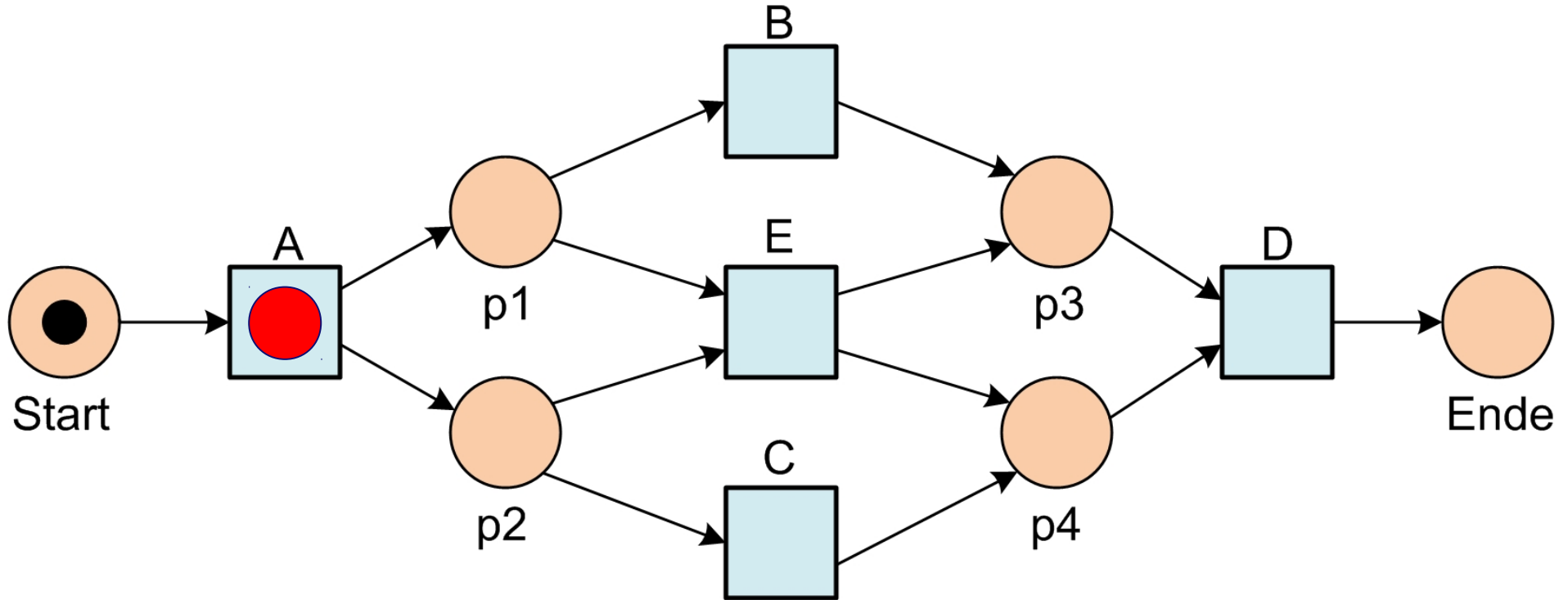




A

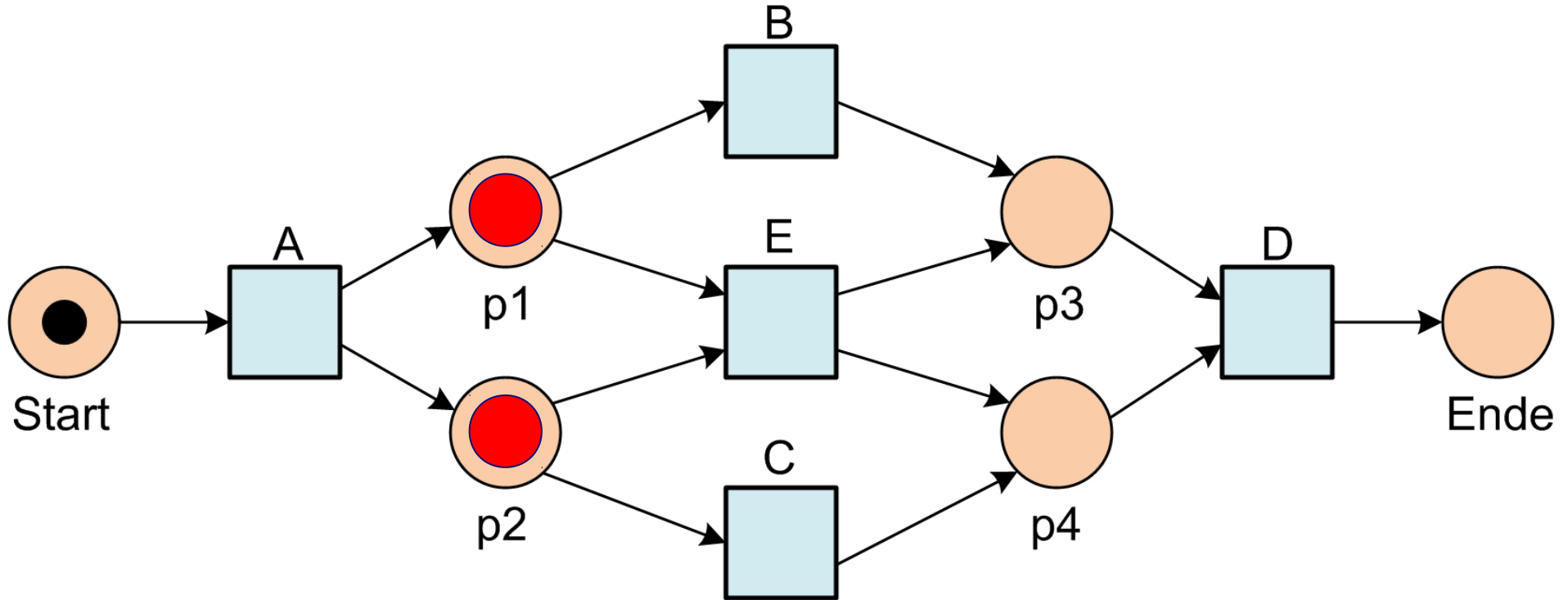
NB: Token auf Stelle ist nicht offizielle Petrinetznotation, sondern hier nur zur Veranschaulichung.

Play-Out: Beispiel



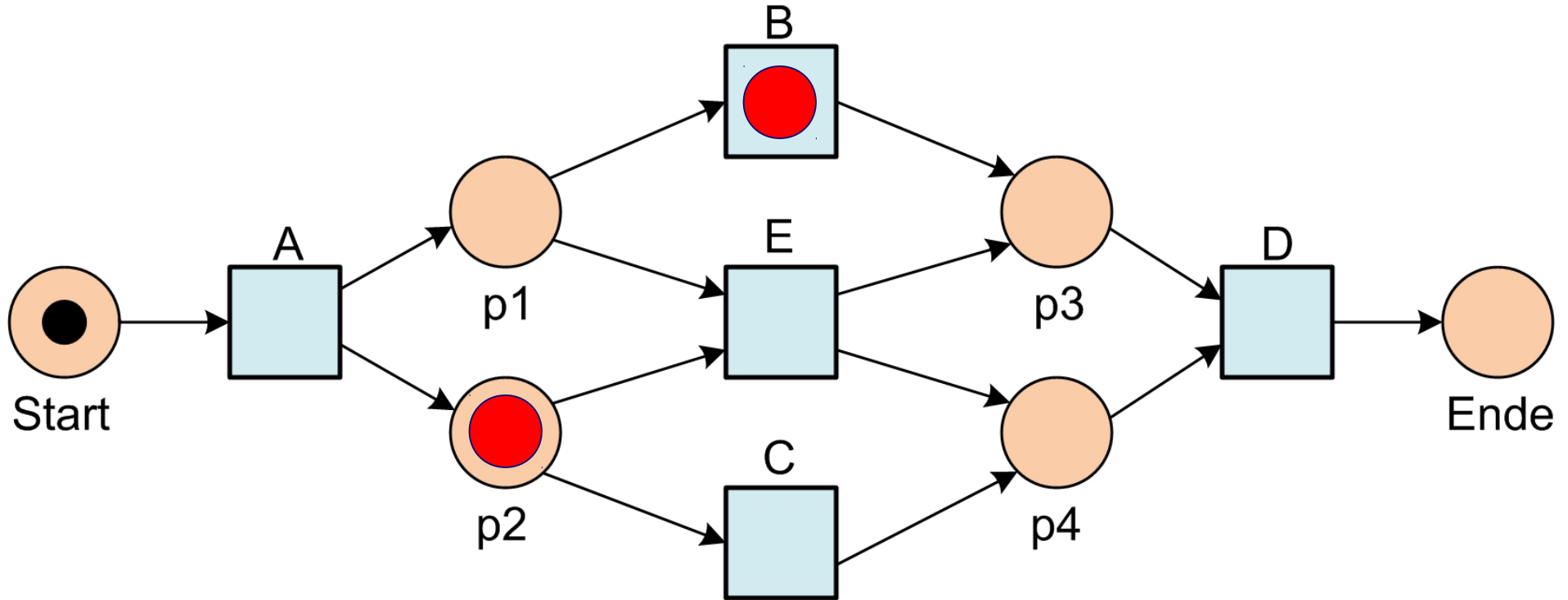
A

Play-Out: Beispiel



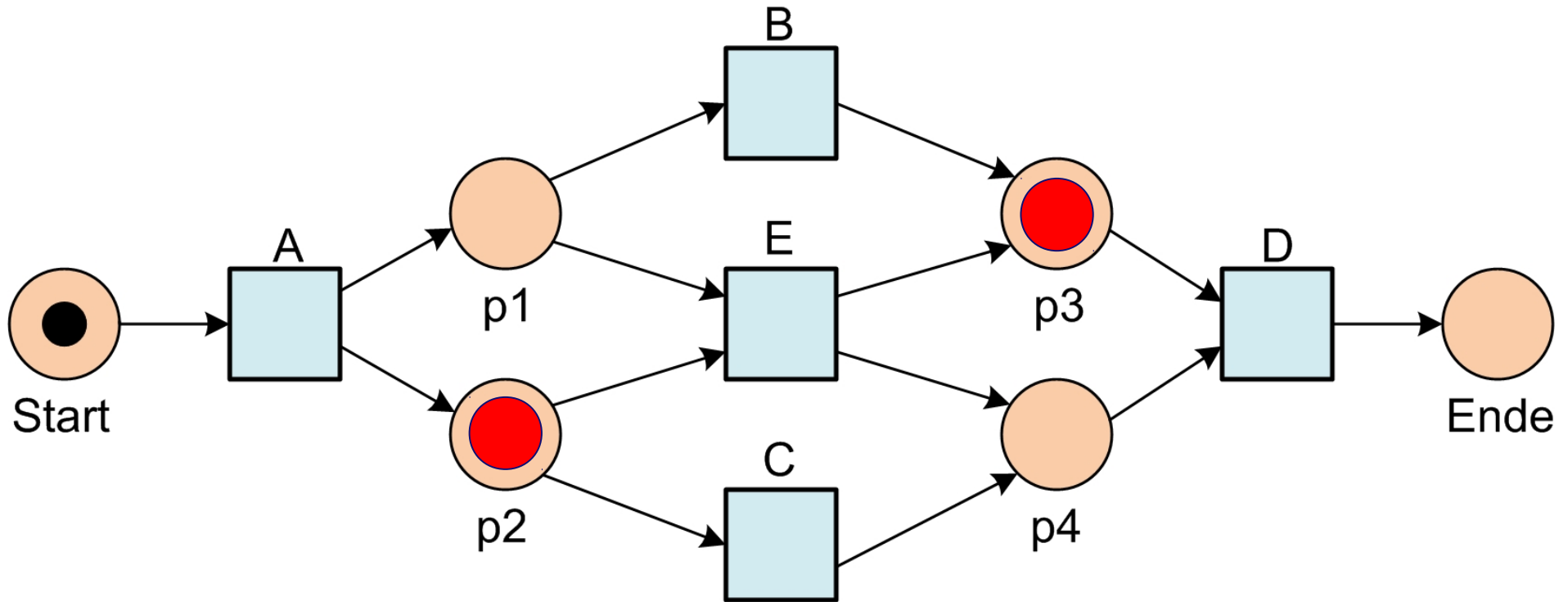
A

Play-Out: Beispiel



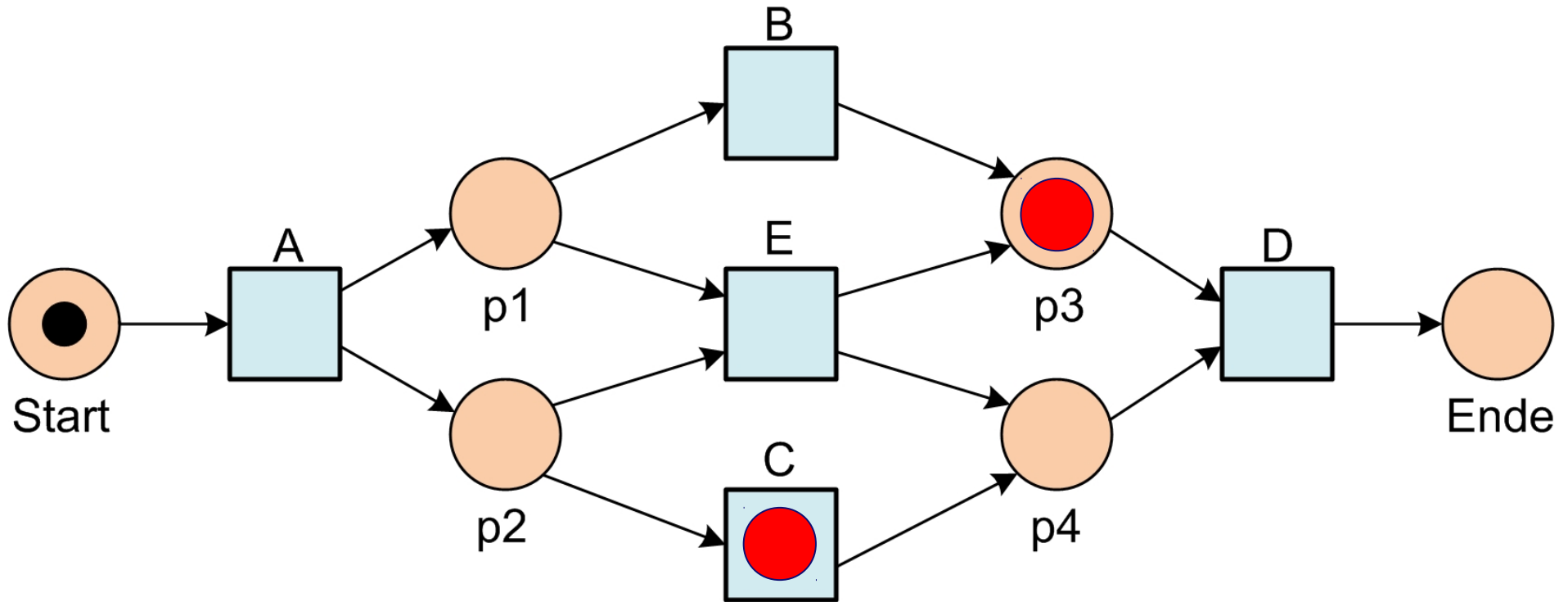
A B

Play-Out: Beispiel



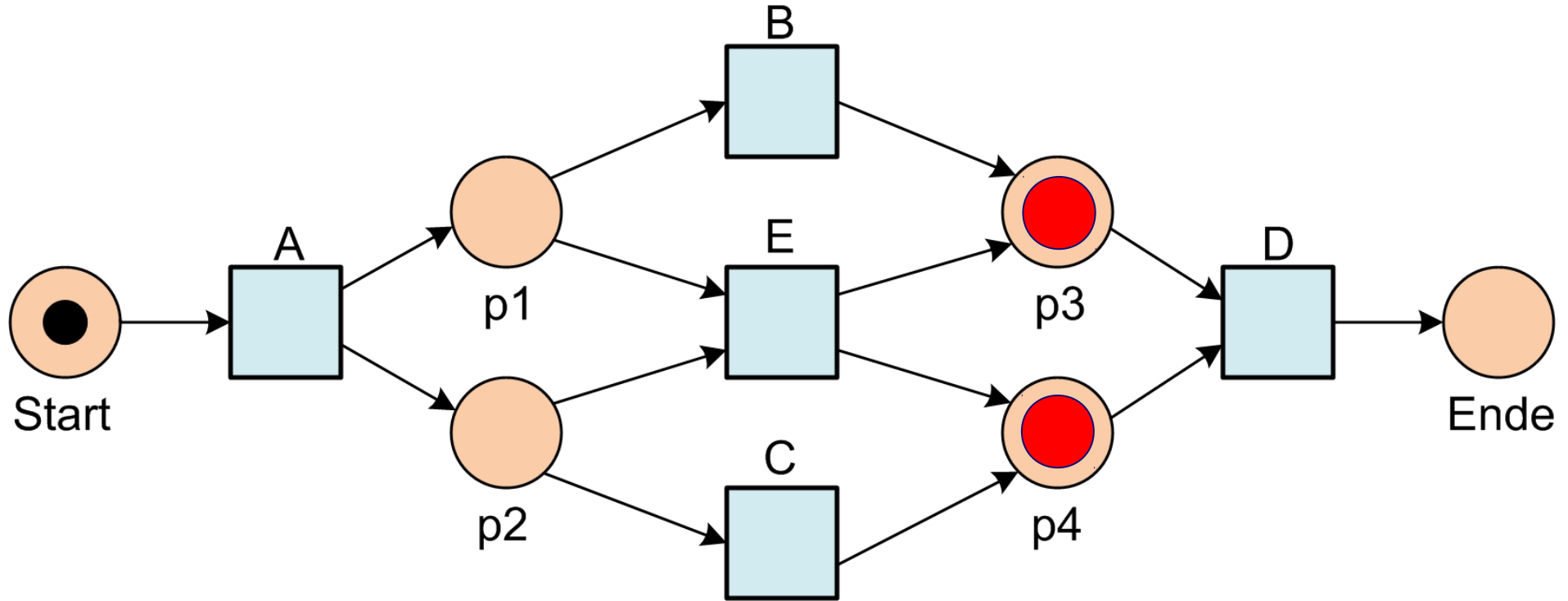
A B

Play-Out: Beispiel



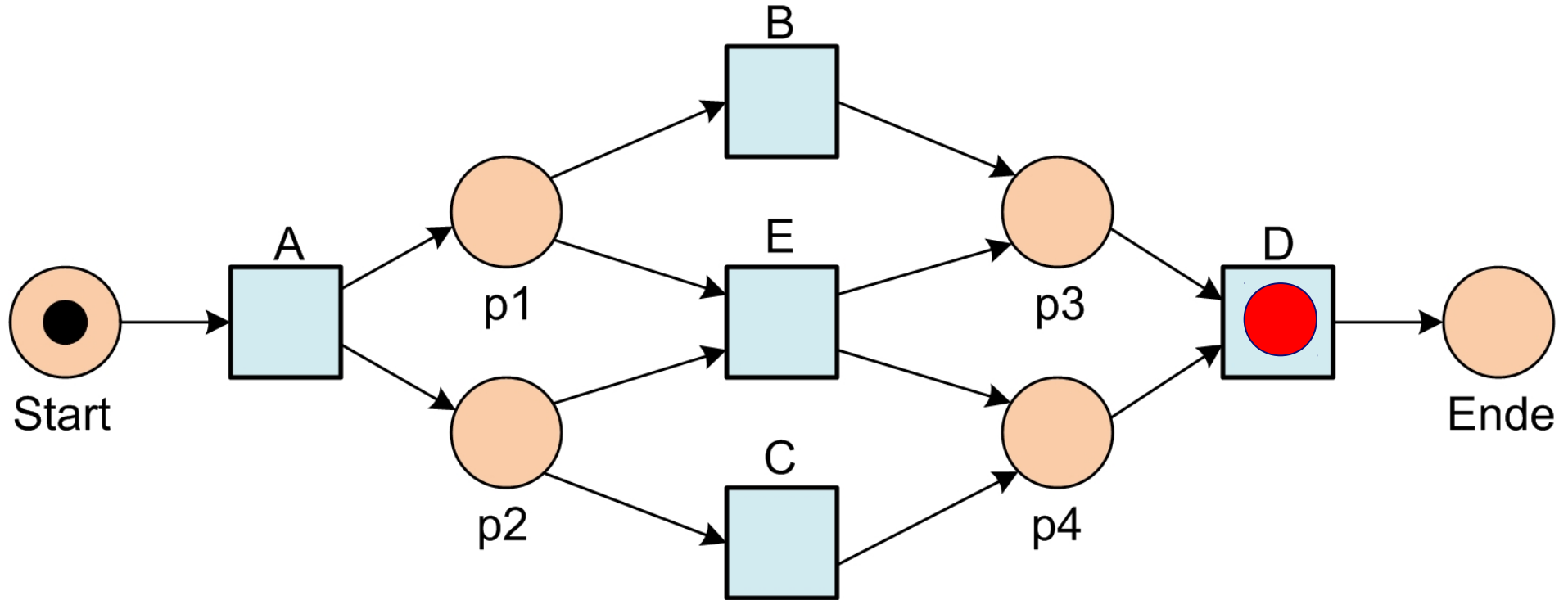
A B C

Play-Out: Beispiel



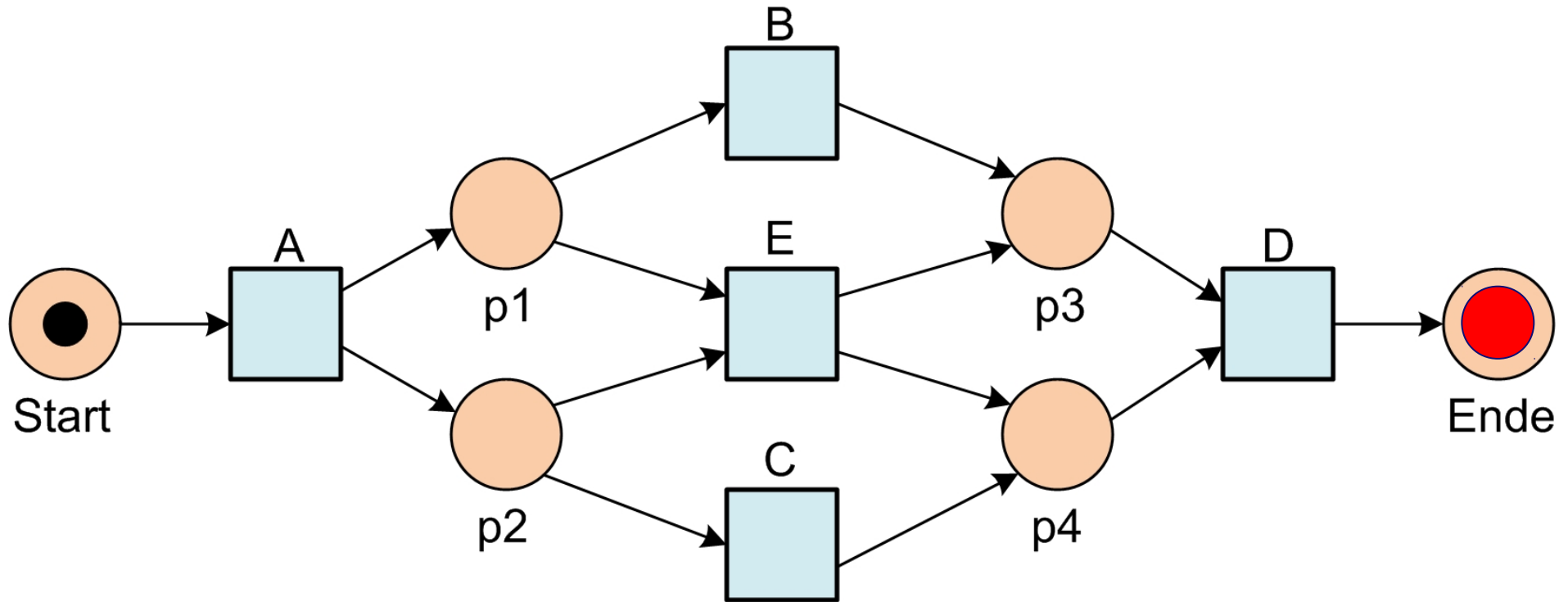
A B C

Play-Out: Beispiel



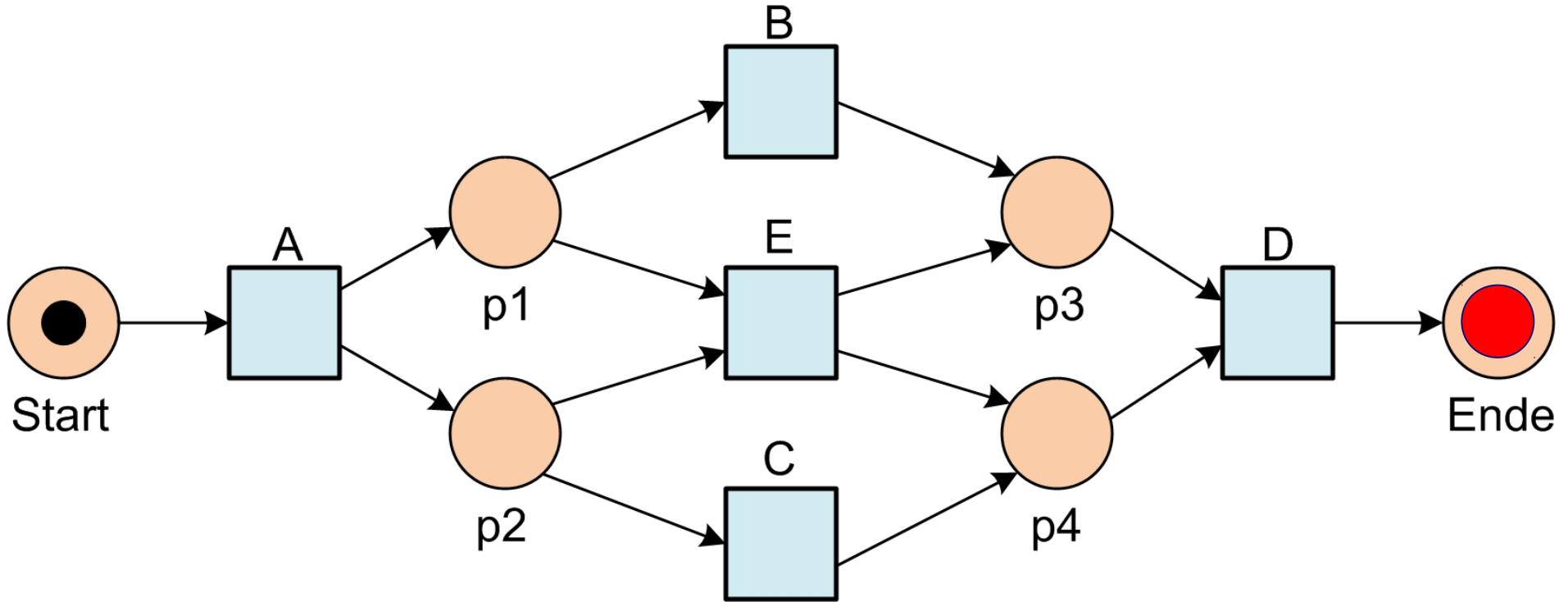
A B C D

Play-Out: Beispiel



A B C D

Play-Out: Beispiel



A B C D

A E D

A E D

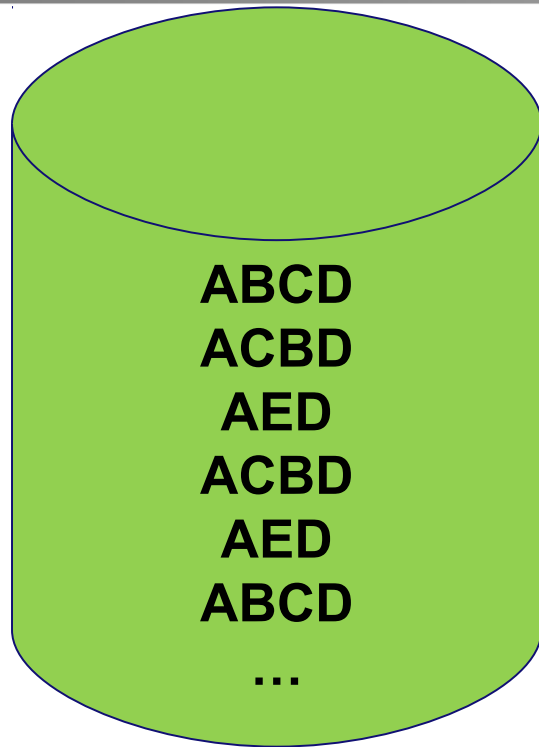
A B C D

A C B D

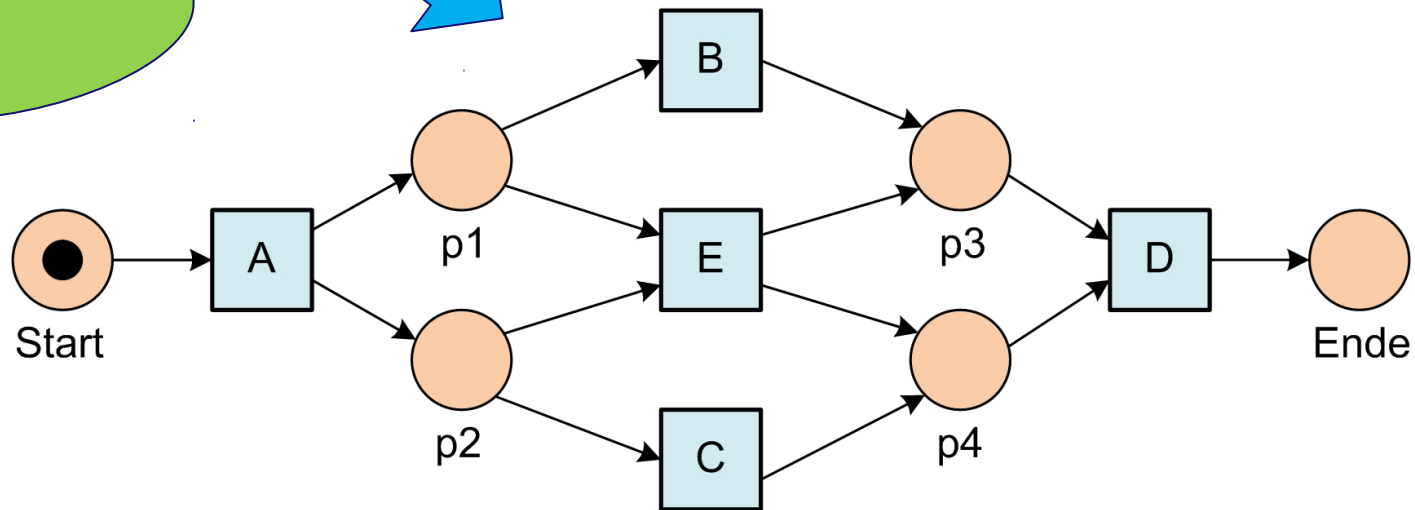
A C B D

A E D

A C B D



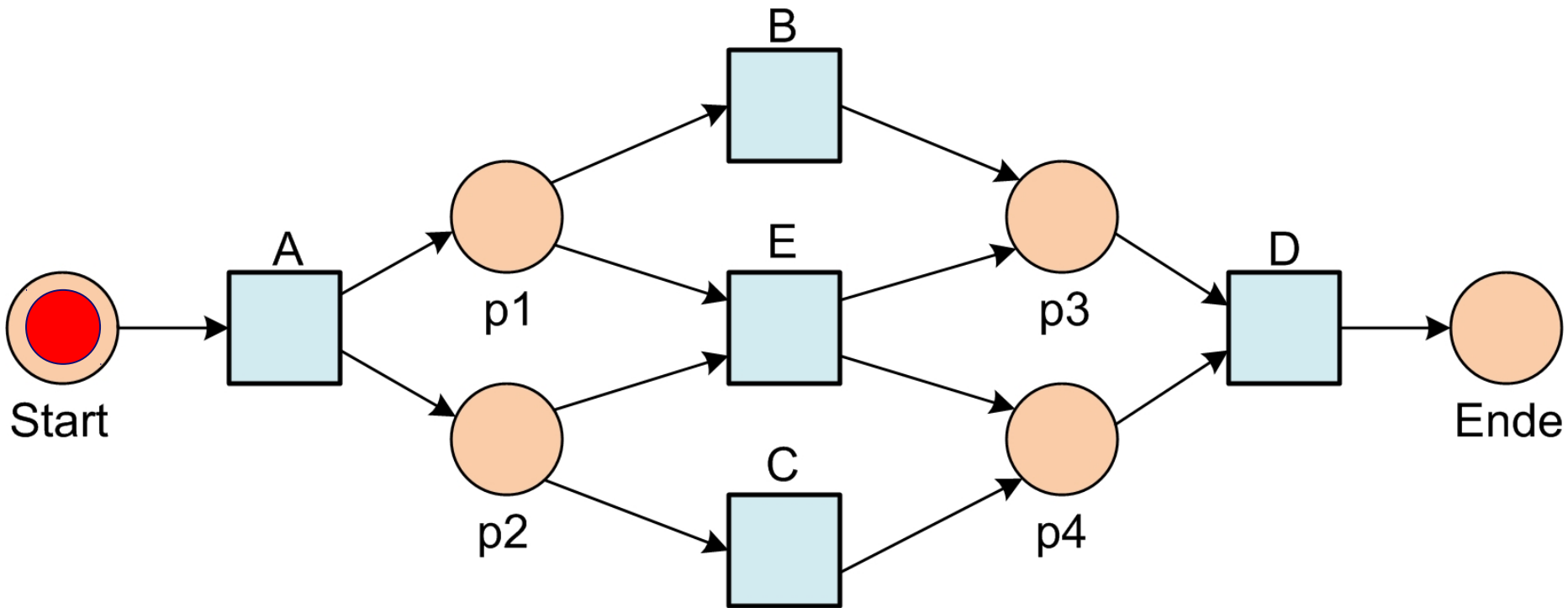
**Prozessextraktions-
Algorithmus** wie z.B.:
 α -Algorithmus



Konformanzanalyse durch Replay: Beispiel



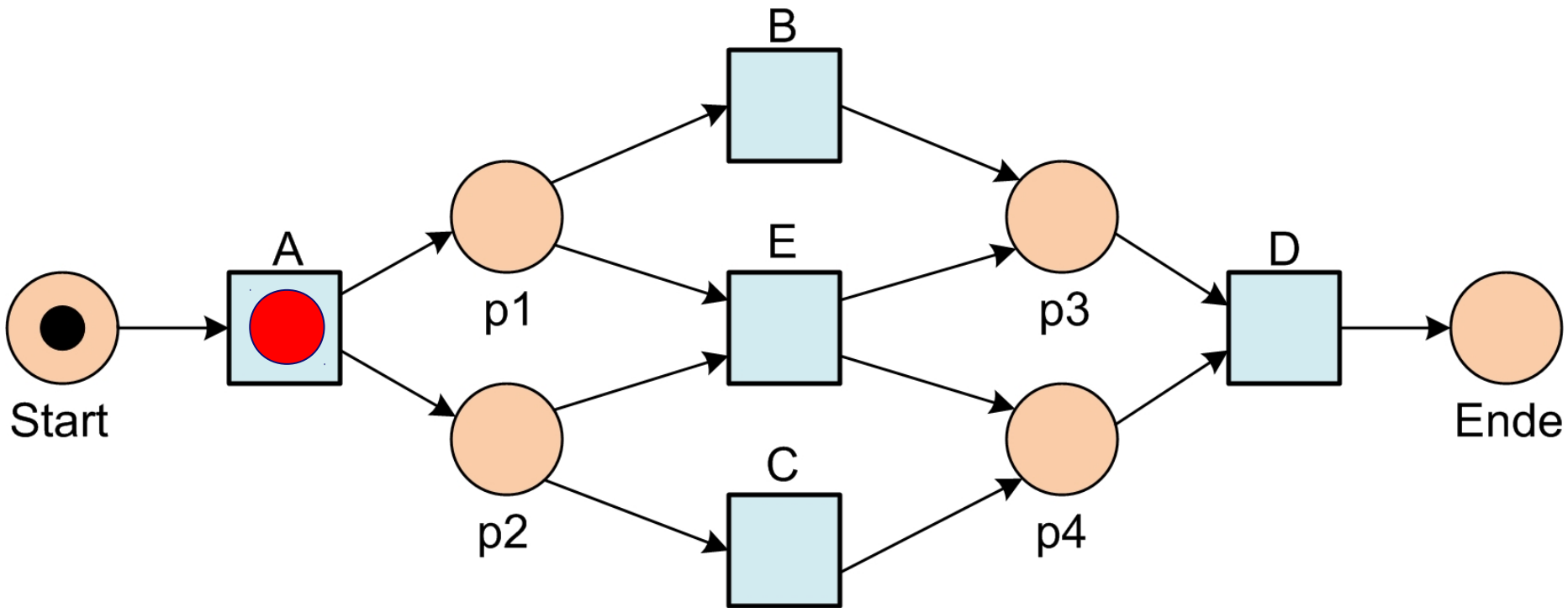
A B C D



Konformanzanalyse durch Replay: Beispiel



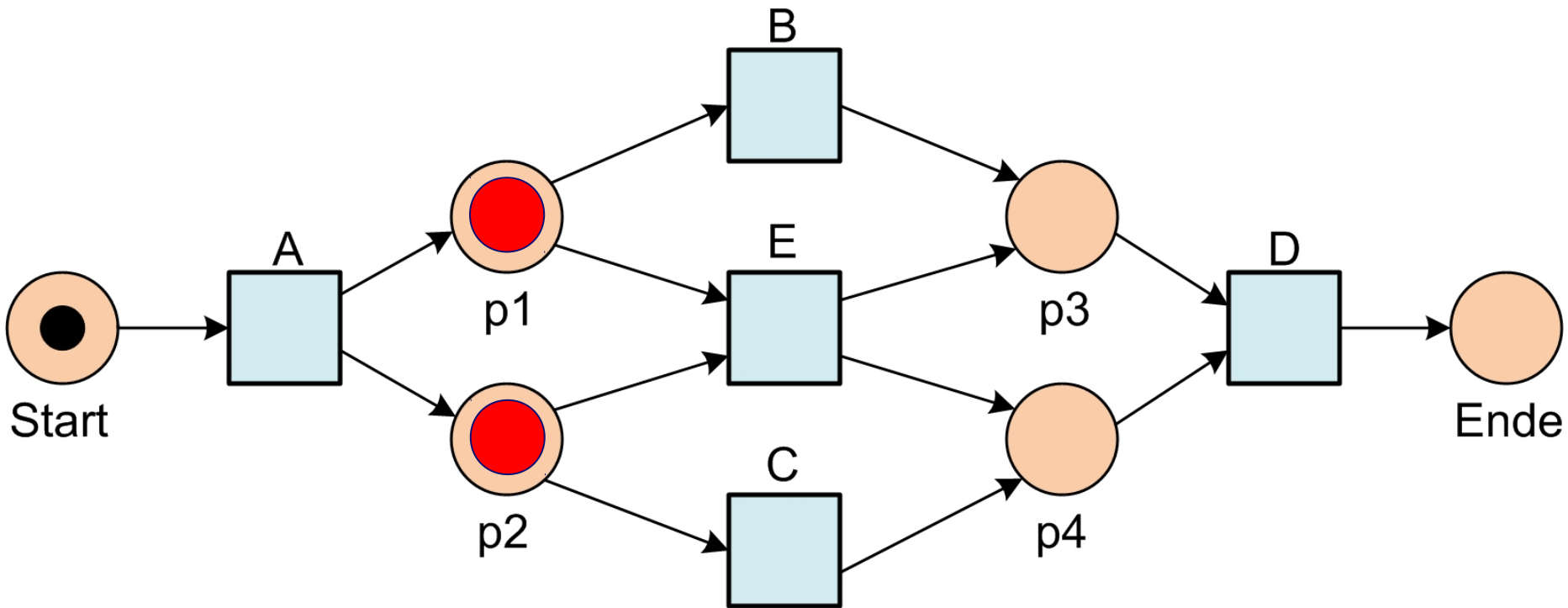
B C D



Konformanzanalyse durch Replay: Beispiel



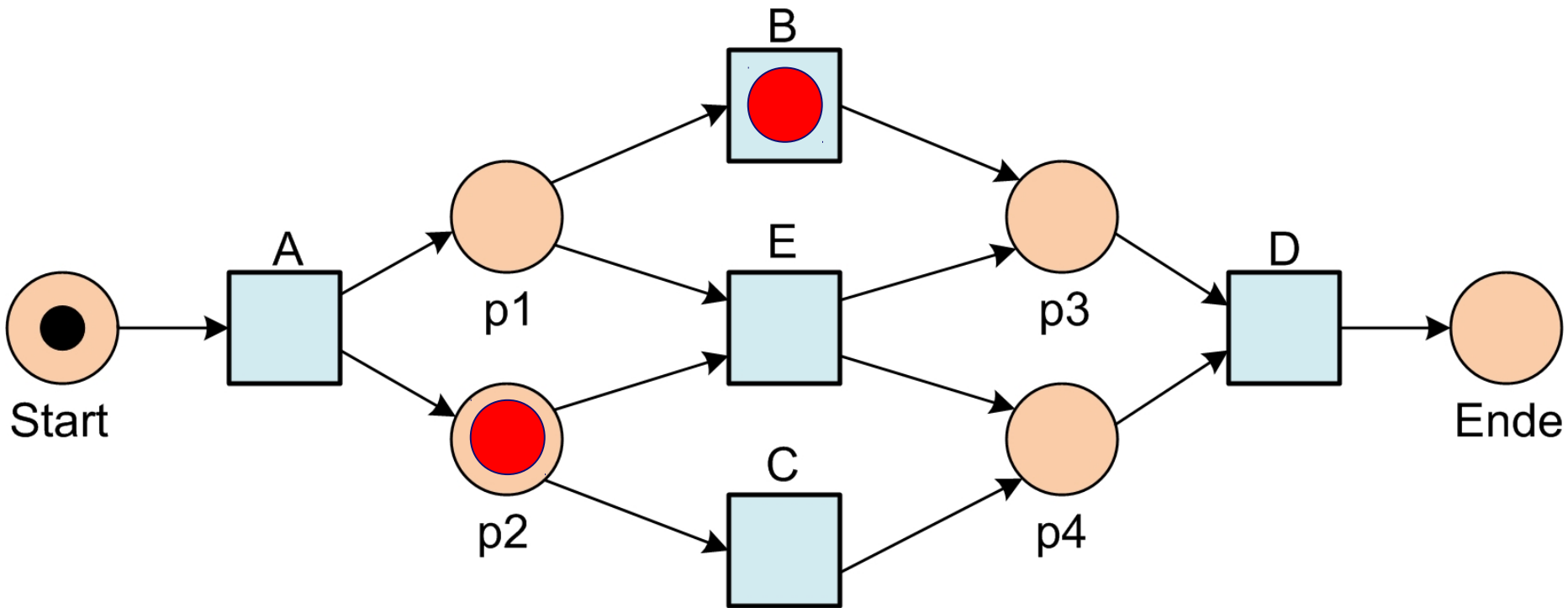
B C D



Konformanzanalyse durch Replay: Beispiel



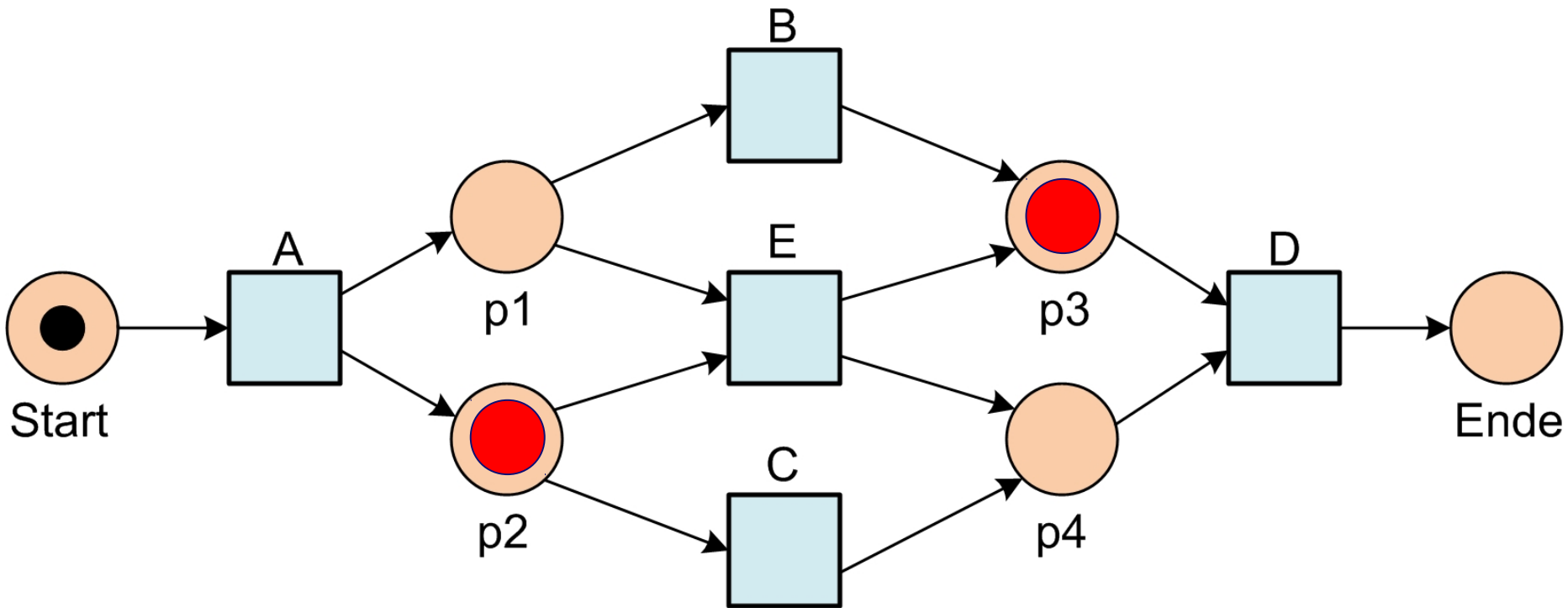
C D



Konformanzanalyse durch Replay: Beispiel



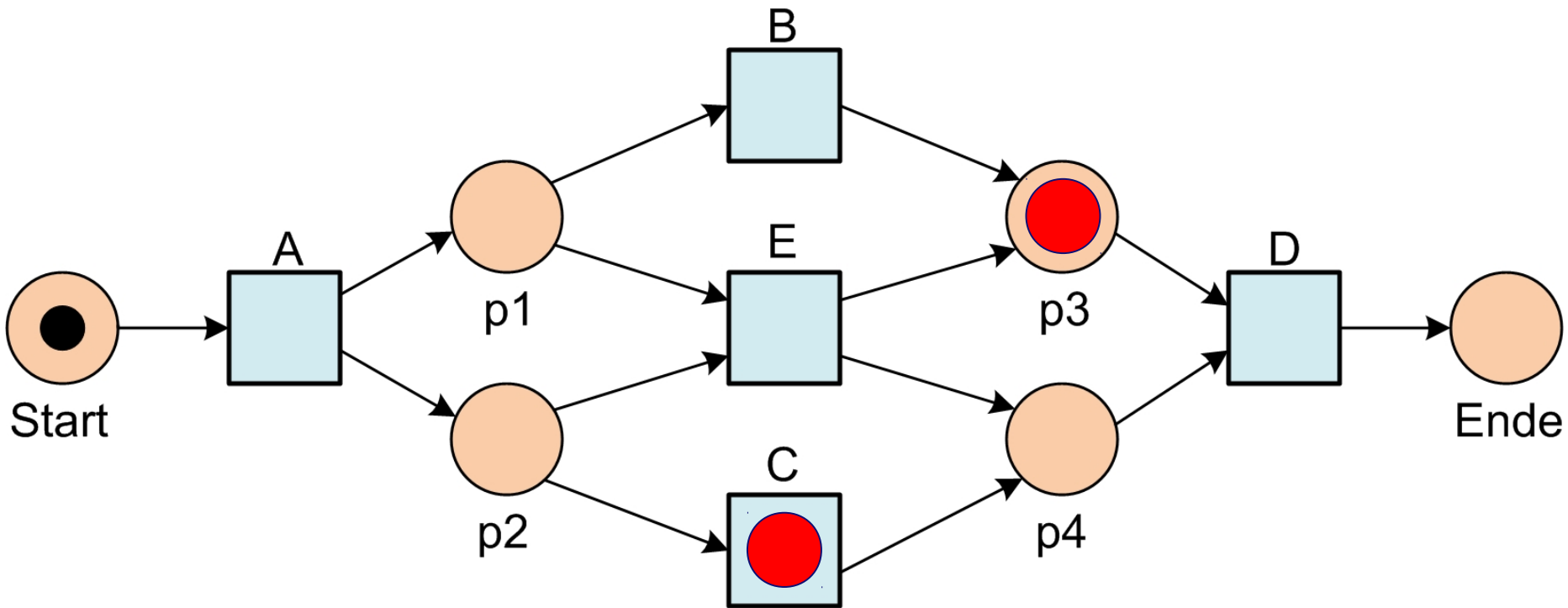
C D



Konformanzanalyse durch Replay: Beispiel



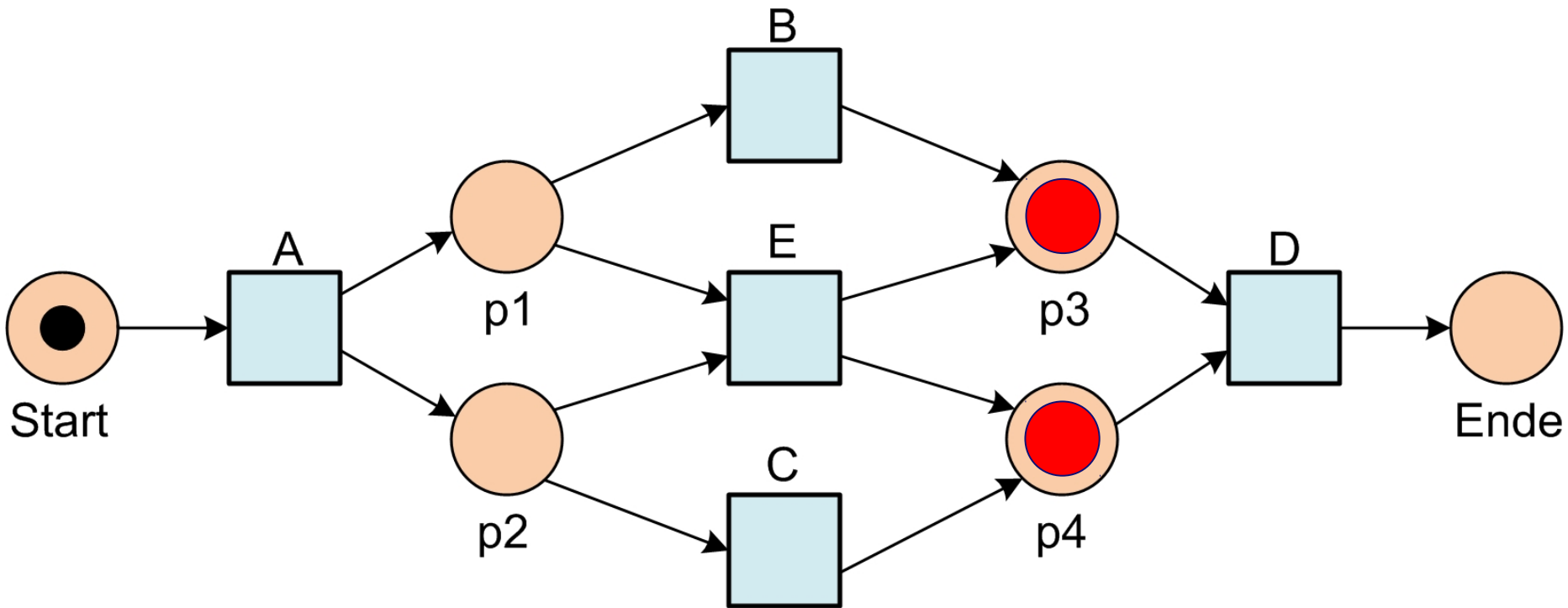
D



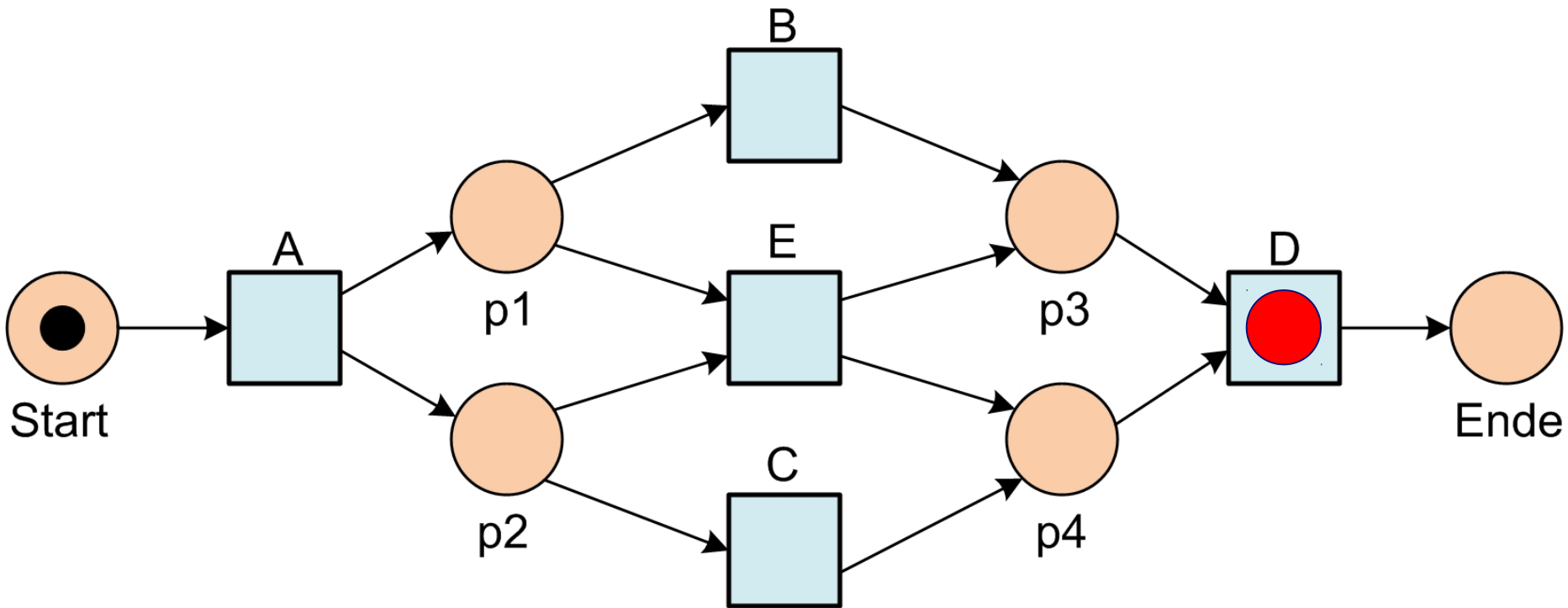
Konformanzanalyse durch Replay: Beispiel



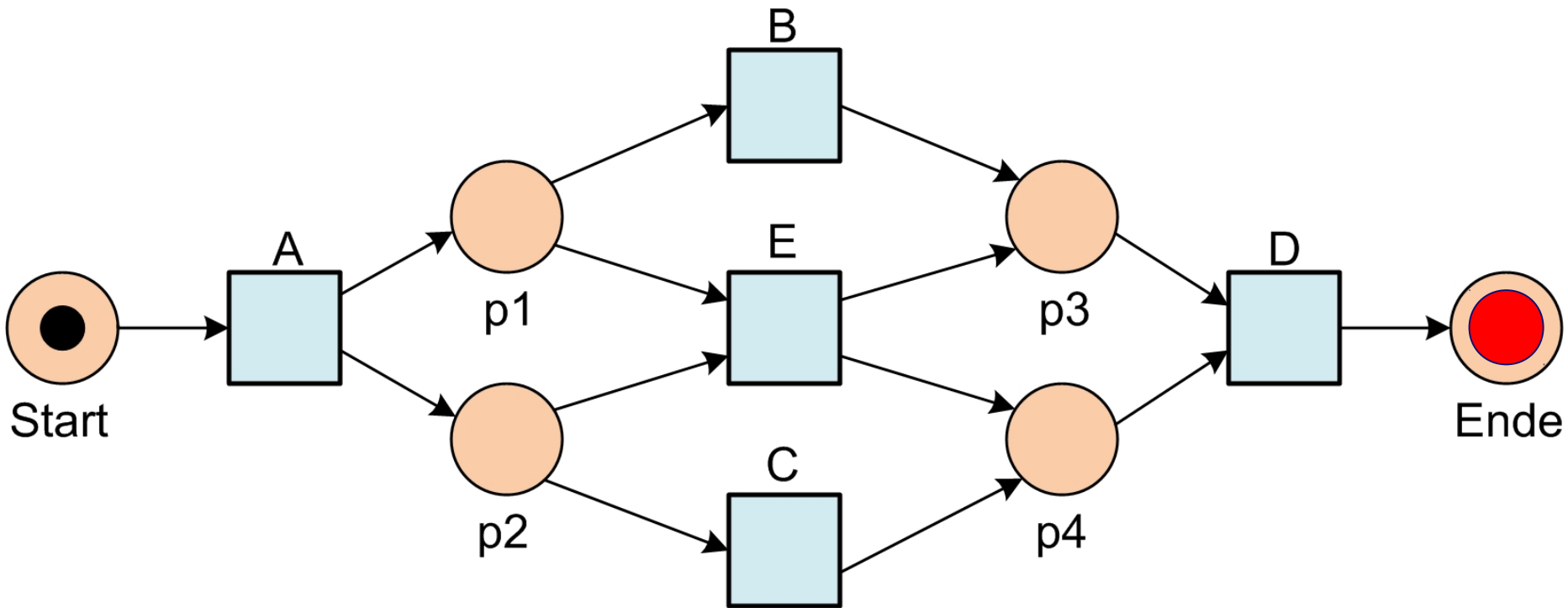
D



Konformanzanalyse durch Replay: Beispiel

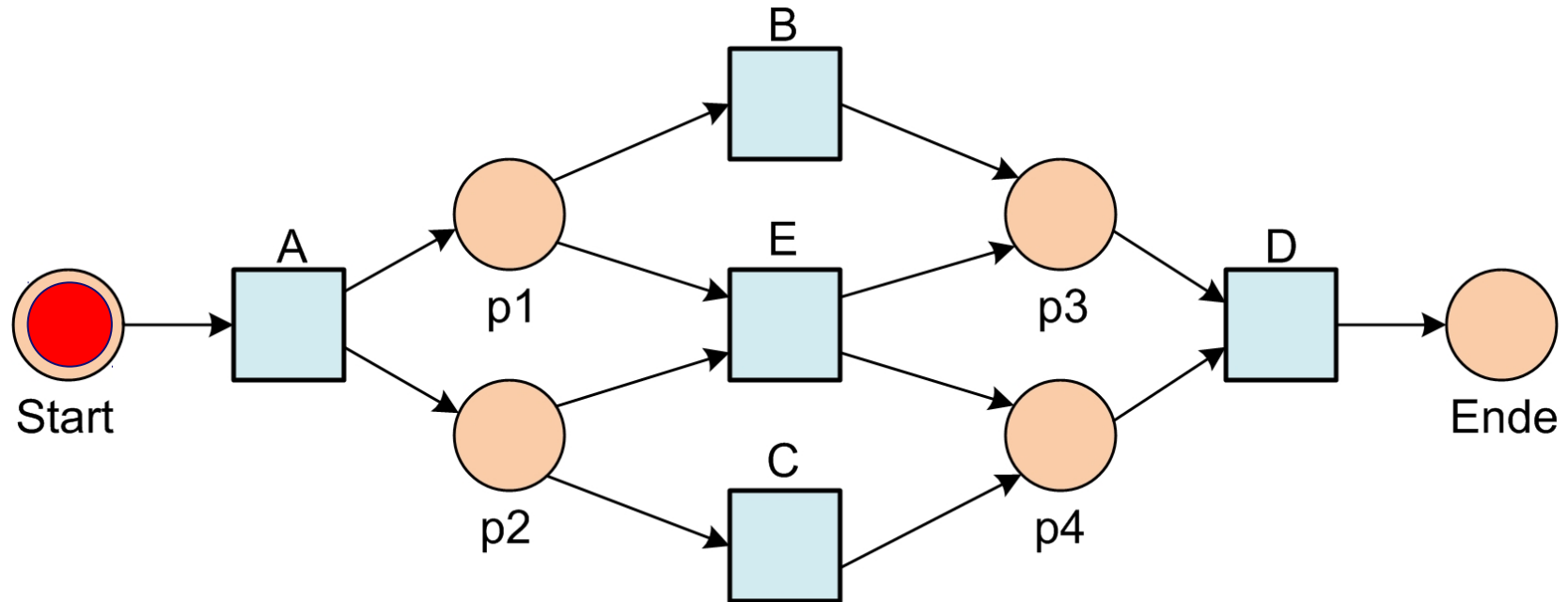


Konformanzanalyse durch Replay: Beispiel



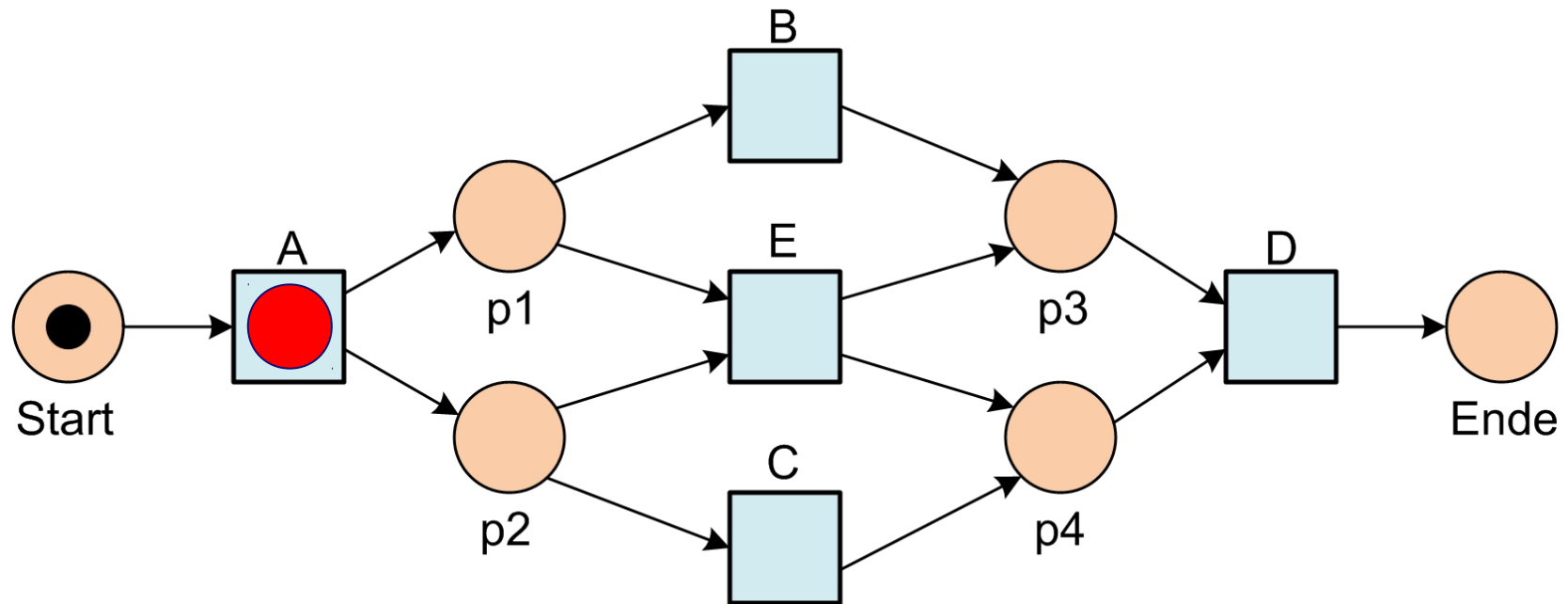
Replay: Mangelnde Konformanz aufdecken

A C D



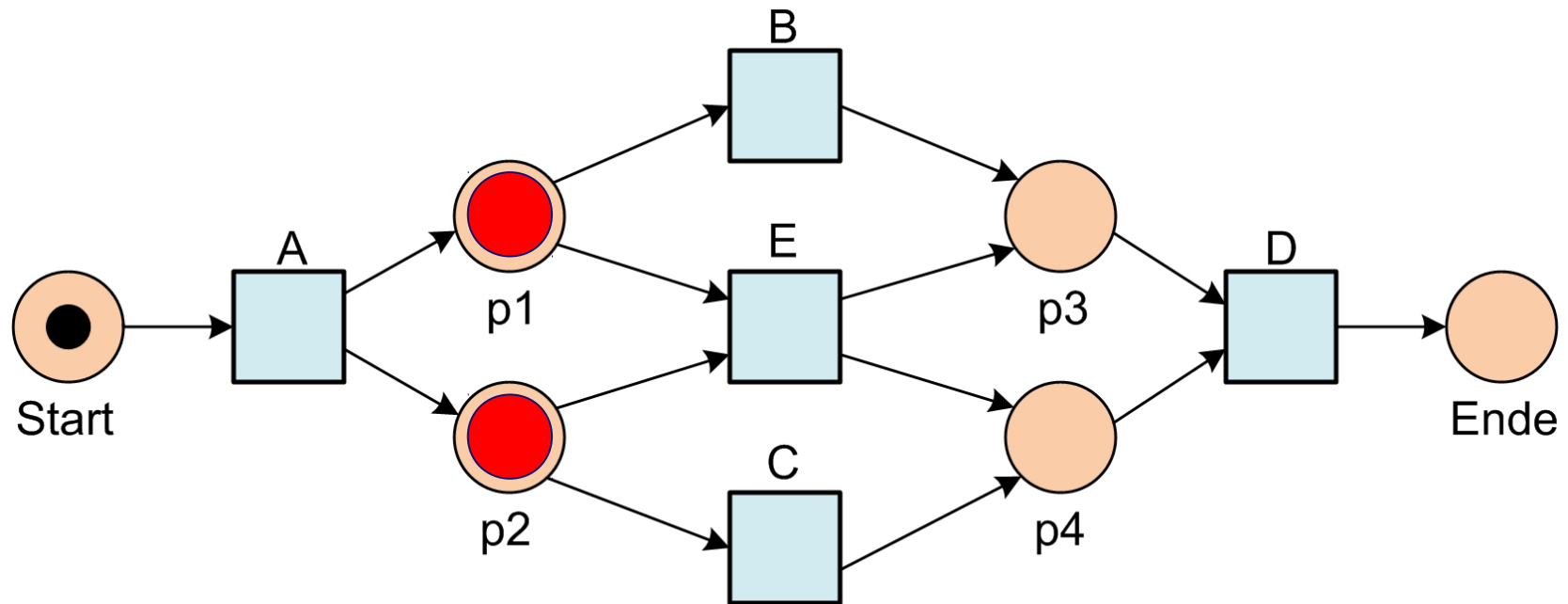
Replay: Mangelnde Konformanz aufdecken

C D



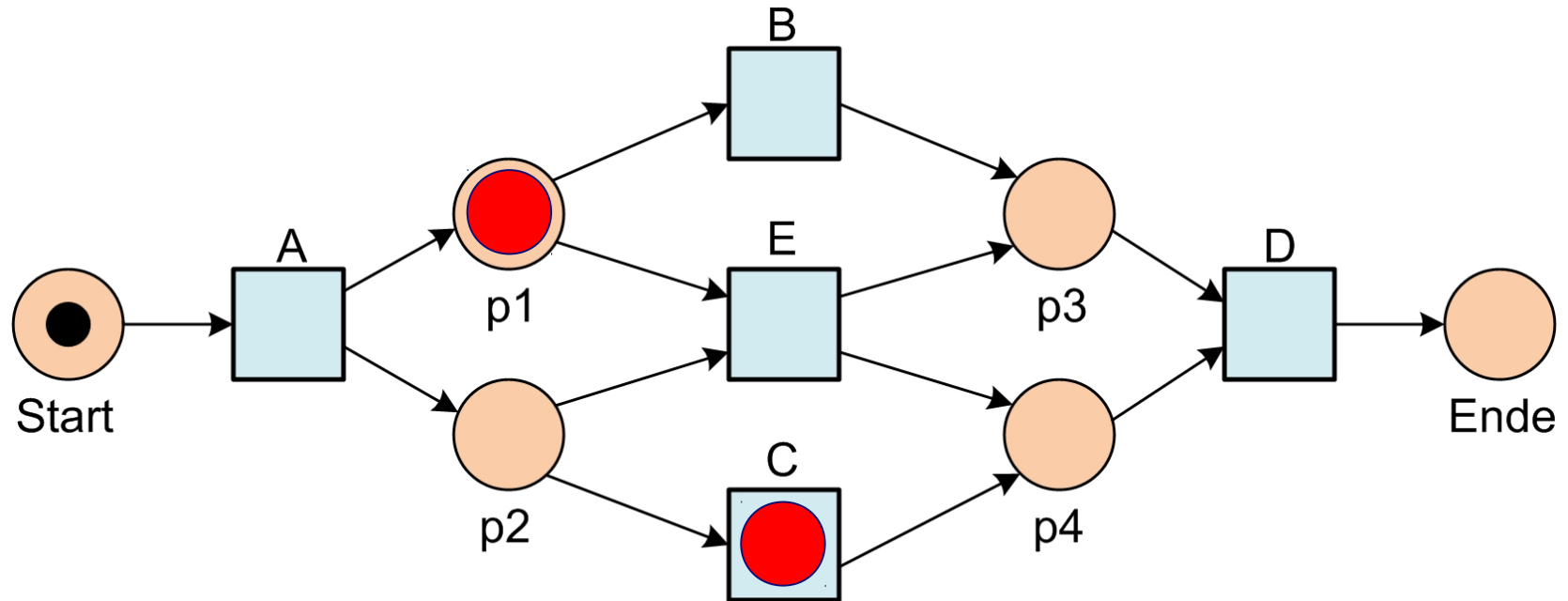
Replay: Mangelnde Konformanz aufdecken

C D



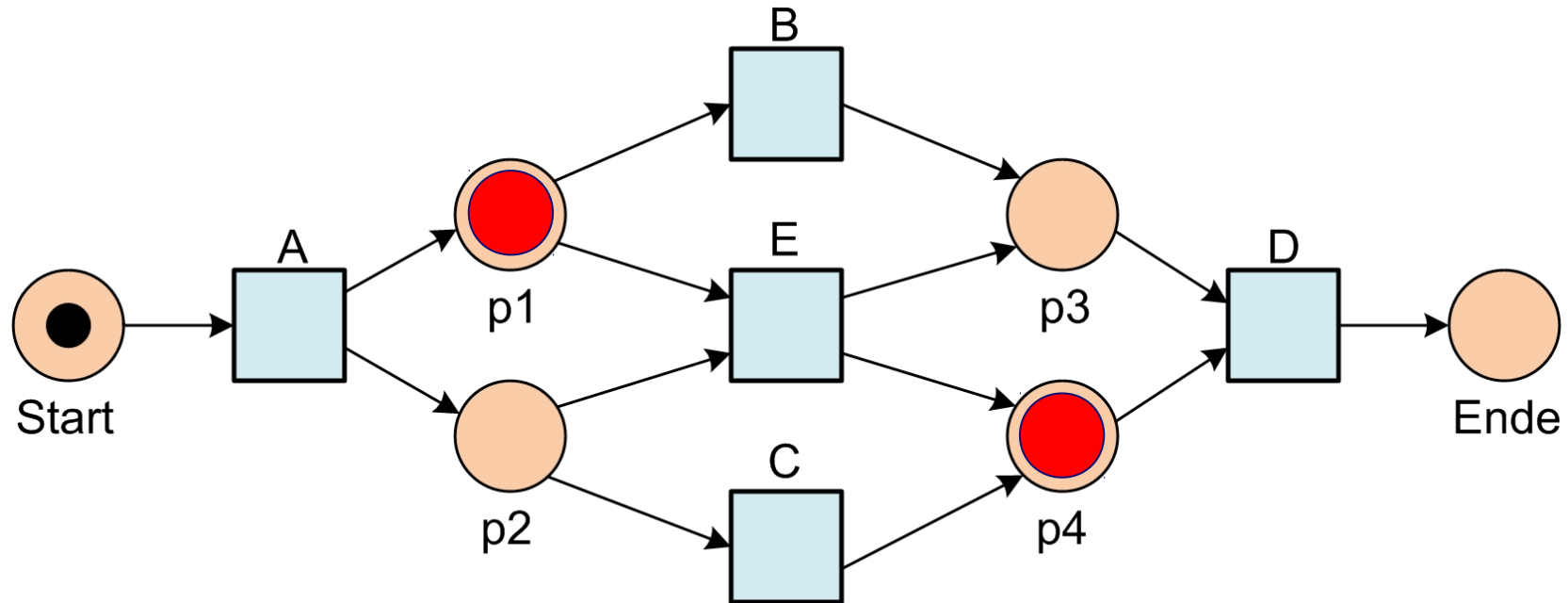
Replay: Mangelnde Konformanz aufdecken

D



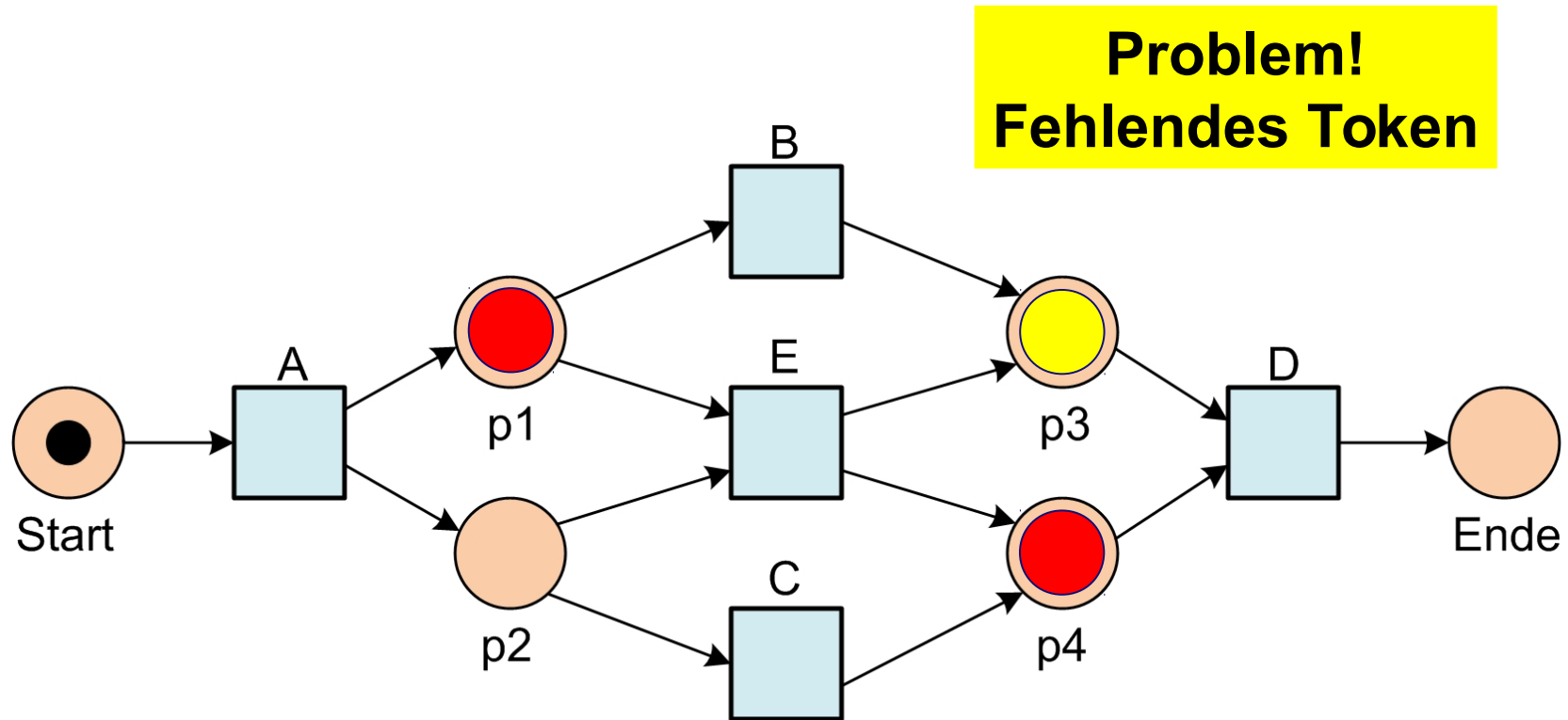
Replay: Mangelnde Konformanz aufdecken

D

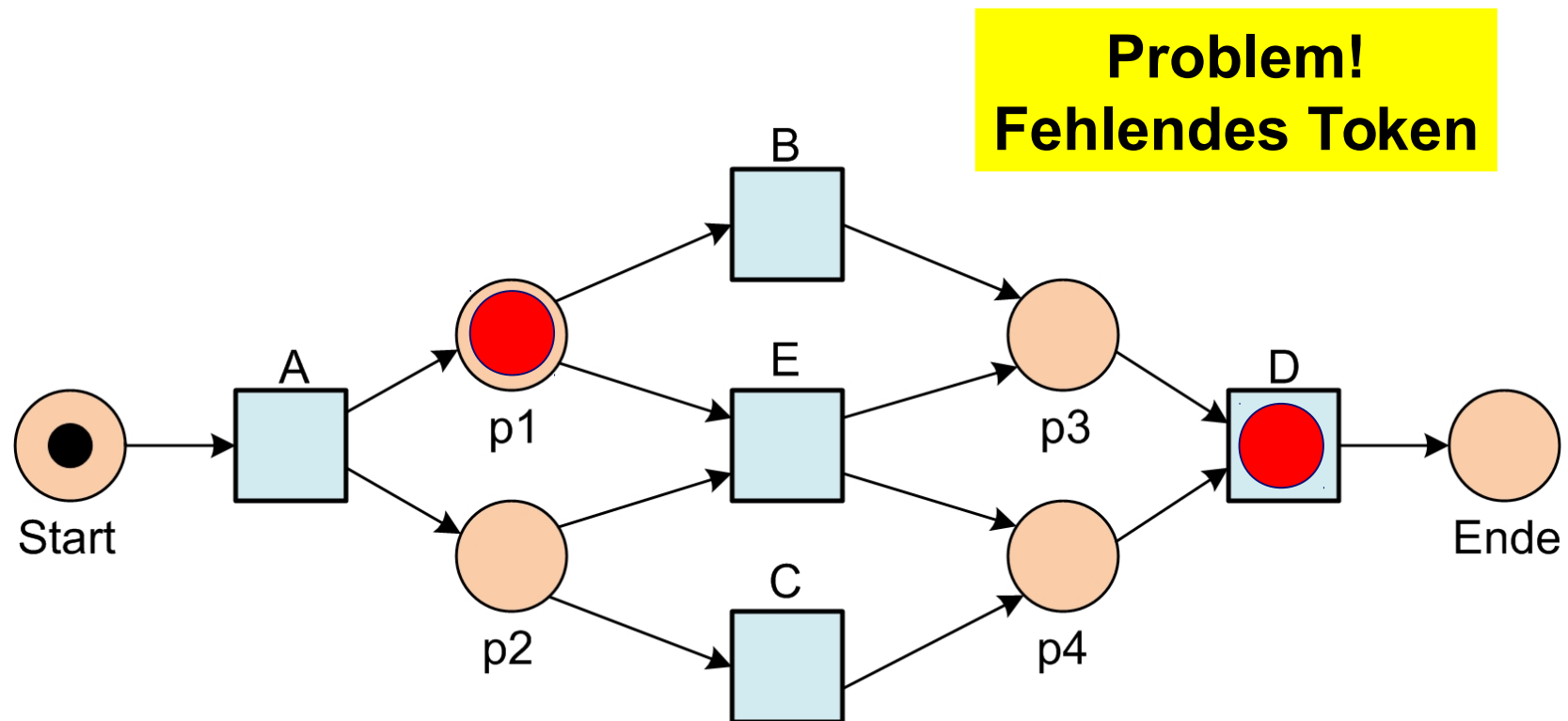


Replay: Mangelnde Konformanz aufdecken

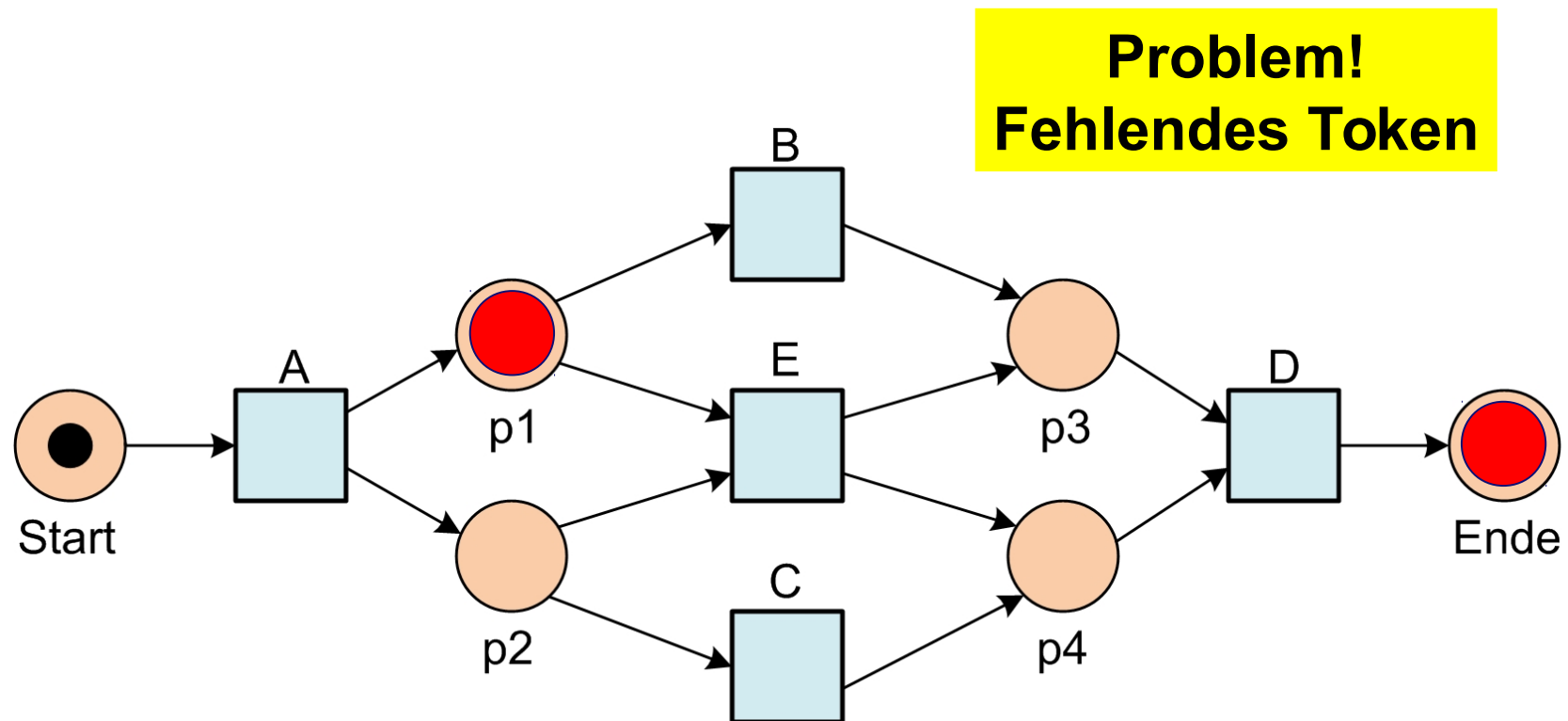
D



Replay: Mangelnde Konformanz aufdecken



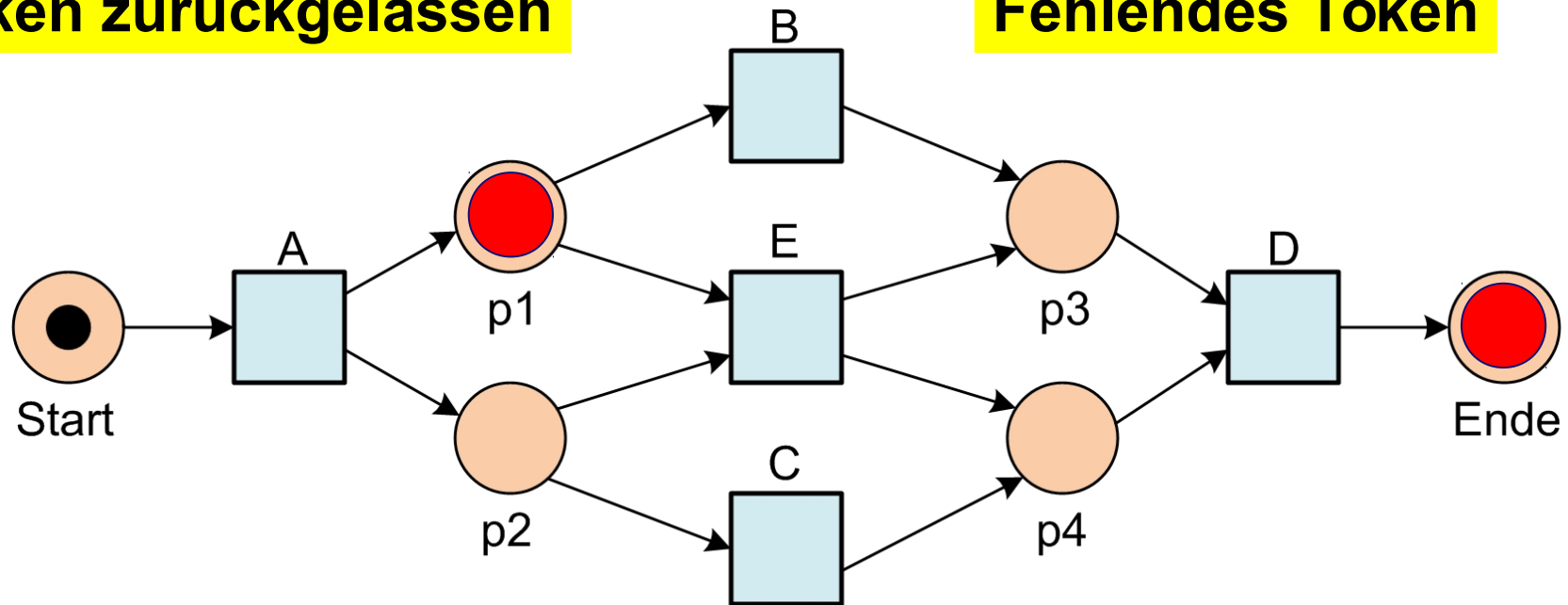
Replay: Mangelnde Konformanz aufdecken



Replay: Mangelnde Konformanz aufdecken

Problem!
Token zurückgelassen

Problem!
Fehlendes Token

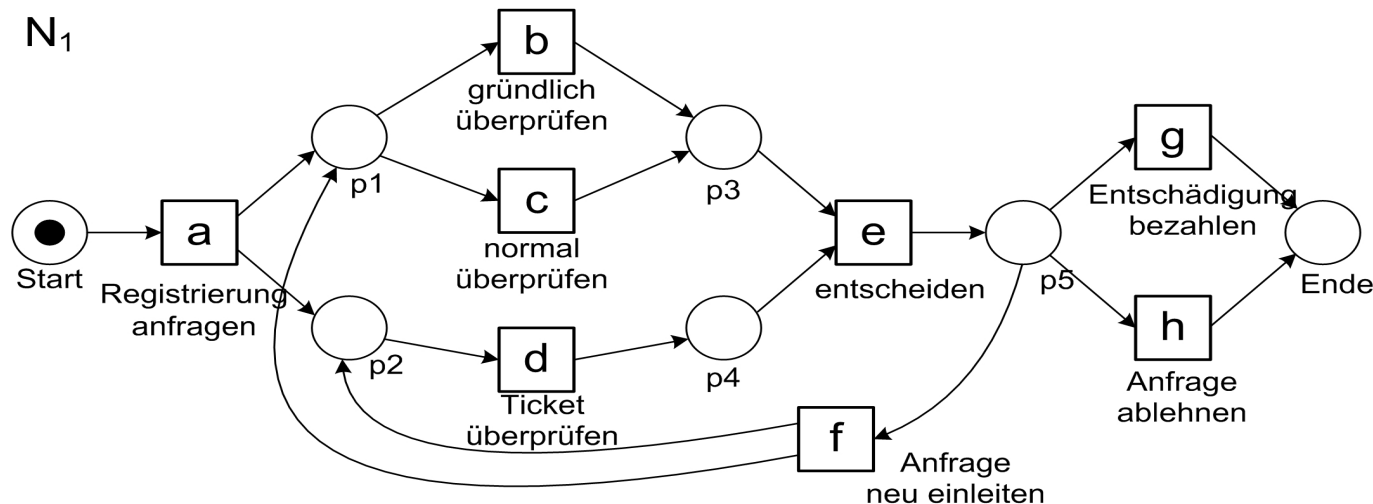


Überblick Konformanzanalyse

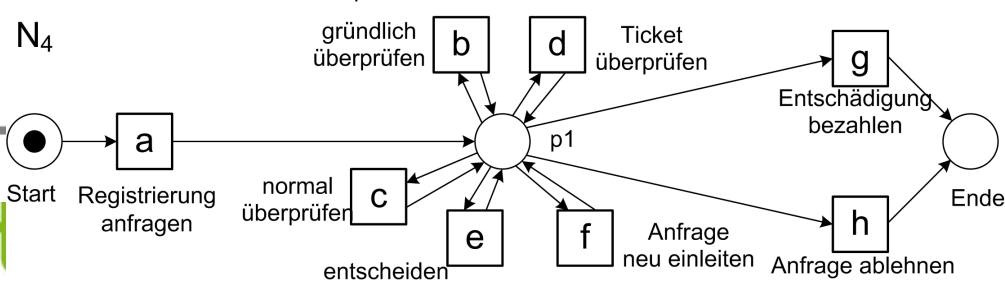
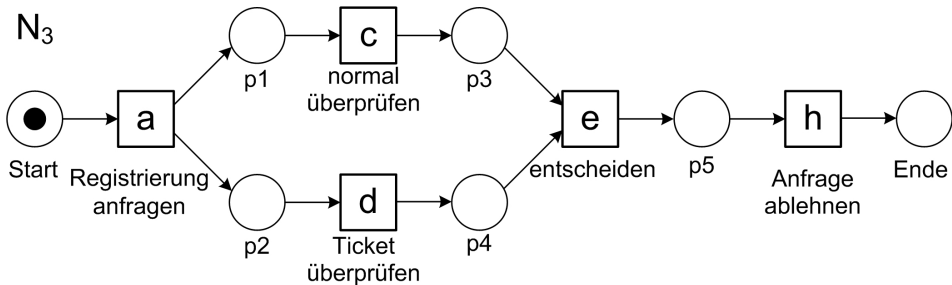
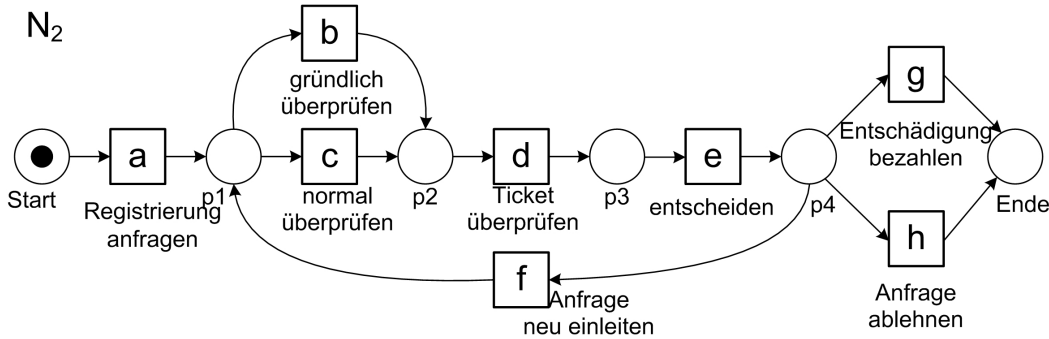
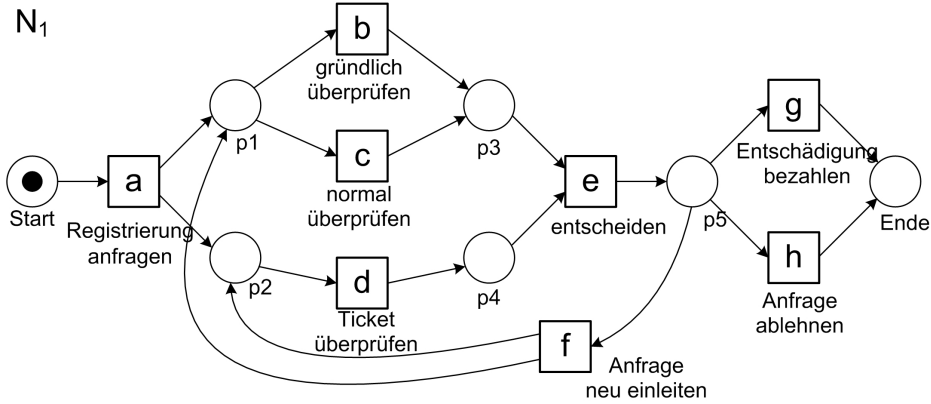
- Einführung
- **Konformanzanalyse basierend auf Replay**
 - Motivation
 - Beispiel
 - **Konformanzmaße**
- Diagnose

Konformanz messen: Abweichungen zählen

- In jedem Schritt gibt es **Zähler**:
 - **p** (produzierte Tokens), **c** (konsumierte Tokens), **m** (fehlende Tokens), **r** (überbleibende Tokens).
- Am Anfang: **alle leer**.
- Umgebung **produziert ein Token** für Stelle *Start* → $p = 1$.
- Transition **a konsumiert ein Token** und produziert 2 Tokens → $p = 3$, $c = 1$.
- Am *Ende* **konsumiert die Umgebung ein Token** → c inkrementieren.



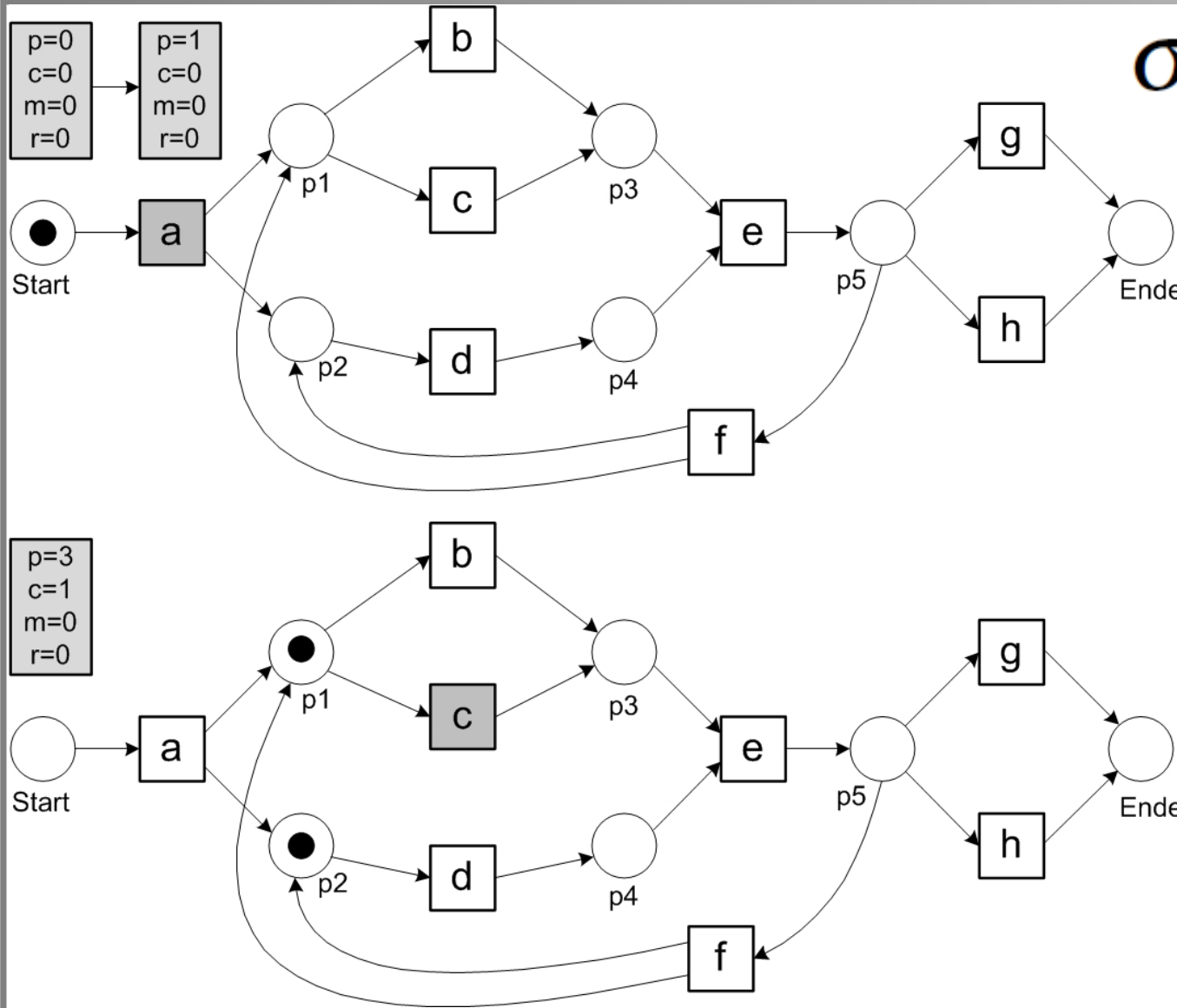
Beispiel: Vier Modelle, ein Log



frequency reference trace

455	σ_1	$\langle a, c, d, e, h \rangle$
191	σ_2	$\langle a, b, d, e, g \rangle$
177	σ_3	$\langle a, d, c, e, h \rangle$
144	σ_4	$\langle a, b, d, e, h \rangle$
111	σ_5	$\langle a, c, d, e, g \rangle$
82	σ_6	$\langle a, d, c, e, g \rangle$
56	σ_7	$\langle a, d, b, e, h \rangle$
47	σ_8	$\langle a, c, d, e, f, d, b, e, h \rangle$
38	σ_9	$\langle a, d, b, e, g \rangle$
33	σ_{10}	$\langle a, c, d, e, f, b, d, e, h \rangle$
14	σ_{11}	$\langle a, c, d, e, f, b, d, e, g \rangle$
11	σ_{12}	$\langle a, c, d, e, f, d, b, e, g \rangle$
9	σ_{13}	$\langle a, d, c, e, f, c, d, e, h \rangle$
8	σ_{14}	$\langle a, d, c, e, f, d, b, e, h \rangle$
5	σ_{15}	$\langle a, d, c, e, f, b, d, e, g \rangle$
3	σ_{16}	$\langle a, c, d, e, f, b, d, e, f, d, b, e, g \rangle$
2	σ_{17}	$\langle a, d, c, e, f, d, b, e, g \rangle$
2	σ_{18}	$\langle a, d, c, e, f, b, d, e, f, b, d, e, g \rangle$
1	σ_{19}	$\langle a, d, c, e, f, d, b, e, f, b, d, e, h \rangle$
1	σ_{20}	$\langle a, d, b, e, f, b, d, e, f, d, b, e, g \rangle$
1	σ_{21}	$\langle a, d, c, e, f, d, b, e, f, c, d, e, f, d, b, e, g \rangle$

Replaying σ_1 auf N_1 (1/3)



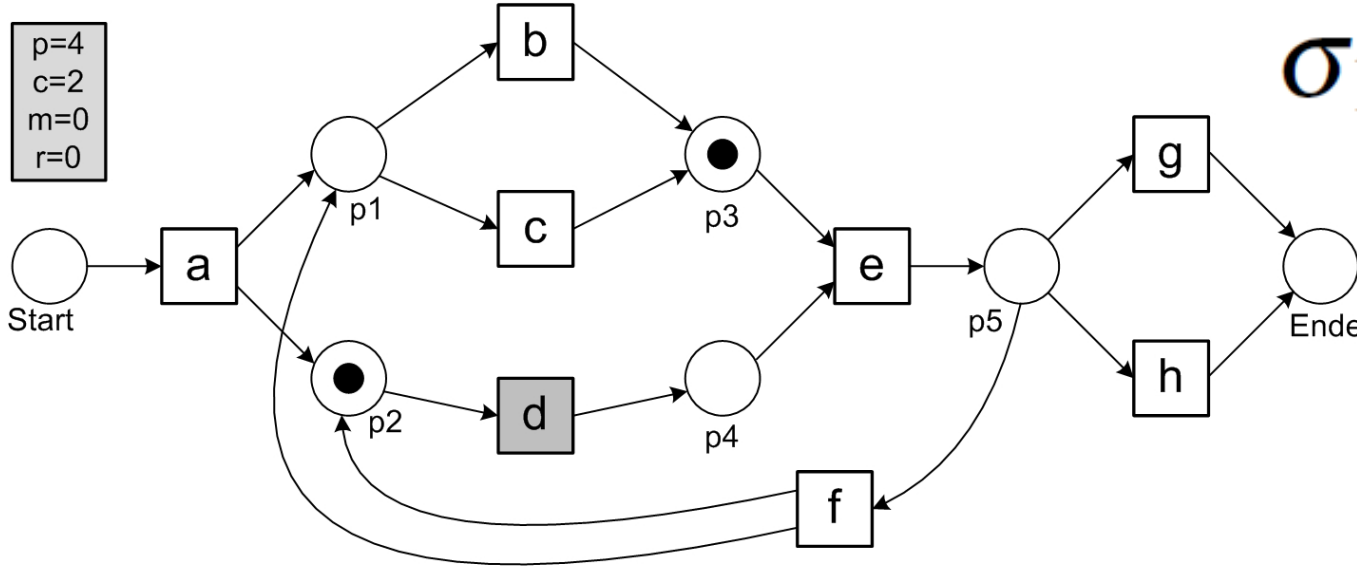
$$\sigma_1 = \langle a, c, d, e, h \rangle$$

- Umwelt produziert 1 Token für Stelle *Start*
- 0 Token konsumiert, **missing** (fehlt) oder **remaining** (übrig)
- a aktiv gemäß Log
- a konsumiert ein Token und produziert 2 Token \rightarrow 3 Token produziert (Start, p1, p2) & 1 Token konsumiert (Start)
- 0 Token **missing** (fehlt) oder **remaining** (übrig)
- c aktiv gemäß Log

Replaying σ_1 auf N_1 (2/3)



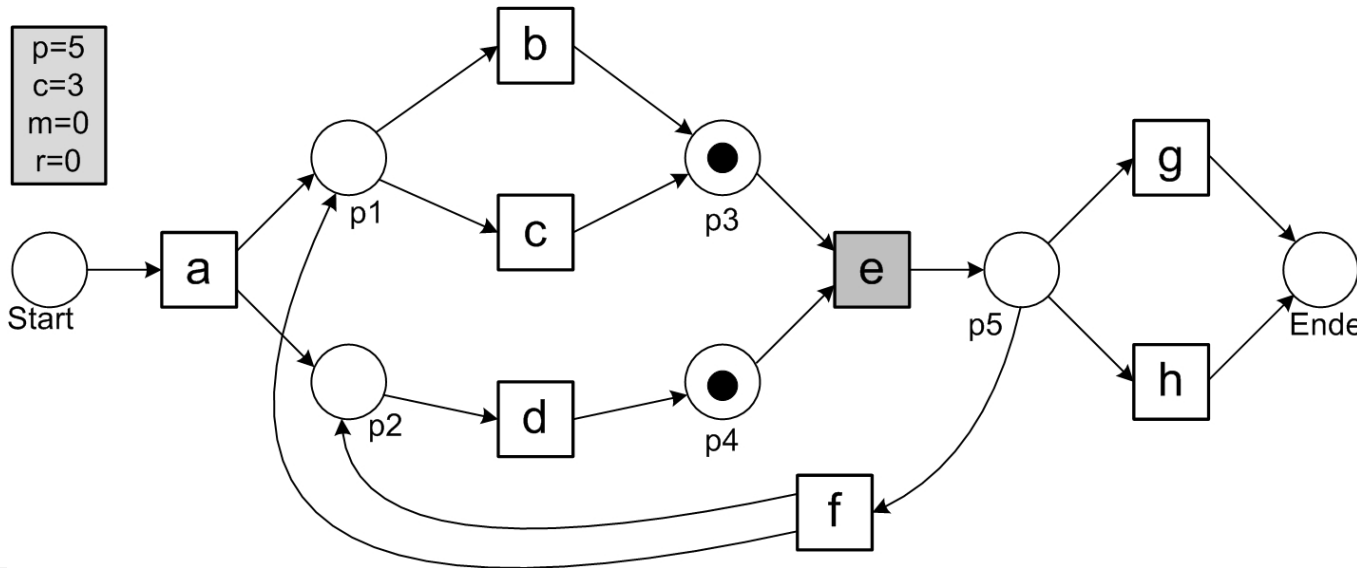
p=4
c=2
m=0
r=0



$$\sigma_1 = \langle a, c, d, e, h \rangle$$

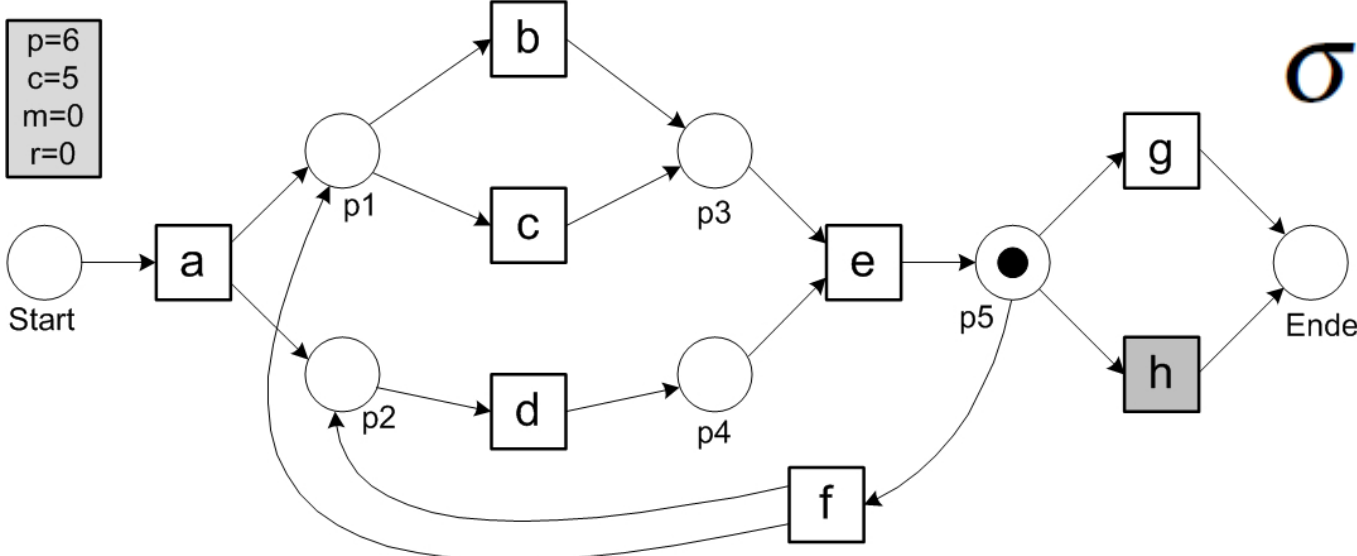
- c konsumiert 1 Token und produziert 1 Token \rightarrow 4 Token produziert & 2 Token konsumiert
- 0 Token missing oder remaining
- d aktiv gemäß Log

p=5
c=3
m=0
r=0



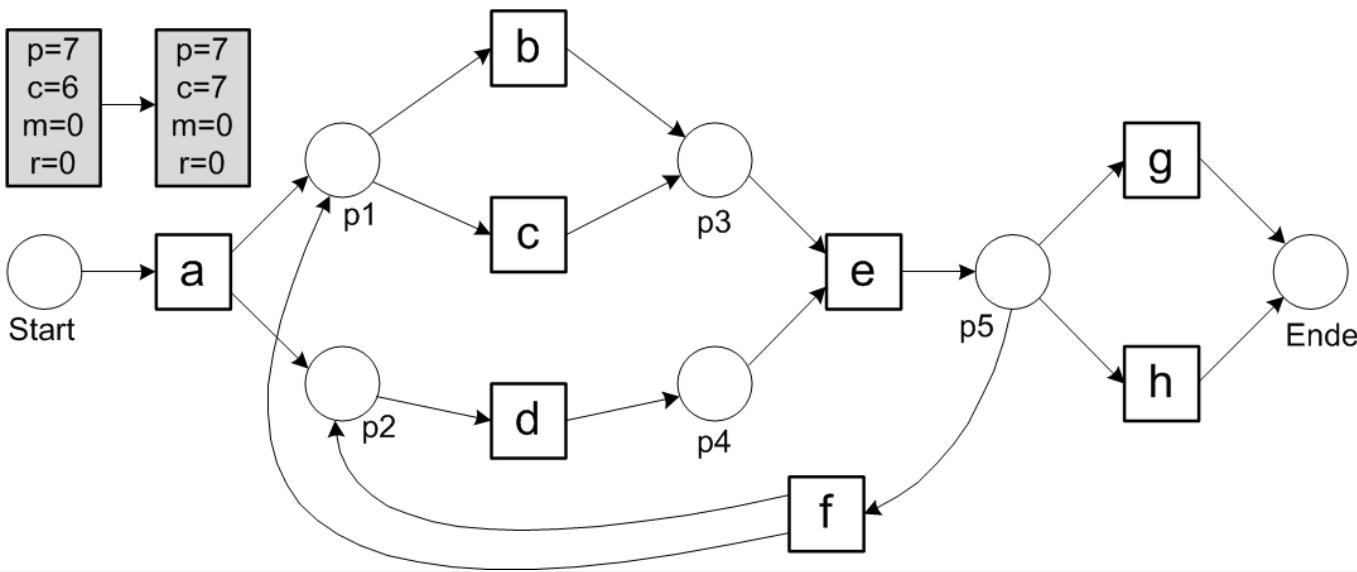
- d konsumiert 1 Token und produziert 1 Token \rightarrow 5 Token produziert & 3 Token konsumiert
- 0 Token missing oder remaining
- e aktiv gemäß Log

Replaying σ_1 auf N_1 (3/3)



$$\sigma_1 = \langle a, c, d, e, h \rangle$$

- e konsumiert 2 Token und produziert 1 Token \rightarrow 6 Token produziert & 5 Token konsumiert
- 0 Token missing oder remaining
- h aktiv gemäß Log

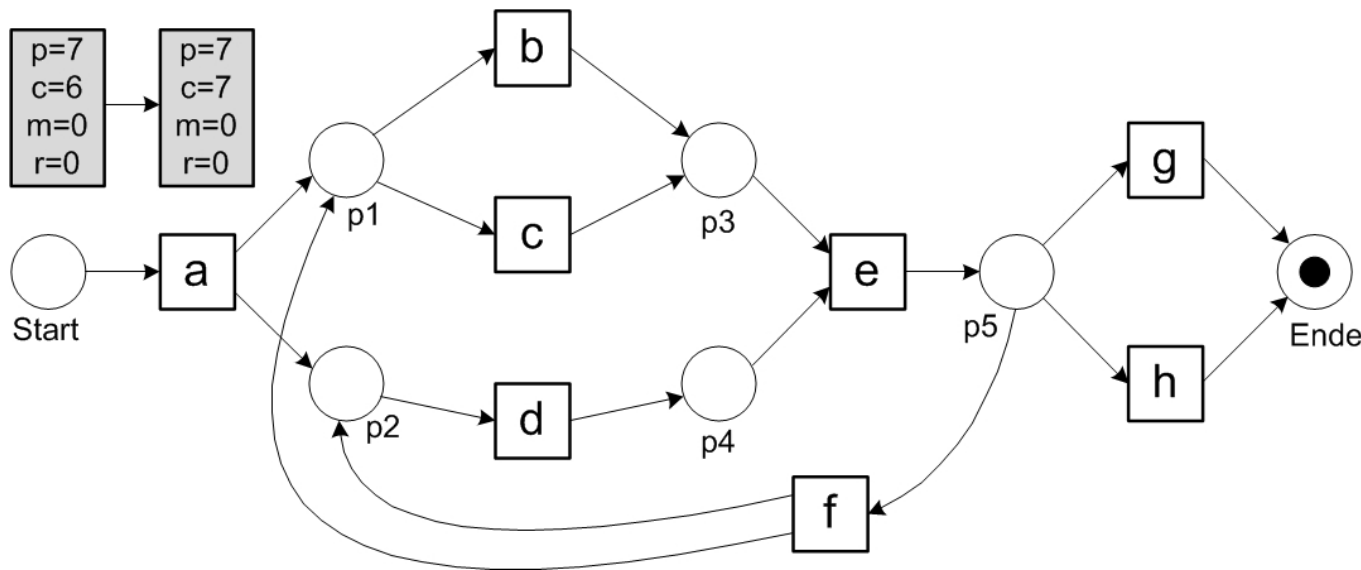


- h konsumiert 1 Token und produziert 1 Token \rightarrow 7 Token produziert & 6 Token konsumiert
- 0 Token missing oder remaining
- h aktiv gemäß Log

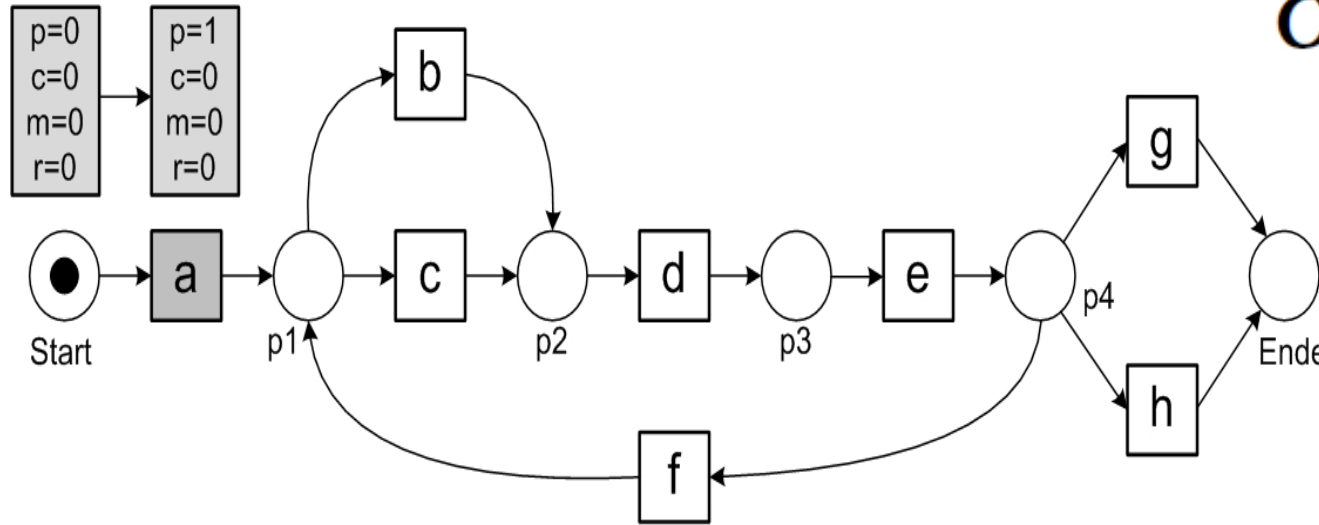
Replaying σ_1 auf N_1 (3/3)

$$\sigma_1 = \langle a, c, d, e, h \rangle$$

- Umwelt konsumiert 1 Token \rightarrow 7 Token produziert & 7 Token konsumiert
- 0 Token missing oder remaining
 \rightarrow **Keine Probleme gefunden**

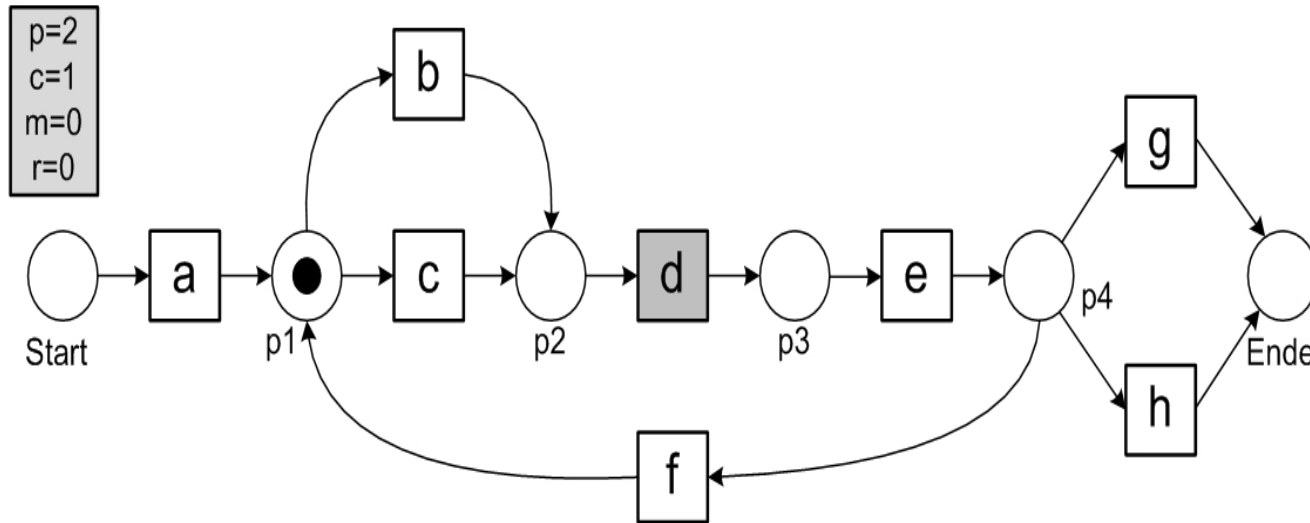


Replaying σ_3 auf N_2 (1/3)



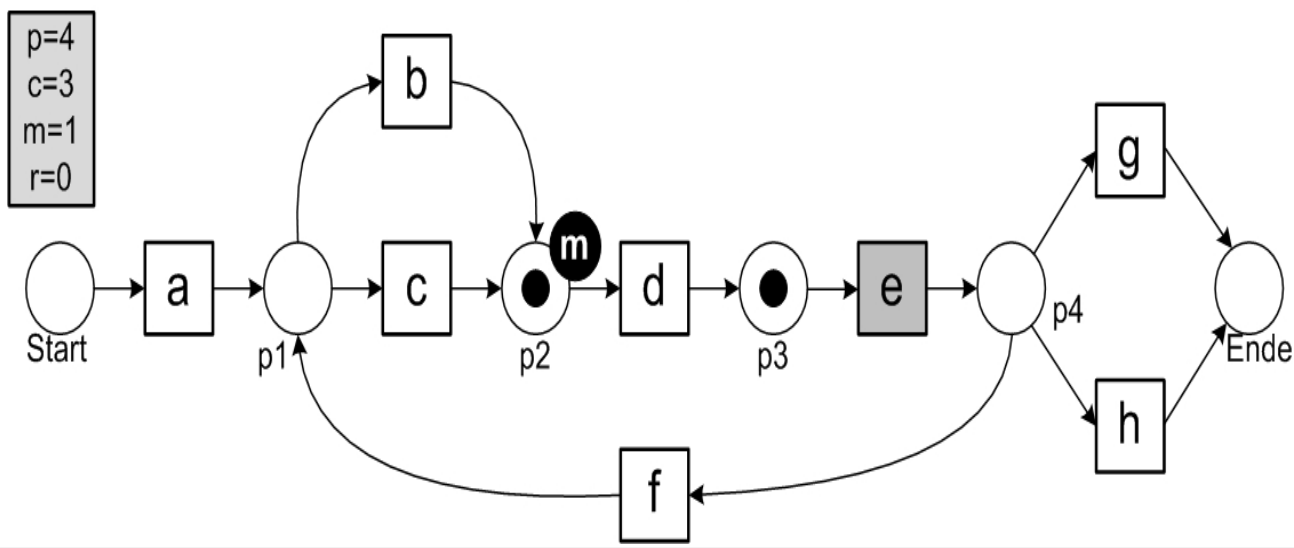
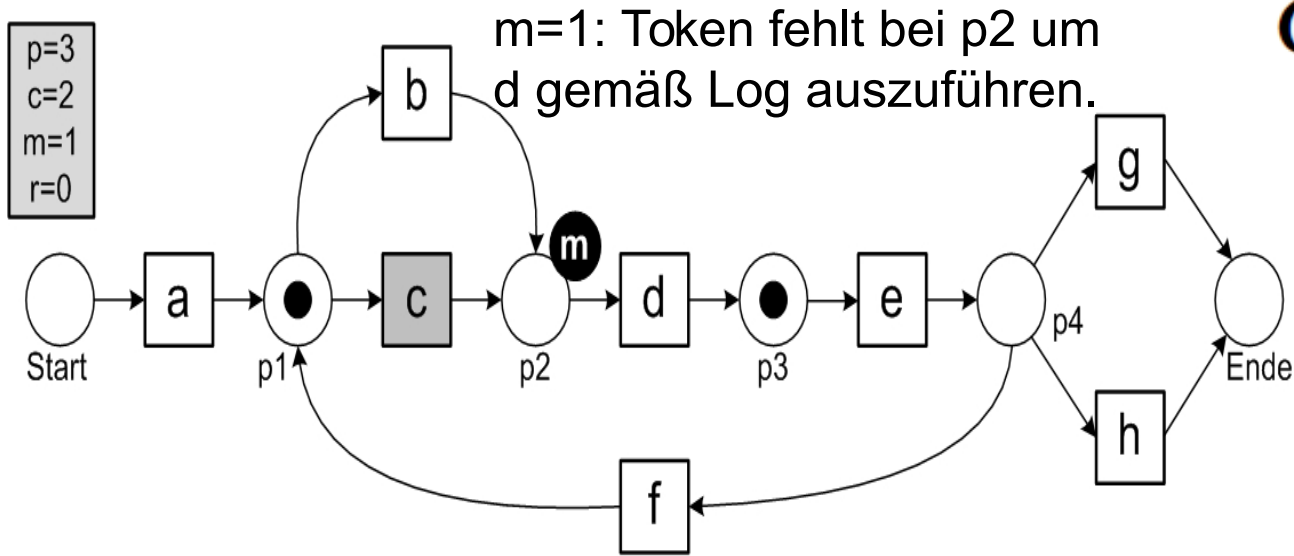
$$\sigma_3 = \langle a, d, c, e, h \rangle$$

- Umwelt produziert 1 Token für Stelle *Start*
- 0 Token konsumiert, **missing** (fehlt) oder **remaining** (übrig)
- a aktiv gemäß Log



- a konsumiert 1 Token und produziert 1 Token \rightarrow 2 Token produziert & 1 Token konsumiert
- 0 Token **missing** oder **remaining**
- d aktiv gemäß Log

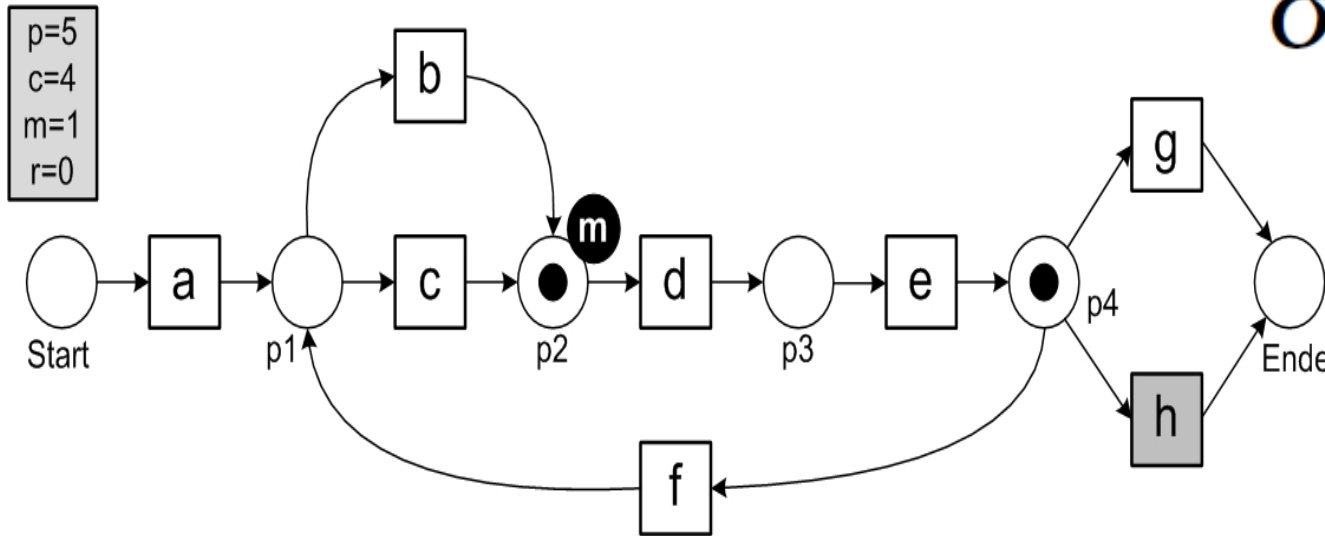
Replaying σ_3 auf N_2 (2/3)



$$\sigma_3 = \langle a, d, c, e, h \rangle$$

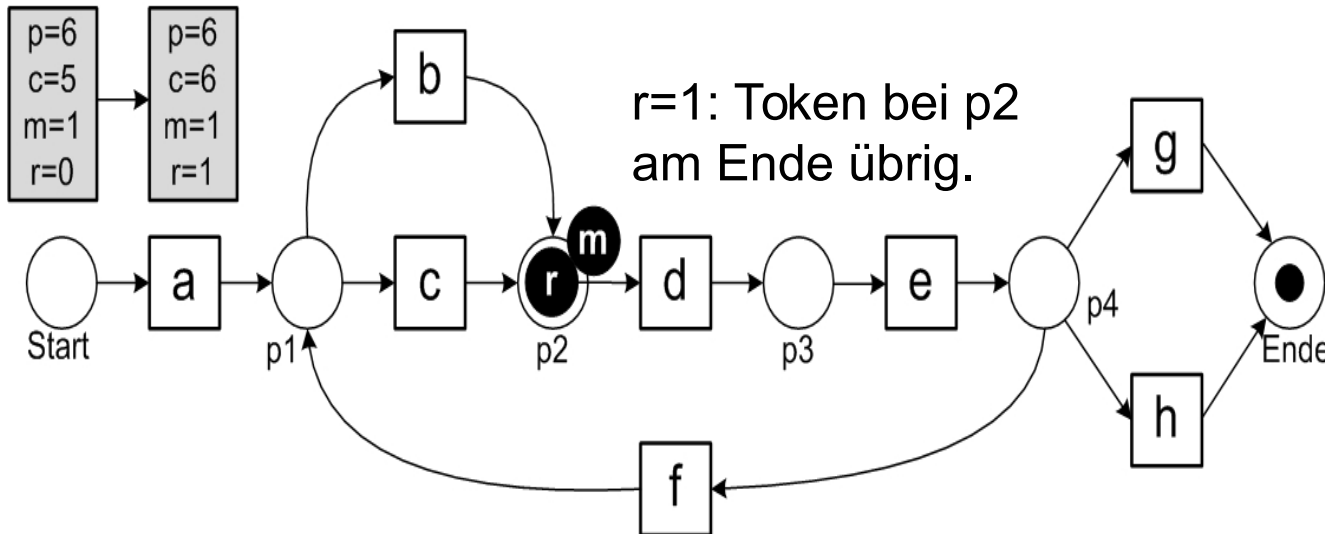
- d nicht freigegeben \rightarrow 1 Token in p2 hinzufügen & markieren
- d konsumiert 1 Token und produziert 1 Token \rightarrow 3 Token produziert & 2 Token konsumiert
- 1 Token **missing**, 0 Token remaining
- c aktiv gemäß Log
- c konsumiert 1 Token und produziert 1 Token \rightarrow 4 Token produziert & 3 Token konsumiert
- 1 Token **missing**, 0 Token remaining
- e aktiv gemäß Log

Replaying σ_3 auf N_2 (3/3)



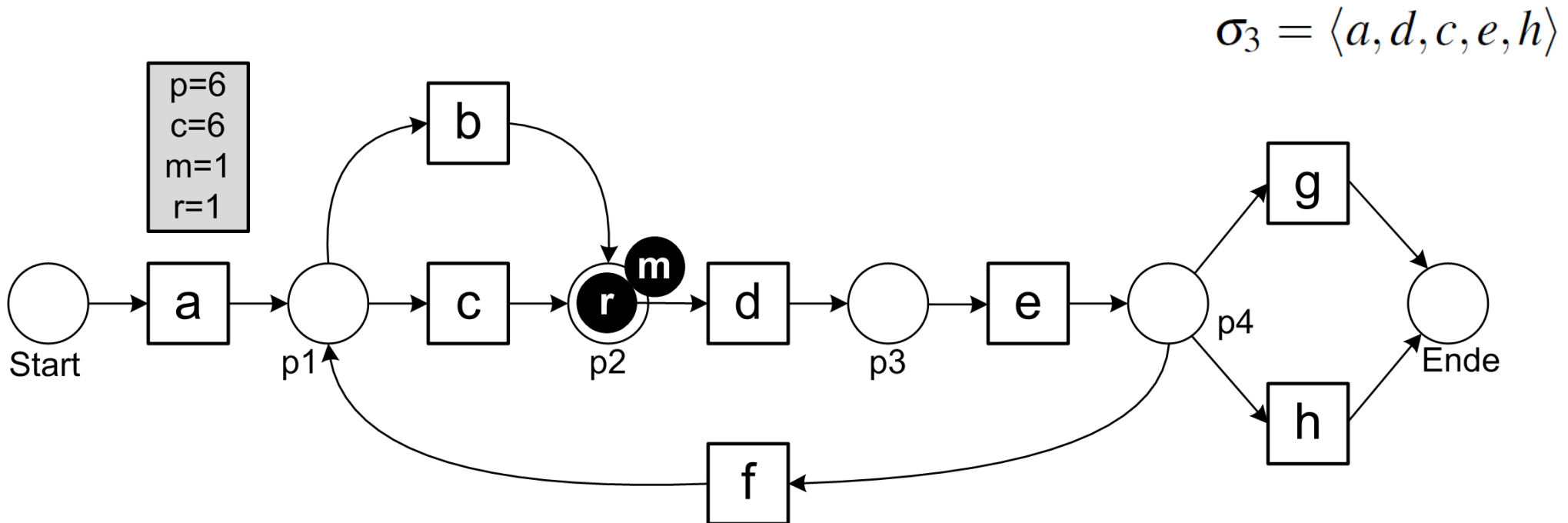
$$\sigma_3 = \langle a, d, c, e, h \rangle$$

- e konsumiert 1 Token und produziert 1 Token \rightarrow 5 Token produziert & 4 Token konsumiert
- 1 Token missing, 0 Token remaining
- h aktiv gemäß Log



- h konsumiert 1 Token und produziert 1 Token \rightarrow 6 Token produziert & 5 Token konsumiert
- 1 Token missing, 0 Token remaining

Probleme beim Replay von σ_3 auf N_2



- Umwelt konsumiert ein Token → 6 Token produziert und konsumiert
- 1 fehlender Token (von 6 aufgenommenen Token) → 1 Token missing
- 1 Token bleibt übrig (von 6 erzeugten Token) → 1 Token remaining

Angemessenheit (Fitness) des Modelles zu Log-Folge

$$fitness(\sigma, N) = \frac{1}{2} \left(1 - \frac{m}{c} \right) + \frac{1}{2} \left(1 - \frac{r}{p} \right)$$

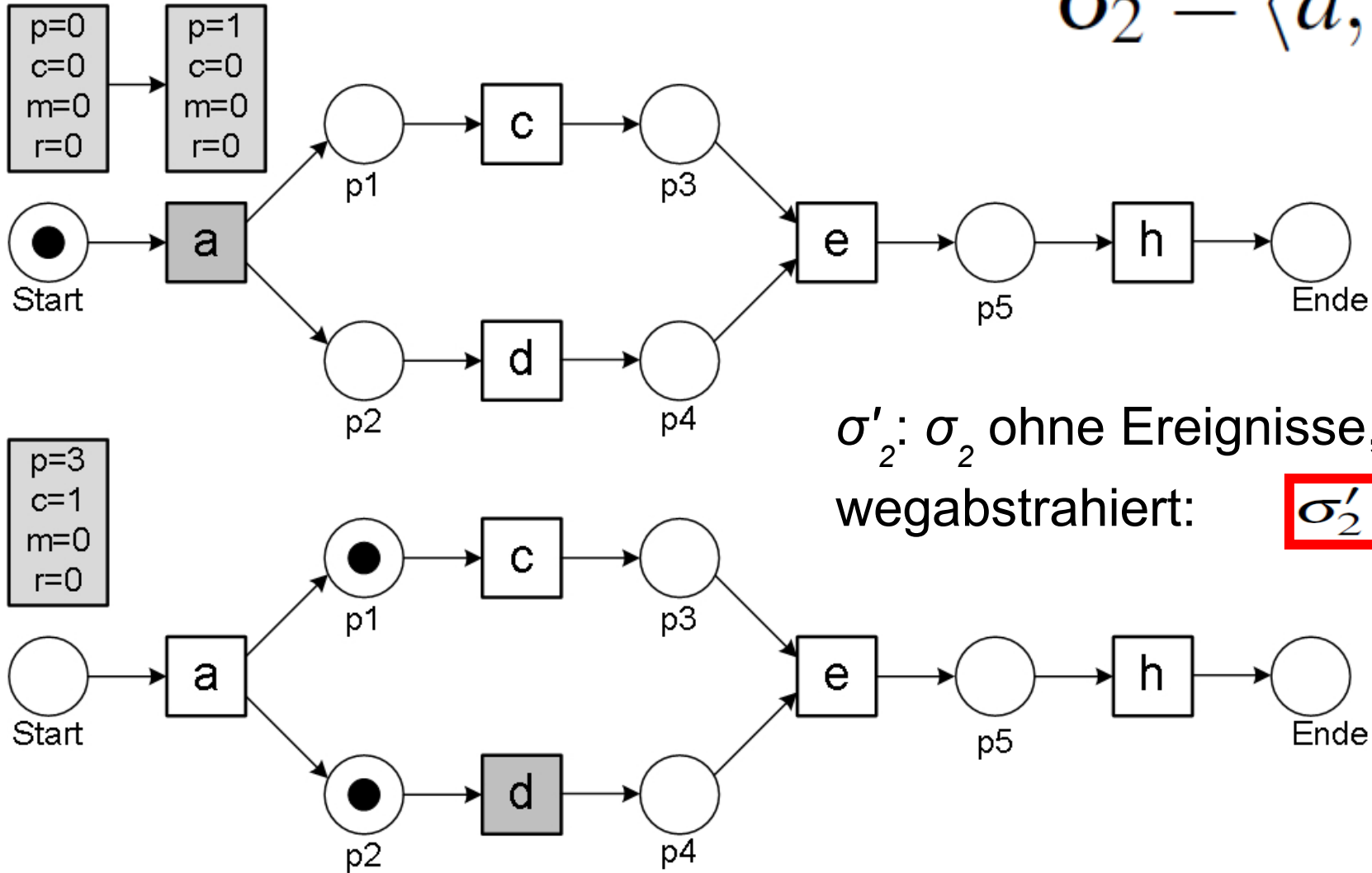
- **produced, consumed, missing, remaining**
- $(1 - m/c)$ berechnet Anteil **fehlender Tokens**.
- $(1 - r/P)$ berechnet Anteil **überbleibender Tokens**.
- **σ** : Log, **N** : Modell.
- $0 \leq fitness(\sigma, N) \leq 1$.
- Falls $fitness(\sigma, N) = 1$: Keine fehlende oder überbleibende Token.

Beispiel:

$$fitness(\sigma_3, N_2) = \frac{1}{2} \left(1 - \frac{1}{6} \right) + \frac{1}{2} \left(1 - \frac{1}{6} \right) = 0.8333$$

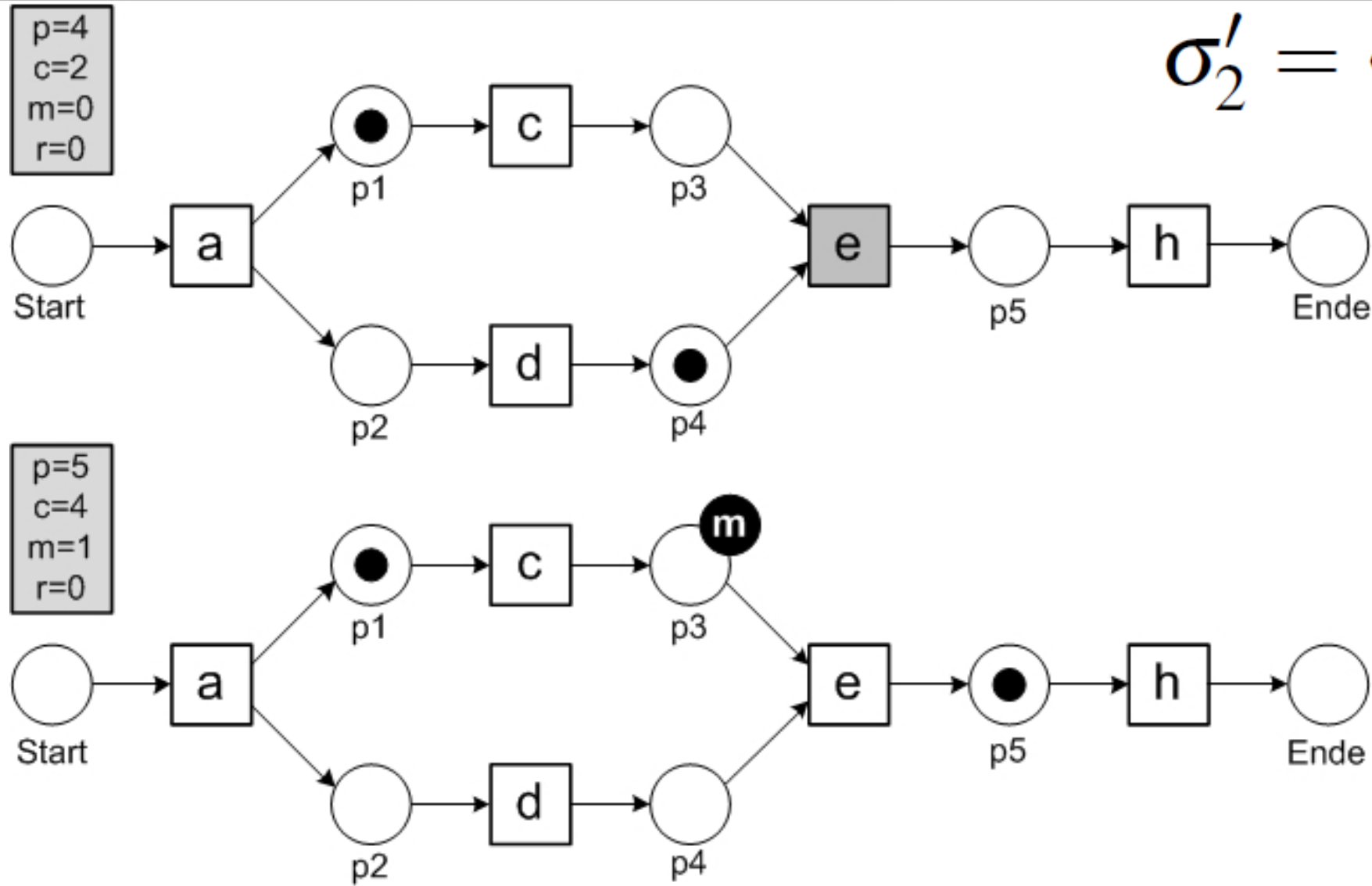
Replaying σ_2 auf N_3 (1/3)

$$\sigma_2 = \langle a, b, d, e, g \rangle$$



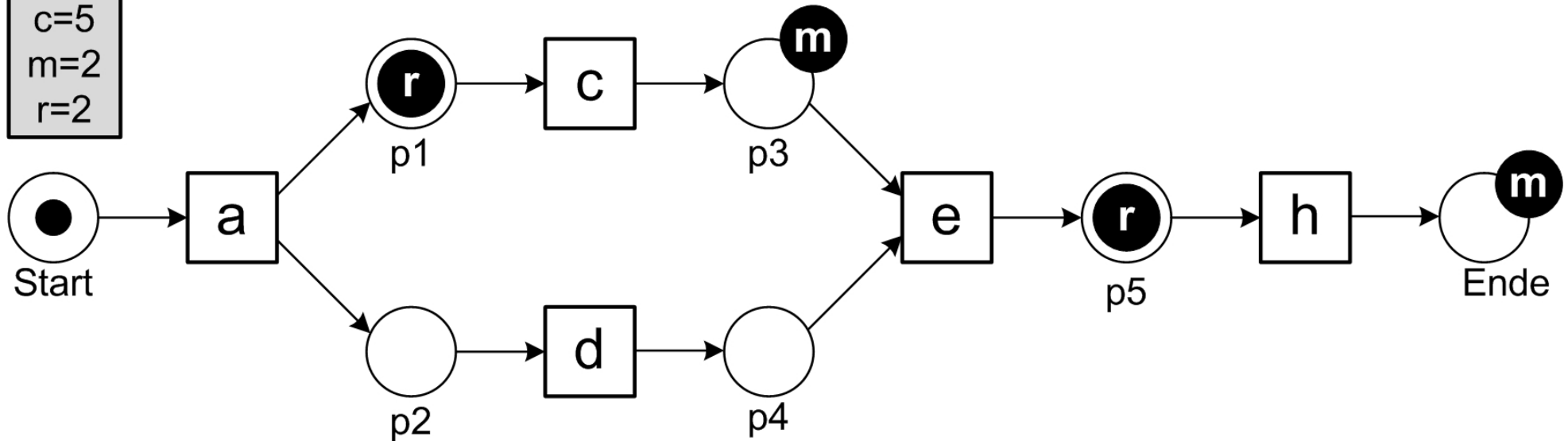
Replaying σ_2 auf N_3 (2/3)

$$\sigma'_2 = \langle a, d, e \rangle$$



Replaying σ_2 auf N_3 (3/3)

p=5
c=5
m=2
r=2



$$\sigma'_2 = \langle a, d, e \rangle$$

$$fitness(\sigma_2, N_3) = \frac{1}{2} \left(1 - \frac{2}{5} \right) + \frac{1}{2} \left(1 - \frac{2}{5} \right) = 0.6$$

Angemessenheit (Fitness) zu Menge von Log-Folgen

$$\mathit{fitness}(L, N) = \frac{1}{2} \left(1 - \frac{\sum_{\sigma \in L} L(\sigma) \times m_{N, \sigma}}{\sum_{\sigma \in L} L(\sigma) \times c_{N, \sigma}} \right) + \frac{1}{2} \left(1 - \frac{\sum_{\sigma \in L} L(\sigma) \times r_{N, \sigma}}{\sum_{\sigma \in L} L(\sigma) \times p_{N, \sigma}} \right)$$

N : Modell

$L(\sigma)$: Häufigkeit der Folge σ in Logdatei L .

produced, **c**onsumed, **m**issing, **r**emaining bei Replay von σ auf N .

Beispielwerte

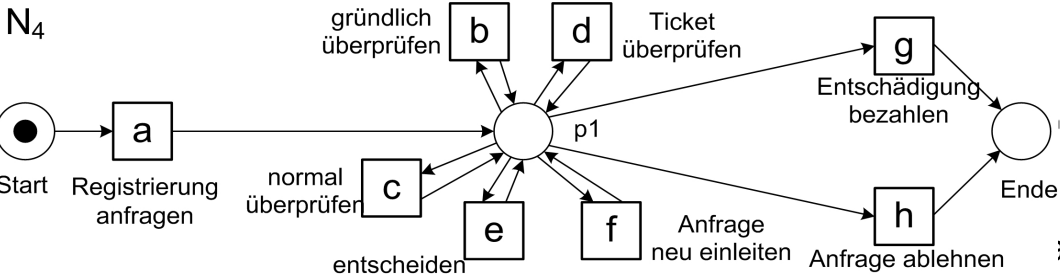
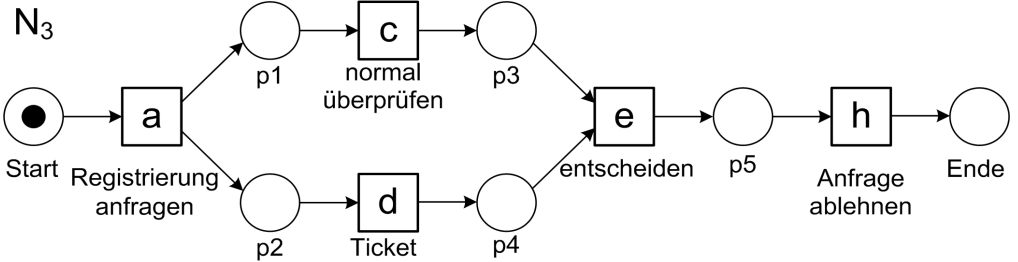
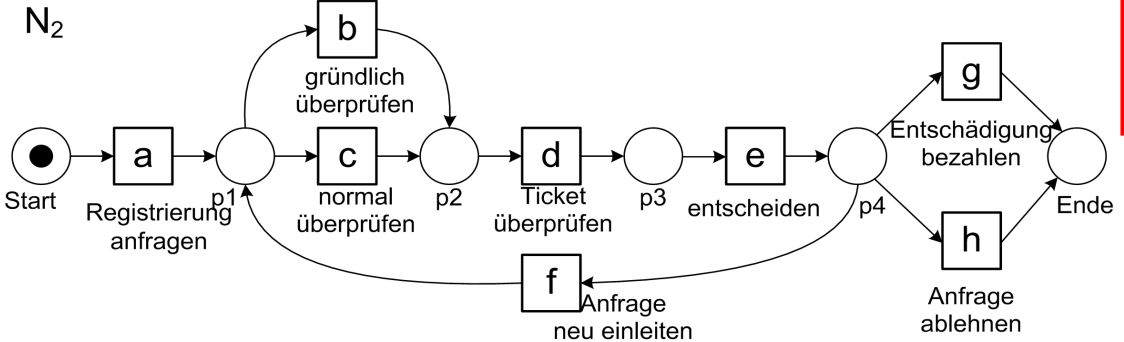
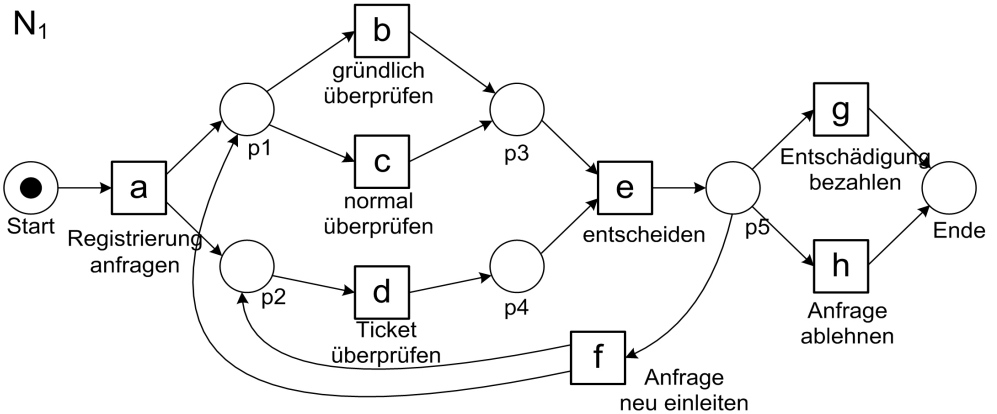


$$fitness(L_{full}, N_1) = 1$$

$$fitness(L_{full}, N_2) = 0.9504$$

$$fitness(L_{full}, N_3) = 0.8797$$

$$fitness(L_{full}, N_4) = 1$$

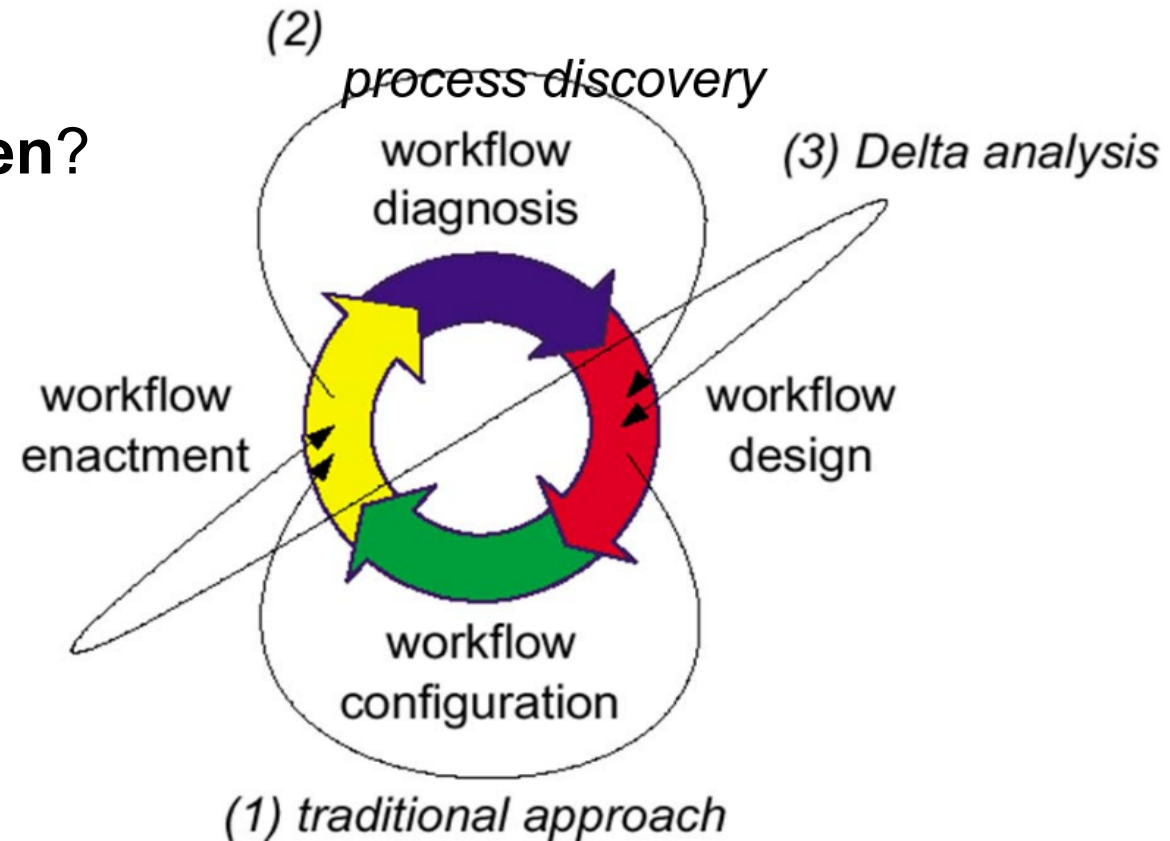


Überblick Konformanzanalyse

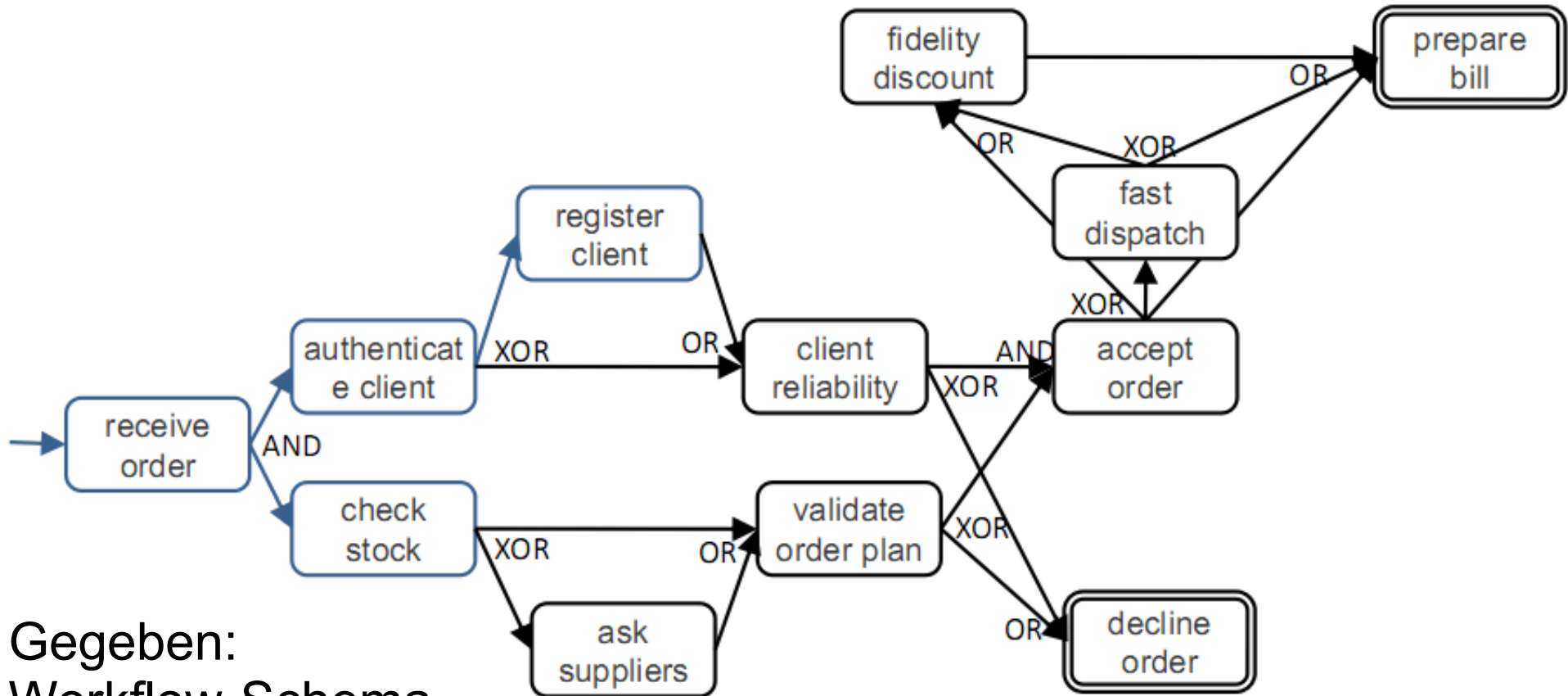
- Einführung
- Konformanzanalyse basierend auf Replay
 - Motivation
 - Beispiele
 - Konformanzmaße
- **Diagnose**

Nächster Schritt („**workflow diagnosis**“):

- Welche Pfade sind **häufig**?
- Welche Instanzen werden wahrscheinlich **abgebrochen**?



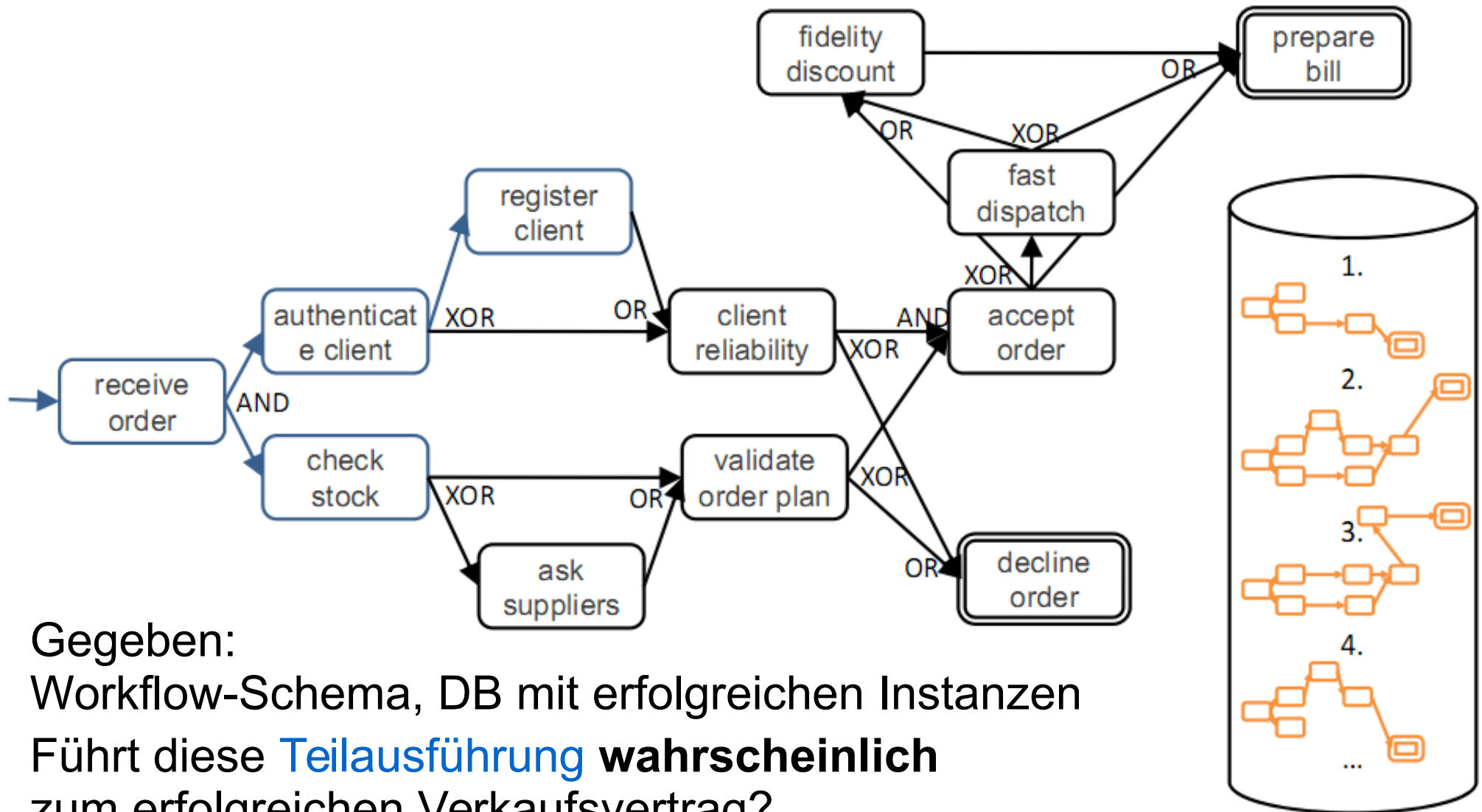
Workflow-Diagnose: Statische Analyse



Gegeben:
Workflow-Schema

Kann diese **Teilausführung** zum erfolgreichen Verkaufsvertrag führen ?

Workflow-Diagnose: Analyse der Laufzeitdaten



Weitere interessante Fragestellungen:

- **Fehlerfreie** Workflow-Ausführung ?
- Workflow-Ausführung mit geringem oder normalem **Ressourcenverbrauch** ?
- Identifikation **kritischer Aktivitäten** (hoher Ressourcenverbrauch).
- Wie sehen **häufige / typische Ausführungen** aus?

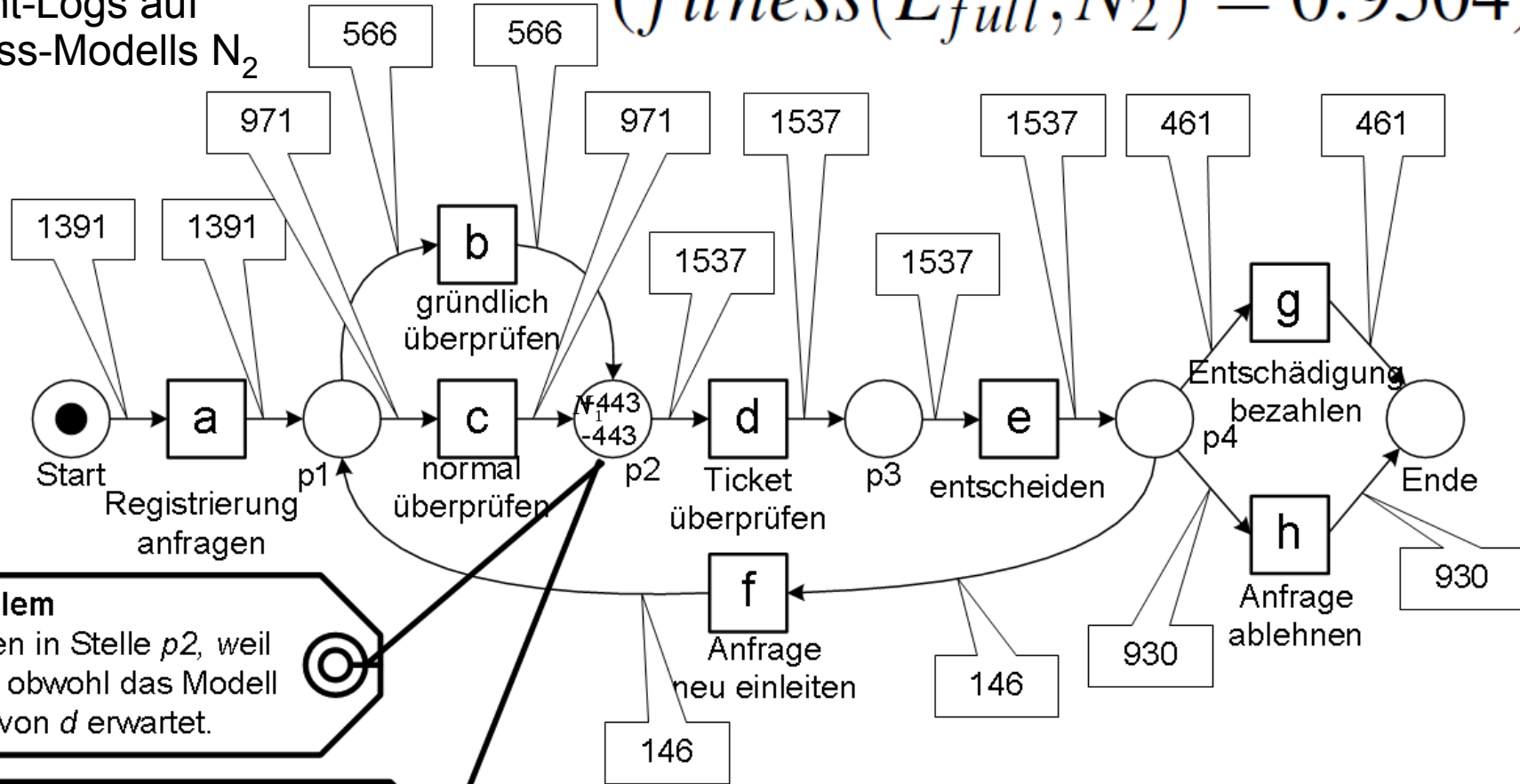
Verwertung der Information:

- Zur Laufzeit: **Scheduling**.
- Information für Management / Optimierung (Business Process Reengineering (BPR), Continuous Process Improvement (CPI))

- Im folgenden betrachten wir zwei Beispiele, bei denen eine **Diagnose** durchgeführt wird.
- Dazu wurde auf Basis eines **Event-Logs** und eines **Prozessmodells** ein **Replay** durchgeführt.
- Anhand daran kann man erkennen, ob **Konformanzprobleme** auftreten.
- Ist das Modell „falsch“ und sollte überarbeitet werden?
- Weicht der Event-Log vom Modell ab?

$$(\text{fitness}(L_{full}, N_2) = 0.9504)$$

Replay des Event-Logs auf
Basis des Prozess-Modells N₂



Problem

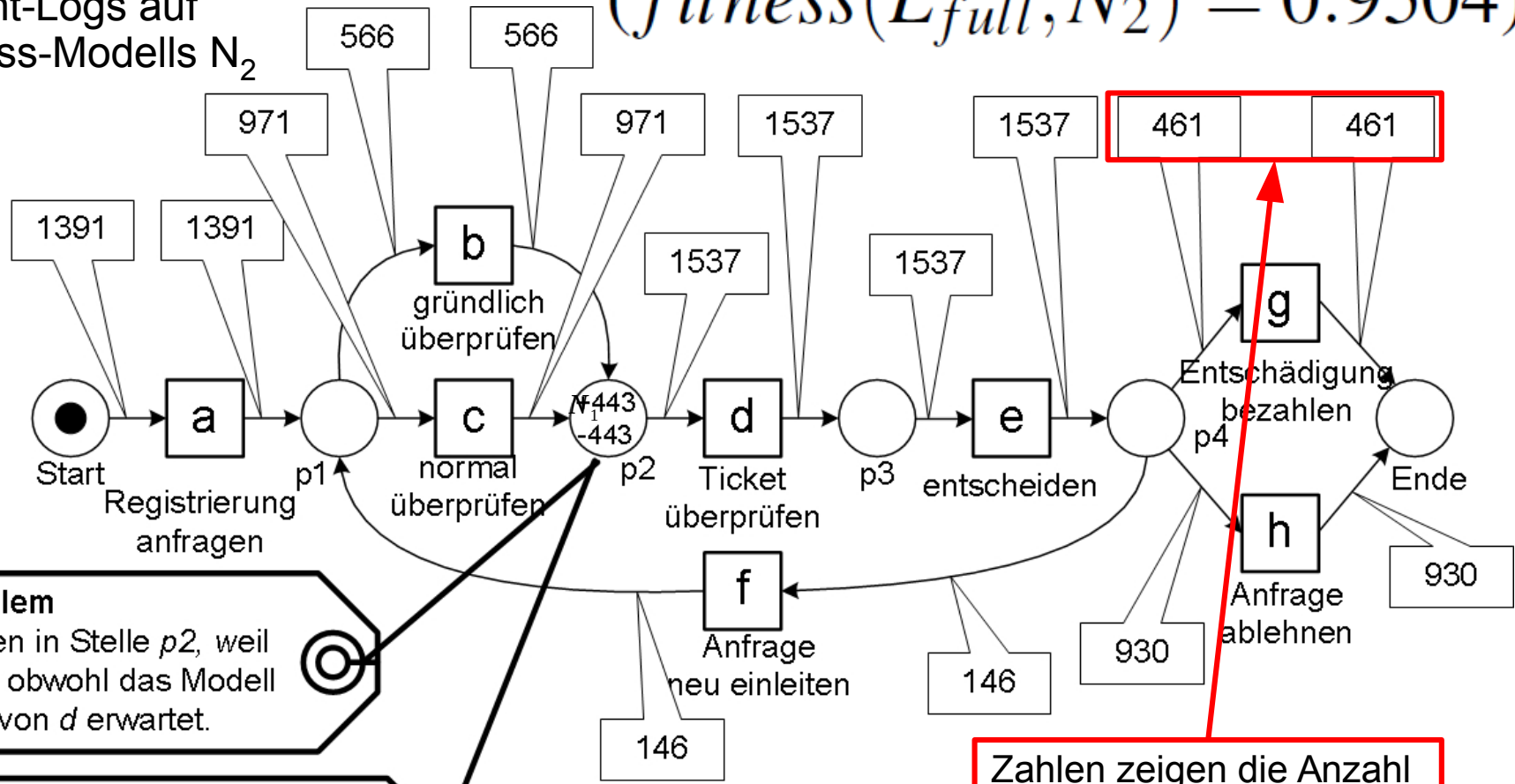
443 Token verbleiben in Stelle p2, weil d nicht erreicht wird, obwohl das Modell die Ausführung von d erwartet.

Problem

Beim Replay fehlen 443 Token in Stelle p2, da d stattfindet, obwohl das, laut Modell, nicht möglich ist.

$$(\text{fitness}(L_{full}, N_2) = 0.9504)$$

Replay des Event-Logs auf
Basis des Prozess-Modells N₂



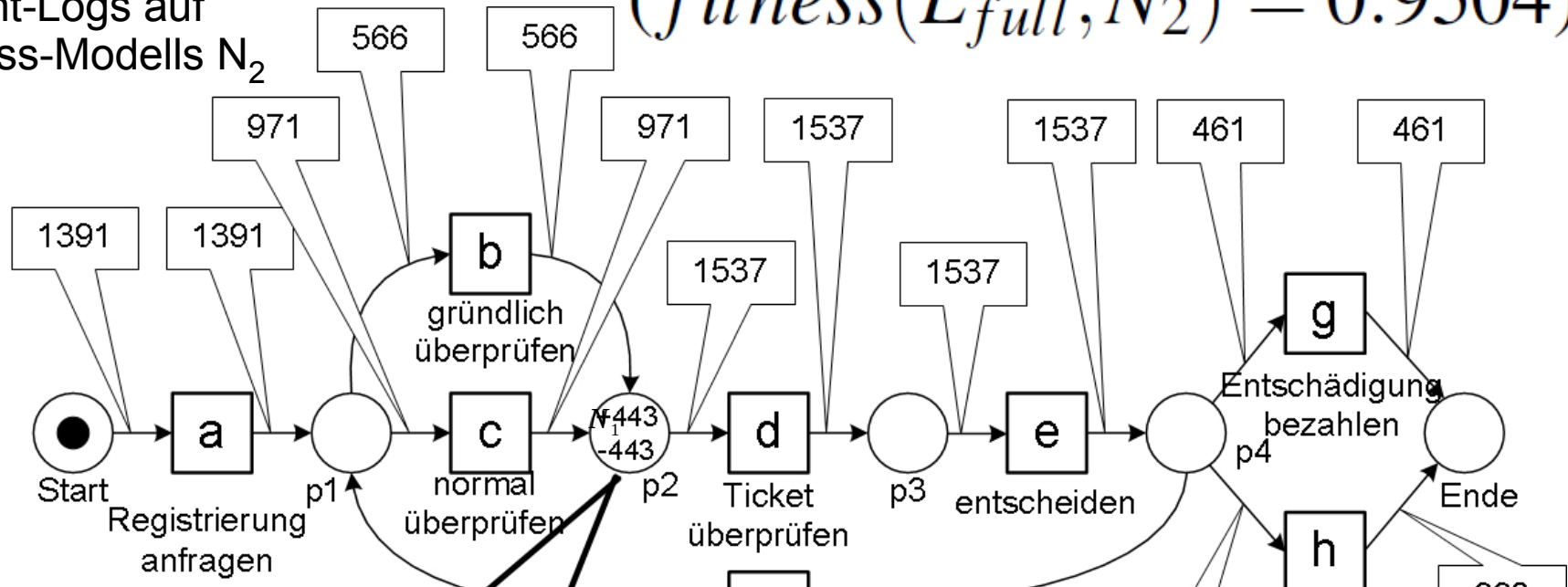
Problem
443 Token verbleiben in Stelle p2, weil d nicht erreicht wird, obwohl das Modell die Ausführung von d erwartet.

Problem
Beim Replay fehlen 443 Token in Stelle p2, da d stattfindet, obwohl das, laut Modell, nicht möglich ist.

Zahlen zeigen die Anzahl der produzierten und konsumierten Token an

$$(\text{fitness}(L_{full}, N_2) = 0.9504)$$

Replay des Event-Logs auf
Basis des Prozess-Modells N₂



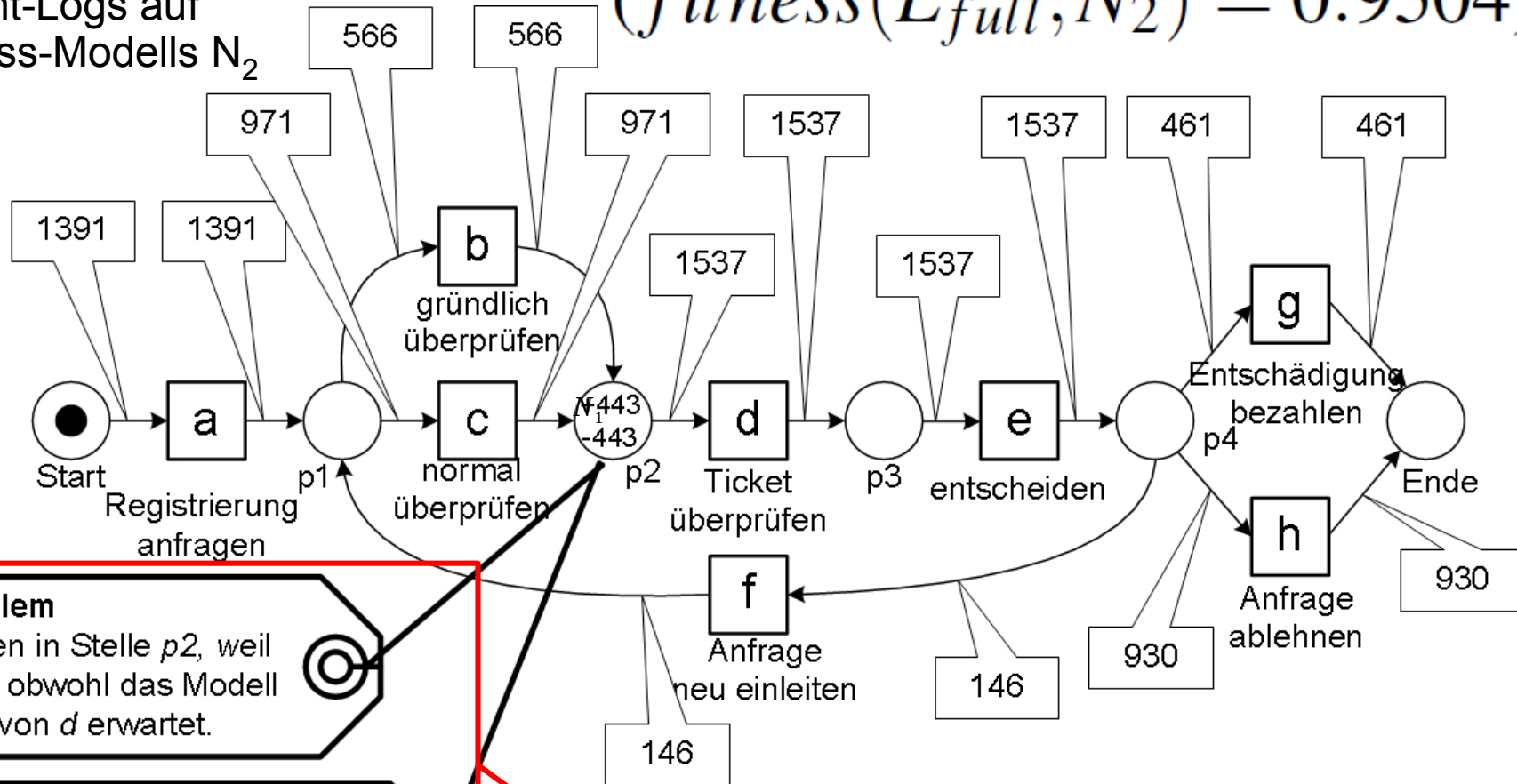
Problem
443 Token verbleiben in Stelle p₂, weil d nicht erreicht wird, obwohl das Modell die Ausführung von d erwartet.

Problem
Beim Replay fehlen 443 Token in Stelle p₂, da d stattfindet, obwohl das, laut Modell, nicht möglich ist.

Hier wurde z.B. 146 eine Anfrage neu eingeleitet

$$(\text{fitness}(L_{full}, N_2) = 0.9504)$$

Replay des Event-Logs auf
Basis des Prozess-Modells N₂



Problem

443 Token verbleiben in Stelle p2, weil d nicht erreicht wird, obwohl das Modell die Ausführung von d erwartet.

Problem

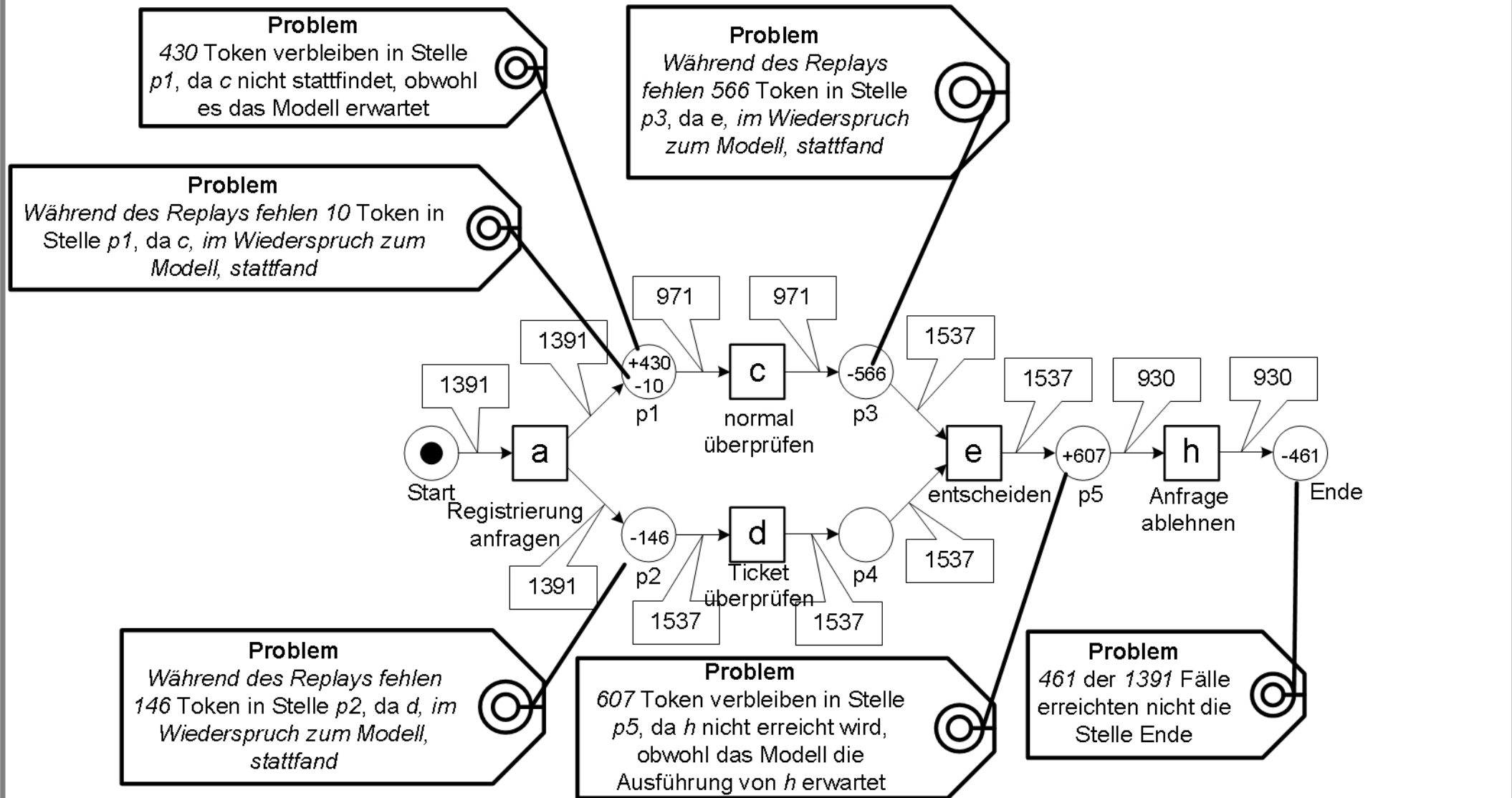
Beim Replay fehlen 443 Token in Stelle p2, da d stattfindet, obwohl das, laut Modell, nicht möglich ist.

Sollte das Modell überarbeitet werden?

Diagnose – Beispiel N₃

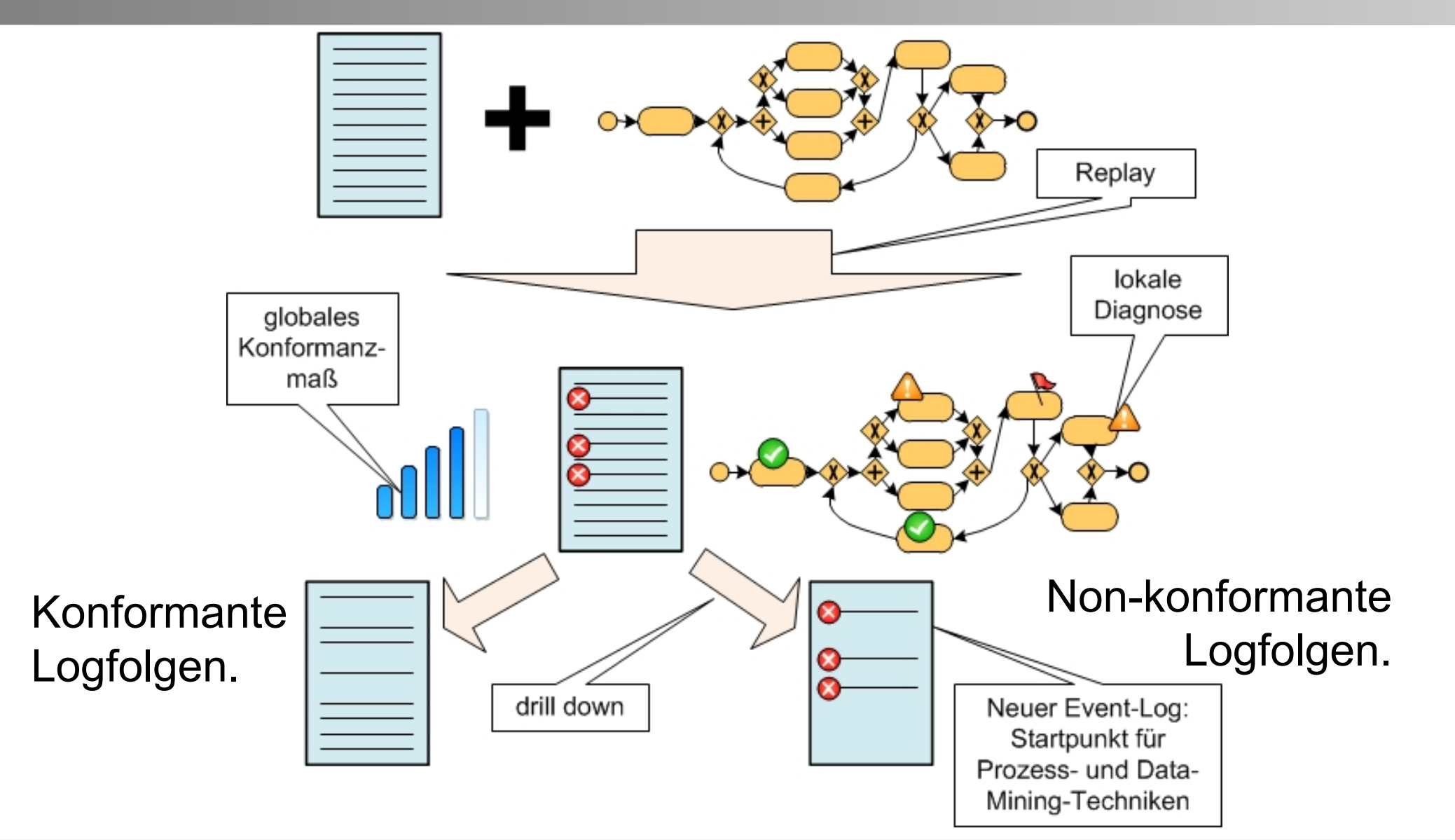


$$(fitness(L_{full}, N_3) = 0.8797)$$



- An den Beispielen kann man erkennen, dass es Fälle gibt, bei denen die Logfolge **konformant** ist, und Fälle, bei denen die Logfolge **nicht konformant** ist.
- Es ist sinnvoll, konformante Logfolgen und nicht konformante Logfolgen zu trennen und **zwei neue Event-Logs** zu erstellen.
- Diese können für weitere Analysen hilfreich sein.
- Kann z.B. für die nicht konformanten Logfolgen ein Prozessmodell gefunden werden ?

Drilling down (Verfeinerte Analyse)



Vergleich der durch α -Algorithmus definierten **Relationen** (footprints):

- aus gegebenen **Logdaten** (z.B. L_{full}) bzw.
- aus **Petrinetz** (z.B. N_1) generierten Logdaten.

Für L_{full} und N_1 (identisch):

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>
<i>a</i>	#	→	→	→	#	#	#	#
<i>b</i>	←	#	#		→	←	#	#
<i>c</i>	←	#	#		→	←	#	#
<i>d</i>	←			#	→	←	#	#
<i>e</i>	#	←	←	←	#	→	→	→
<i>f</i>	#	→	→	→	←	#	#	#
<i>g</i>	#	#	#	#	←	#	#	#
<i>h</i>	#	#	#	#	←	#	#	#

Vergleiche Footprints.

Oben: N_2 . Unten: L_{full} und N_1 .

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>
<i>a</i>	#	→	→	#	#	#	#	#
<i>b</i>	←	#	#	→	#	←	#	#
<i>c</i>	←	#	#	→	#	←	#	#
<i>d</i>	#	←	←	#	→	#	#	#
<i>e</i>	#	#	#	←	#	→	→	→
<i>f</i>	#	→	→	#	←	#	#	#
<i>g</i>	#	#	#	#	←	#	#	#
<i>h</i>	#	#	#	#	←	#	#	#

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>
<i>a</i>	#	→	→	→	#	#	#	#
<i>b</i>	←	#	#		→	←	#	#
<i>c</i>	←	#	#		→	←	#	#
<i>d</i>	←			#	→	←	#	#
<i>e</i>	#	←	←	←	#	→	→	→
<i>f</i>	#	→	→	→	←	#	#	#
<i>g</i>	#	#	#	#	←	#	#	#
<i>h</i>	#	#	#	#	←	#	#	#

Vergleiche Footprints.

Oben: N_2 . Unten: L_{full} und N_1 .

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>
<i>a</i>	#	→	→	#	#	#	#	#
<i>b</i>	←	#	#	→	#	←	#	#
<i>c</i>	←	#	#	→	#	←	#	#
<i>d</i>	#	←	←	#	→	#	#	#
<i>e</i>	#	#	#	←	#	→	→	→
<i>f</i>	#	→	→	#	←	#	#	#
<i>g</i>	#	#	#	#	←	#	#	#
<i>h</i>	#	#	#	#	←	#	#	#

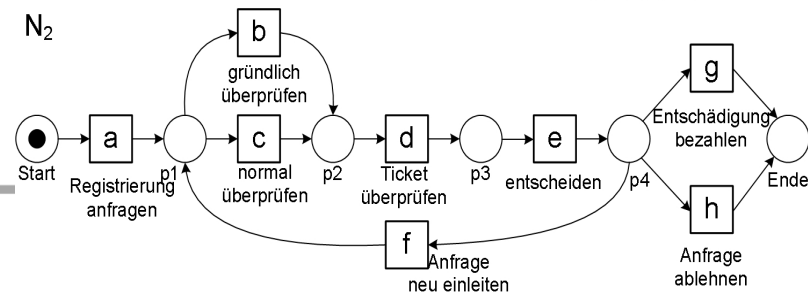
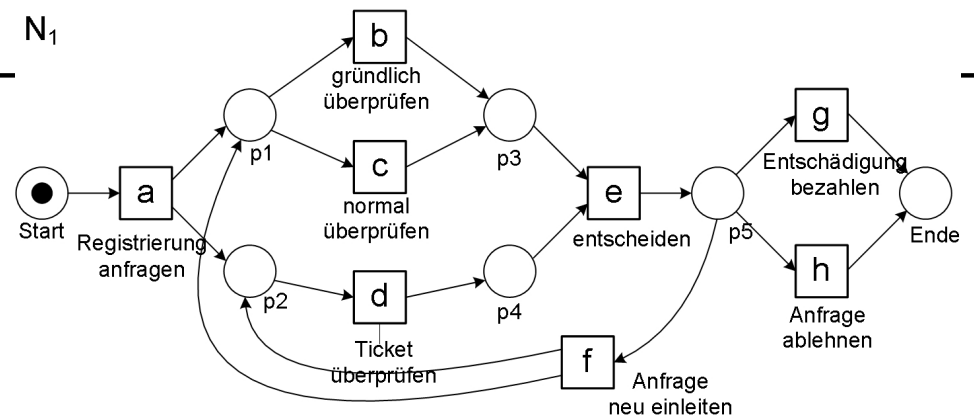
	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>
<i>a</i>	#	→	→	→	#	#	#	#
<i>b</i>	←	#	#		→	←	#	#
<i>c</i>	←	#	#		→	←	#	#
<i>d</i>	←			#	→	←	#	#
<i>e</i>	#	←	←	←	#	→	→	→
<i>f</i>	#	→	→	→	←	#	#	#
<i>g</i>	#	#	#	#	←	#	#	#
<i>h</i>	#	#	#	#	←	#	#	#

Diagnose: Unterschiede quantifiziert

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>
<i>a</i>				$\rightarrow: \#$				
<i>b</i>				$\parallel: \rightarrow$	$\rightarrow: \#$			
<i>c</i>				$\parallel: \rightarrow$	$\rightarrow: \#$			
<i>d</i>	$\leftarrow: \#$	$\parallel: \leftarrow$	$\parallel: \leftarrow$			$\leftarrow: \#$		
<i>e</i>		$\leftarrow: \#$	$\leftarrow: \#$					
<i>f</i>				$\rightarrow: \#$				
<i>g</i>								
<i>h</i>								

($x:y$ x ist in Log und y in N_2)

$$1 - \frac{12}{64} = 0.8125$$



In diesem Abschnitt:

- Einführung
- Konformanzanalyse basierend auf Replay
 - Motivation
 - Beispiel
 - Konformanzmaße
- Diagnose

Im nächsten Abschnitt:

- Mining: Zusätzliche Perspektiven (Organizational und Decision Mining)