

# Erlernen deterministischer Echtzeitautomaten aus Beispieldaten: RTI und RTI+

Präsentation von Lennart Leder  
Seminar Wissensbasierte Systeme  
Paderborn, 31. Januar 2014

# Überblick

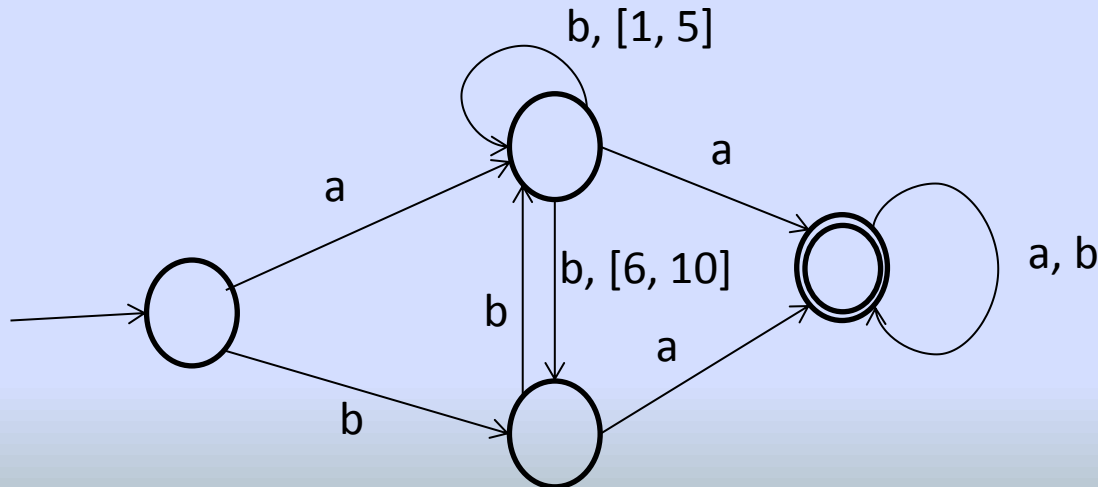
1. (Probabilistisch)-Deterministische Echtzeitautomaten
2. Elemente der Algorithmen
  1. Prefix Tree Acceptors (PTA)
  2. Merge-Operation
  3. Split-Operation
3. Der RTI-Algorithmus als überwachtes Lernverfahren
4. Der RTI+-Algorithmus als unüberwachtes Lernverfahren
5. Ausblick: Online-Induktion von probabilistischen Echtzeitautomaten

# Motivation

- Modellierung von Systemen mit unbekanntem Systemverhalten
  - z.B. Verhalten eines Lkw-Fahrers
- Zeitliche Abstände als zusätzliche Informationsquelle
- Automatische Einteilung von Ereignisfolgen in Klassen, z.B. erwünscht/unerwünscht (RTI)
- Vorhersage künftiger Ereignisse basierend auf den bisher gesehenen (RTI+)

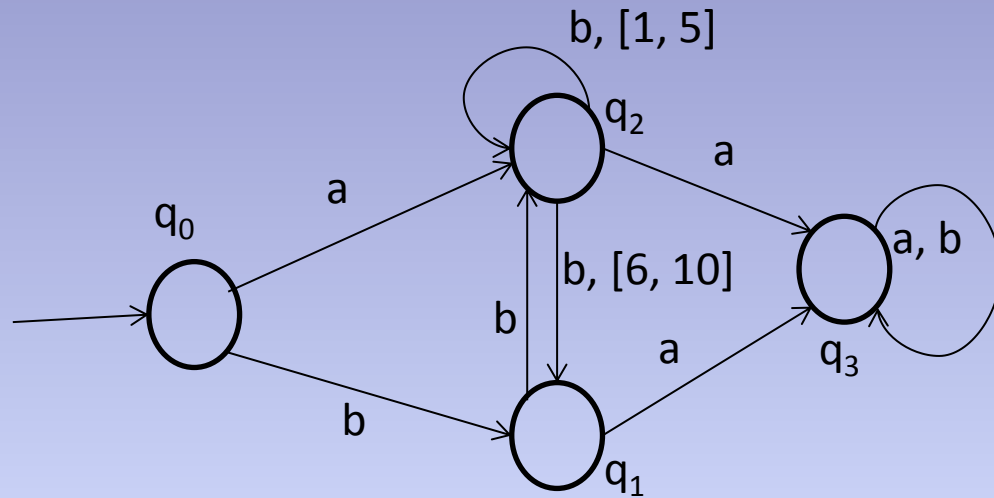
# Deterministische Echtzeitautomaten (DRTA)

- 5-Tupel  $A = (Q, \Sigma, \Delta, q_0, F)$
- $Q$ : Menge der Zustände
- $\Sigma$ : Eingabealphabet
- $\Delta: Q \times \Sigma \times \mathbb{N} \rightarrow Q$  Übergangsfunktion
- $q_0 \in Q$  Startzustand
- $F \subseteq Q$  Menge der Zielzustände



# Probabilistisch-deterministische Echtzeitautomaten (PDRTA)

- 4-Tupel  $A = (A', H, S, T)$
- $A = (Q, \Sigma, \Delta, q_0)$  deterministischer Echtzeitautomat ohne Zielzustände
- $H$  Menge von disjunkten Zeitintervallen
- $S$  Menge von Wahrscheinlichkeitsverteilungen über  $\Sigma$
- $T$  Menge von Wahrscheinlichkeitsverteilungen über  $H$



$\text{Pr}(s q)$	a	b
$q_0$	0.5	0.5
$q_1$	0.2	0.8
$q_2$	0.6	0.4
$q_3$	1	0

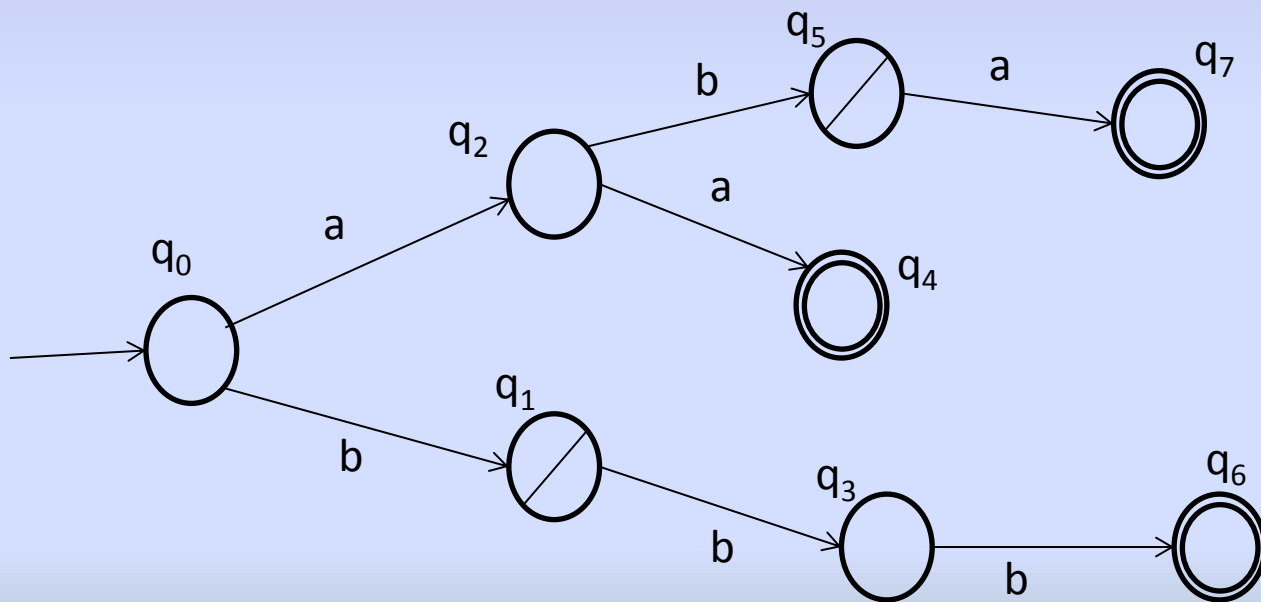
$\text{Pr}(h q)$	[1,5]	[6,10]
$q_0$	0.8	0.2
$q_1$	0.3	0.7
$q_2$	0.5	0.5
$q_3$	0.7	0.3

# Prefix Tree Acceptor

- Technik aus dem Erlernen von DFAs
- Aufbau eines Baumes aus Beispielmenge
- Jedes Beispiel repräsentiert durch einzelnen Pfad im Automaten
- 3 Arten von Zuständen: akzeptierende, ablehnende und neutrale
- Bei gezeiteten Eingaben inkonsistente Zustände möglich

# Prefix Tree Acceptor - Beispiel

- $S_+ = \{aa, aba, bbb\}$
- $S_- = \{b, ab\}$



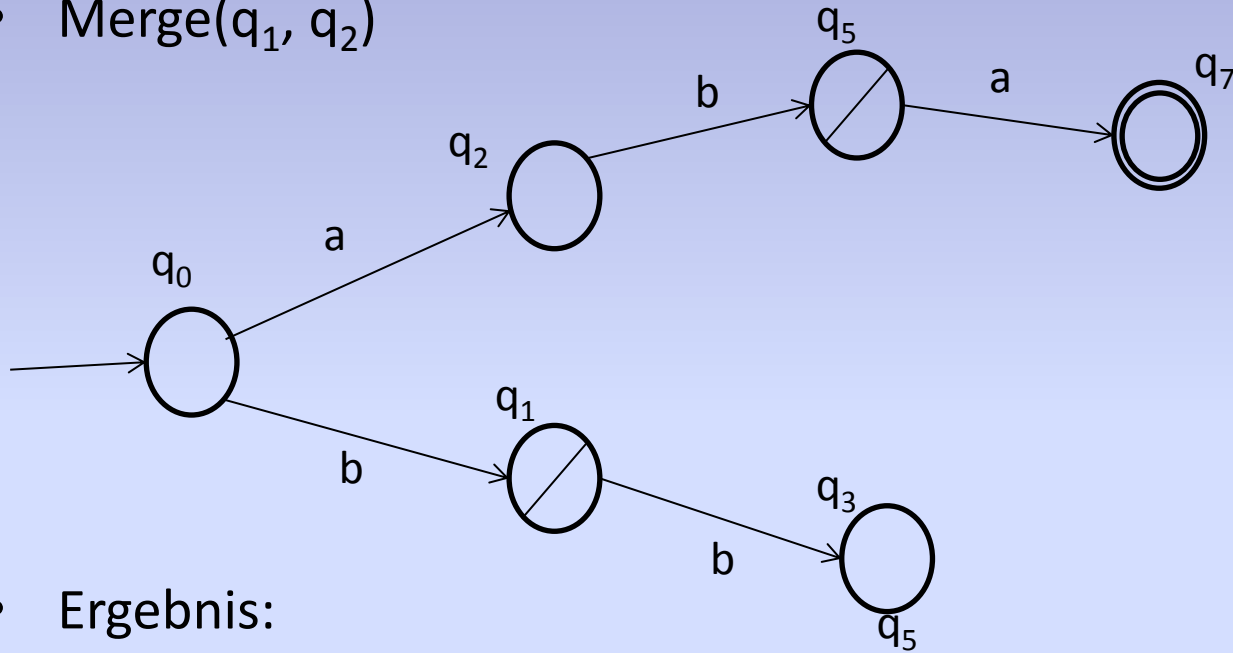


# Die Merge-Operation

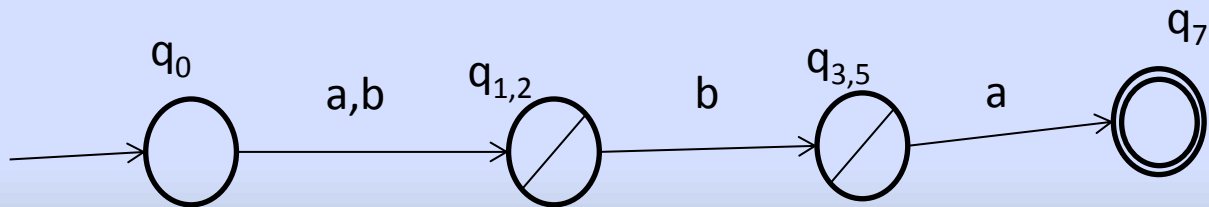
- Zusammenfassen zweier Zustände
- Bildung eines neuen Zustandes mit allen Transitionen beider Vorgängerzustände
- Möglicherweise Determinisierung erforderlich
- Inkonsistente Merges bei DFAs unzulässig
- Bei DRTAs müssen Inkonsistenzen später beseitigt werden

# Die Merge-Operation - Beispiel

- Merge( $q_1, q_2$ )



- Ergebnis:

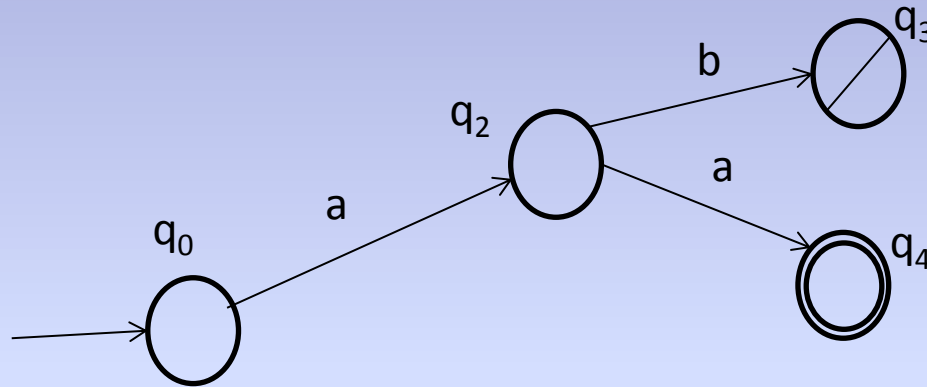


# Die Split-Operation

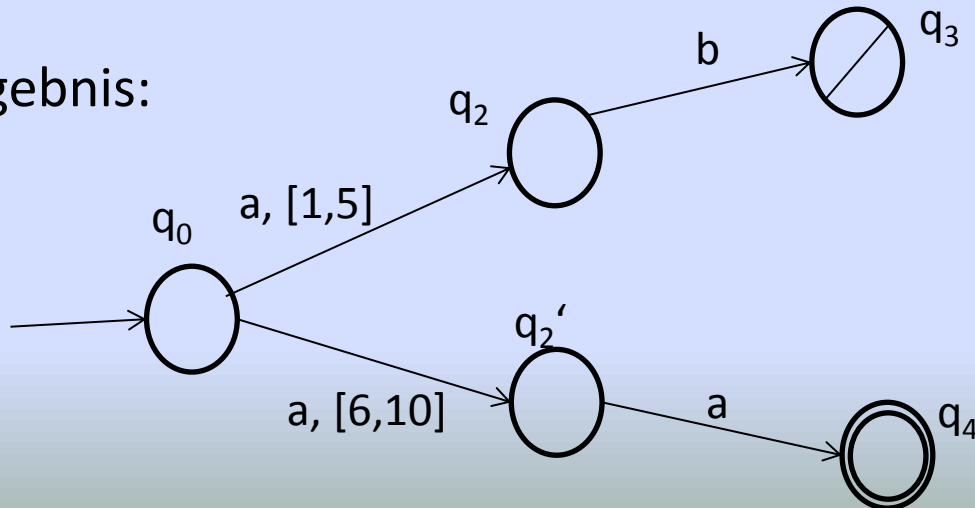
- Besonderheit beim Erlernen von gezeiteten Automaten
- Aufteilen einer Transition in zwei verschiedene an einem bestimmten Zeitwert
- Folgezustände der beiden Transitionen abhängig von Beispieldaten
- Ermöglicht Beseitigung von Inkonsistenzen

# Die Split-Operation: Beispiel

- $\text{Split}((q_0, a, [1,10], q_2), 5)$

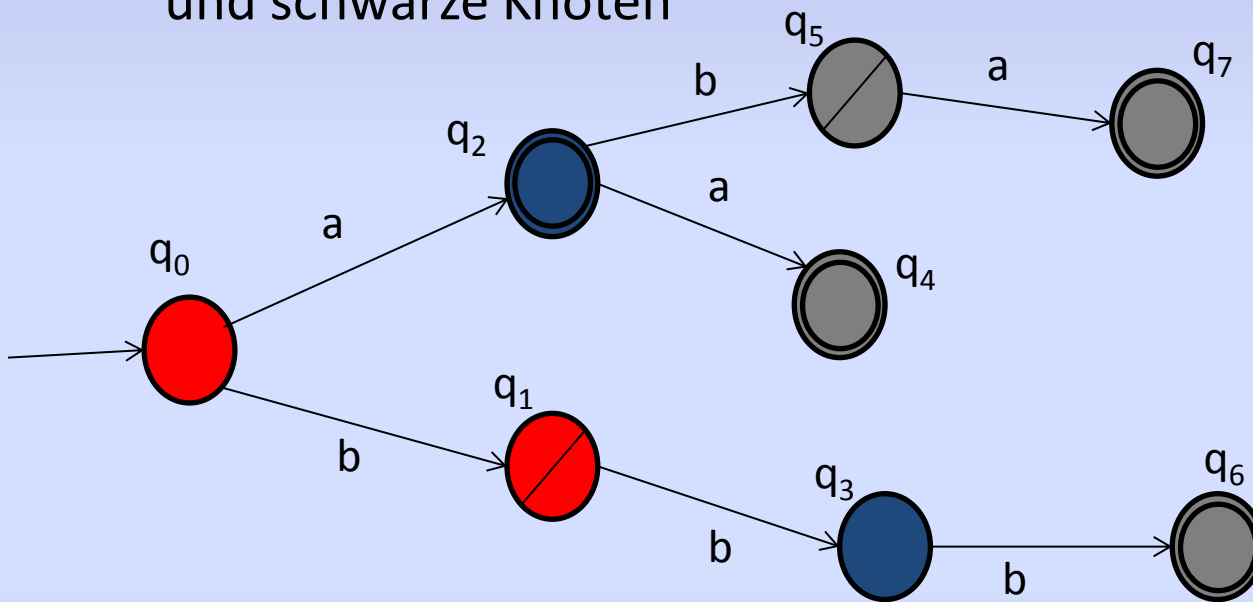


- Ergebnis:



# Der RTI-Algorithmus: Vorgehen

1. Konstruktion eines Prefix Tree Acceptors aus den Beispieldaten
2. Transformation des Automaten mittels Merges und Splits:
  - Färbung des Automaten ähnlich einer Breitensuche: Rote, blaue und schwarze Knoten



- Merge immer zwischen rotem und blauem Knoten
- Split nur auf Transitionen zwischen roten und blauen Knoten

# RTI: Auswahl der nächsten Operation

- Invariante: keine inkonsistenten roten Knoten
- Zuordnung einer Punktzahl (Score) an jede zulässige Operation
- Score beschreibt erwarteten Effekt einer Operation auf die Größe des berechneten Automaten
- Berechnung basierend auf Heuristik; einbezogene Elemente sind
  - Anzahl der korrekt zusammengefassten roten Zustände
  - Konsistenzwerte
  - Inkonsistenzwerte
- RTI terminiert, sobald alle Knoten rot gefärbt sind

# RTI - Eigenschaften

- Erzeugt stets konsistente Automaten
- Keine Garantien bezüglich optimaler Größe des Automaten
- Polynomielle Laufzeit
- Testergebnisse zeigen bessere Performance als ein Sampling-Ansatz mit DFAs
- Hohe Klassifikationsraten (> 80%) allerdings nur für relativ kleine Probleminstanzen
- Schlechtere Klassifikationsraten bei großen Beispielmengen

# Der RTI+-Algorithmus: Idee

- Eingabedaten nur von einem Typ
- Klassifikation der Eingabedaten nicht relevant oder nicht möglich
- Ziel: Modellierung eines Systems zur Vorhersage des künftigen Systemverhaltens
- Verwendung von probabilistisch-deterministischen Echtzeitautomaten
- Jeder Zustand definiert Wahrscheinlichkeitsverteilungen
- Definition einer Wahrscheinlichkeit für jede mögliche Eingabe
- Grundsätzliche Arbeitsweise identisch zum RTI-Algorithmus

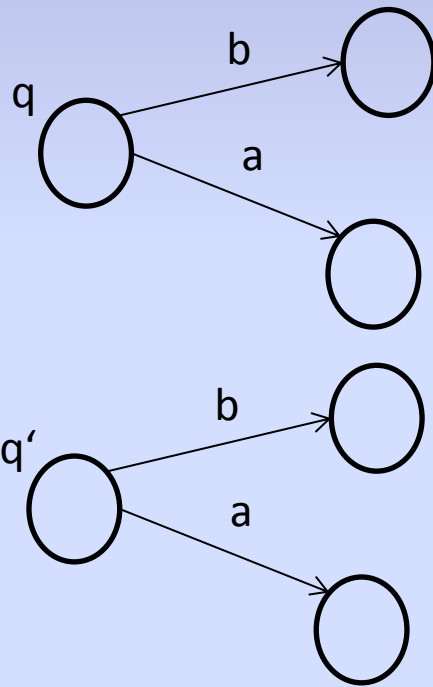


# RTI+: Auswahl der nächsten Operation

- Konsistenz bedeutungslos
- Stattdessen statistisches Verfahren: Hypothesentest
- Vergleich von Wahrscheinlichkeitsverteilung mittels  $\chi^2$ -Anpassungstest
- Zusammenfassen von Zuständen, wenn die Wahrscheinlichkeitsverteilung nicht zu stark verändert wird
- Statistisch beste Split- oder Merge-Operation wird ausgeführt
- Signifikanzniveau 0.05

# Hypothesentest - Beispiel

- Fragestellung: Sollten  $q$  und  $q'$  zusammengefasst werden?
- $S_1 = \{4*a, 1*b\}$ ,  $S_2 = \{1*a, 4*b\}$



- $\Pr(S_1 | q) = \Pr(S_2 | q') = 0.8^4 * 0.2 = 0.08192$
- $\Pr(S_1 | q \cap S_2 | q') = 0.08192^2 \approx 0.00671$
- $\Pr(S_1 \cup S_2 | q'') = 0.5^{10} \approx 0.00098$
- $LR = 0.00098 / 0.00671 \approx 0.14554$
- $-2 * \ln(LR) \approx 3.8546$
- $p \approx 0.0296$  (1 Freiheitsgrad)

- $\chi^2$ -Anpassungstest berechnet aus der Likelihood-ratio den sogenannten p-Wert

# RTI+: Eigenschaften

- Wahrscheinlichkeitsverteilungen basierend auf Maximum-Likelihood-Schätzern
- Polynomielle Laufzeit
- Test des Algorithmus mittels Daten eines zufällig erzeugten PDRTAs:
  - Schwierigkeiten beim Erlernen von zeitabhängigen Transitionen
  - Kann Automatenstruktur nachbilden
  - Skalierbarkeit fraglich: Maximal 8 Zustände bei 2000 erzeugten Beispieldaten

# Ausblick: Inkrementelles Lernen von PDRTA

- Neuerer Ansatz: Konstruktion eines Automaten aus einzelner Datum
- Inkrementelles Einbinden weiterer Daten
- Vorteile:
  - Konstruktion eines PTA entfällt
  - Zur Laufzeit kann weiteres Wissen eingebunden werden
  - Geeignet für Datenströme
  - Erkennung zeitlicher Veränderungen

# Zusammenfassung

- Deterministische Echtzeitautomaten als Erweiterung von deterministischen endlichen Automaten
- RTI als überwachtes Lernverfahren zum Erlernen eines möglichst kleinen konsistenten DRTA
- RTI+ als unüberwachtes Lernverfahren zur statistischen Analyse und Vorhersage von Systemverhalten
- Dynamische Ansätze als Alternative zu den statischen Lernverfahren

# Zentrale Literatur

RTI ist beschrieben in:

Verwer, Sicco, Mathijs DeWeerd, and Cees Witteveen. An algorithm for learning real-time automata. Proceedings of the Sixteenth Annual Machine Learning Conference of Belgium and the Netherlands (Benelearn). 2007.

RTI+ ist beschrieben in:

Verwer, Sicco, Mathijs de Weerd, and Cees Witteveen. A likelihood-ratio test for identifying probabilistic deterministic real-time automata from positive data. Grammatical Inference: Theoretical Results and Applications. Springer Berlin Heidelberg, 2010. 203-216.