Nichtdeterministische Platzklassen

Sommerakademie Rot an der Rot — AG 1
Wieviel Platz brauchen Algorithmen wirklich?

Ulf Kulau

TU Braunschweig

12. August 2010

Outline

Einleitung

Warum sich den Kopf zerbrechen?

- Platz sollte heutzutage nicht mehr das Problem sein
- 1 GByte kosten $\approx$ 7 Cent

Einleitung

Warum sich den Kopf zerbrechen?

- Platz sollte heutzutage nicht mehr das Problem sein
- 1 GByte kosten $\approx$ 7 Cent

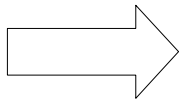
Beispiel für logarithmischen Platz:
1 TByte Eingabedaten, bei für die Berechnung angenommenem logarithmischem Platzbedarf ($log_2(n) \Rightarrow log_2(8 \cdot 10^{12}\text{bit}) \approx 43 Bit$)

Einleitung

Warum sich den Kopf zerbrechen?

- Platz sollte heutzutage nicht mehr das Problem sein
- 1 GByte kosten $\approx$ 7 Cent

Beispiel für logarithmischen Platz:
1 TByte Eingabedaten, bei für die Berechnung angenommenem logarithmischem Platzbedarf ($log_2(n) \Rightarrow log_2(8 \cdot 10^{12}\text{bit}) \approx 43 Bit$)



Einleitung

Eigentliche Fragestellung des Vortrags

Wieviel Platz benötigt eine nichtdeterministische Maschine für ihre Berechnungen?

Einleitung

Eigentliche Fragestellung des Vortrags

Wieviel Platz benötigt eine nichtdeterministische Maschine für ihre Berechnungen?

Nichtdeterministische Berechnung:

"Herrlich für den Theoretiker, furchtbar für den Praktiker"

Einleitung

Eigentliche Fragestellung des Vortrags

Wieviel Platz benötigt eine nichtdeterministische Maschine für ihre Berechnungen?

Nichtdeterministische Berechnung:

"Herrlich für den Theoretiker, furchtbar für den Praktiker"

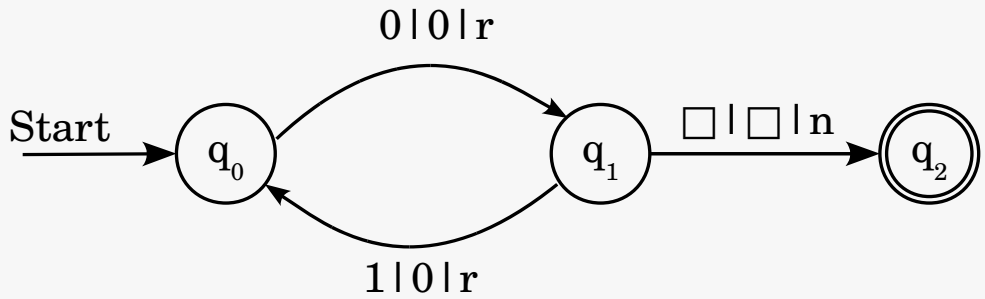
- Nichtdeterministische Maschinen kann man nicht bauen!
- Jedoch hilfreich bei der Optimierung/Lösung praktischer Probleme

Warum ist geringer Platzverbrauch hilfreich?

Beispiel: Speicherhirachien im Mikroprozessor

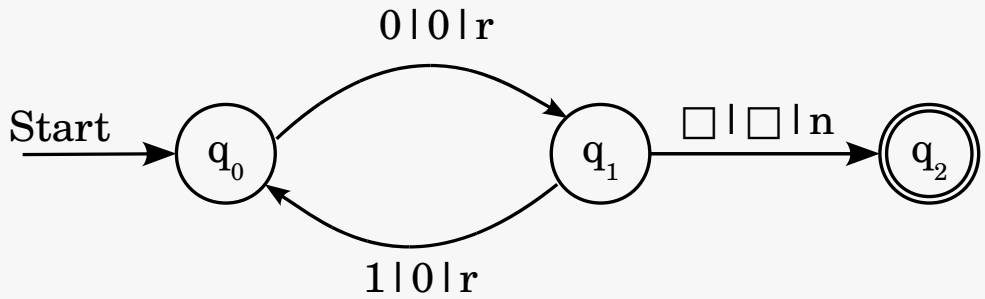
(Nicht)Deterministische Maschinen

Erinnerung an eine deterministische Maschine:



(Nicht)Deterministische Maschinen

Erinnerung an eine deterministische Maschine:

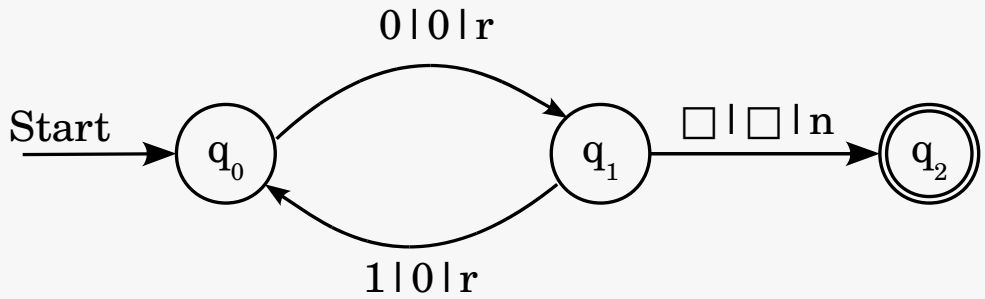


Was passiert bei den Eingabeworten:

- $w = \square 010 \square$?

(Nicht)Deterministische Maschinen

Erinnerung an eine deterministische Maschine:

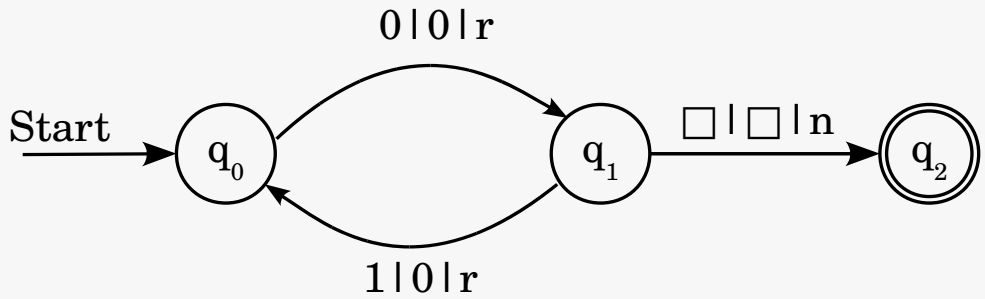


Was passiert bei den Eingabeworten:

- $w = \square 010 \square$? $\Rightarrow$ wird akzeptiert!
- $w = \square 011 \square$?

(Nicht)Deterministische Maschinen

Erinnerung an eine deterministische Maschine:



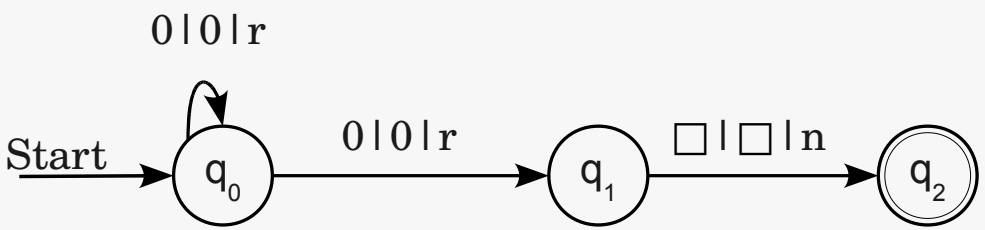
Was passiert bei den Eingabeworten:

- $w = \square 010 \square$? $\Rightarrow$ wird akzeptiert!
- $w = \square 011 \square$? $\Rightarrow$ wird nicht akzeptiert!

Bei einer deterministischen Maschine ist der Weg der Berechnung klar definiert.

(Nicht)deterministische Maschine

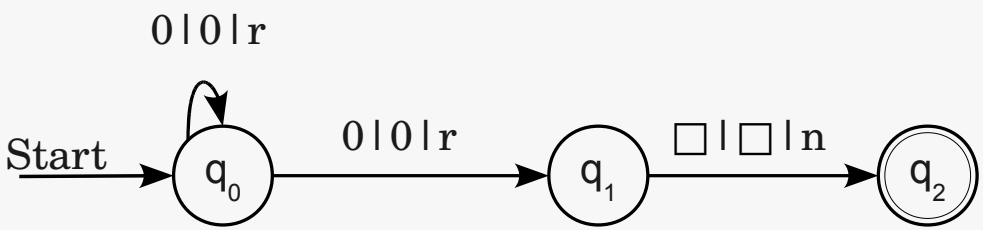
Beispiel einer nichtdeterministischen Maschine



Was passiert in q_0, wenn als Eingabe eine 0 gelesen wird?

(Nicht)deterministische Maschine

Beispiel einer nichtdeterministischen Maschine

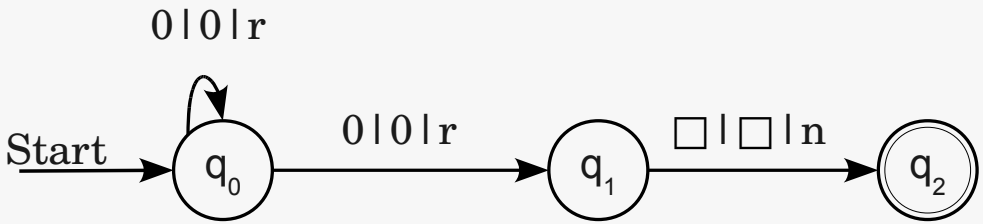


Was passiert in q_0, wenn als Eingabe eine 0 gelesen wird?

- in q_0 bleiben und nach rechts shiften
- nach q_1 gehen und ebenfalls nach rechts shiften

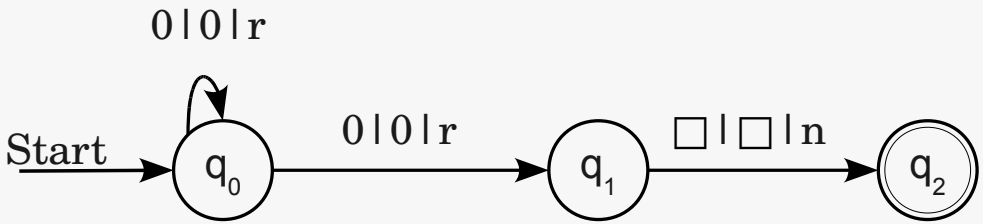
Nichtdeterministische Maschinen akzeptieren, wenn es mindestens eine Lösung gibt.

Eigenschaften nichtdeterministischer Maschinen



- keine klar definierte "Berechnung"
- Eine Eingabe wird akzeptiert (q_2), wenn es eine akzeptierende Berechnung gibt (**Berechnungsbaum**)

Eigenschaften nichtdeterministischer Maschinen



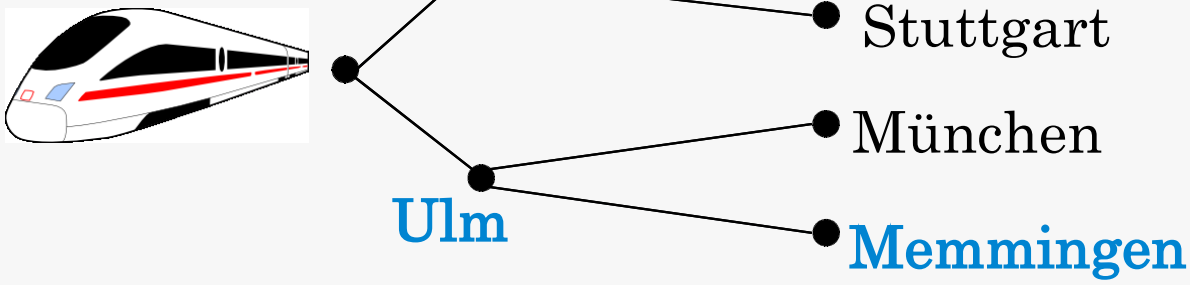
- keine klar definierte "Berechnung"
- Eine Eingabe wird akzeptiert (q_2), wenn es eine akzeptierende Berechnung gibt (**Berechnungsbaum**)

Orakel-Eigenschaft

Es entsteht der Eindruck, dass eine NTM eine Lösung "erraten" kann.

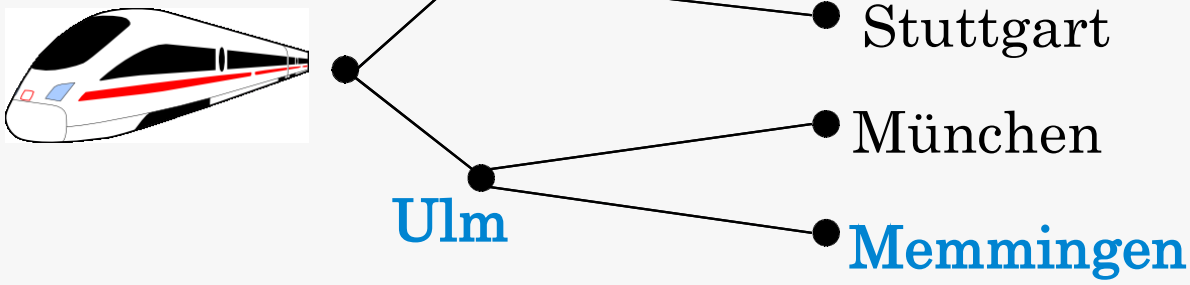
Deterministische und Nichtdeterministische Lösungsstrategien

Beispiel: Graphensuche (Mit dem Zug nach Rot)



Deterministische und Nichtdeterministische Lösungsstrategien

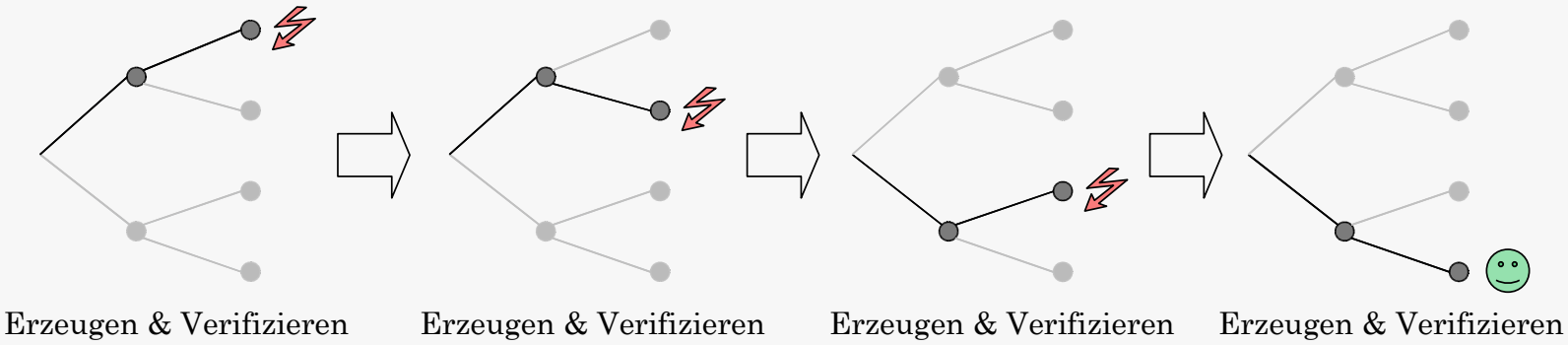
Beispiel: Graphensuche (Mit dem Zug nach Rot)



- Wie sieht eine deterministische Lösung aus?
- Wie eine nichtdeterministische?

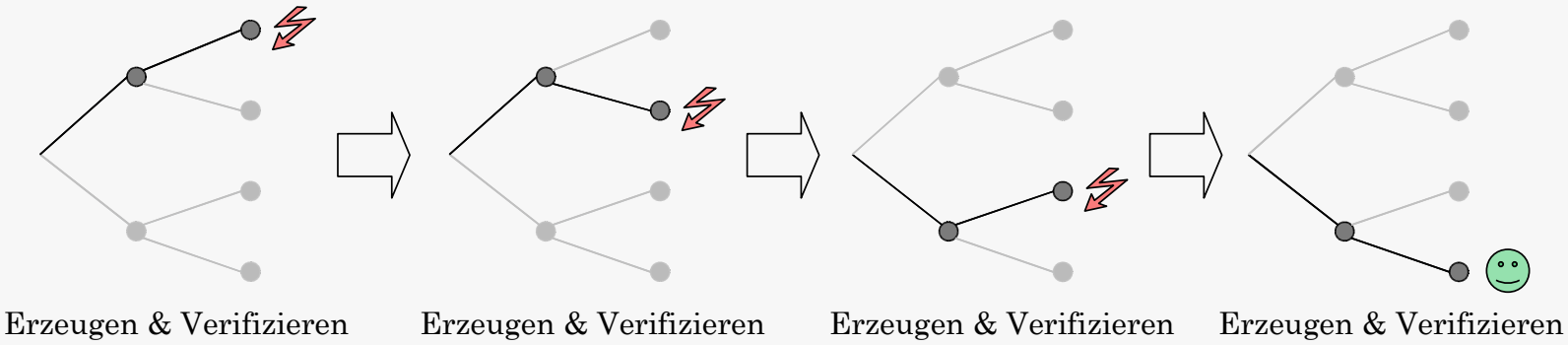
Deterministischer Lösungsansatz

- Systematisches "Durchkämmen" des Suchraumes
- Brute-Force-Methode



Deterministischer Lösungsansatz

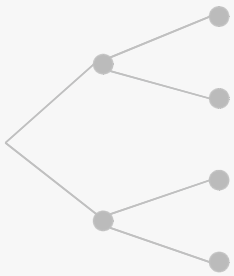
- Systematisches "Durchkämmen" des Suchraumes
- Brute-Force-Methode



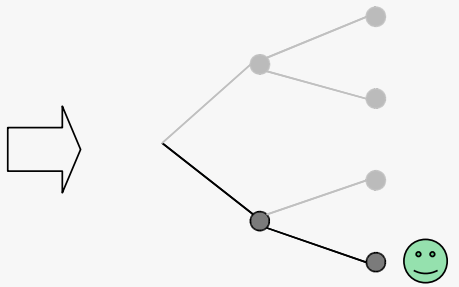
Lediglich für kleine Suchräume akzeptabel!

Nichtdeterministischer Lösungsansatz

- Es existiert eine akzeptierende Berechnung
- Lösung "erraten" und verifizieren



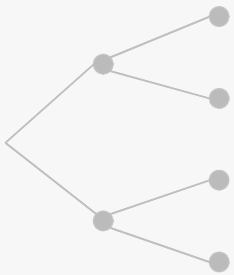
„Orakel“



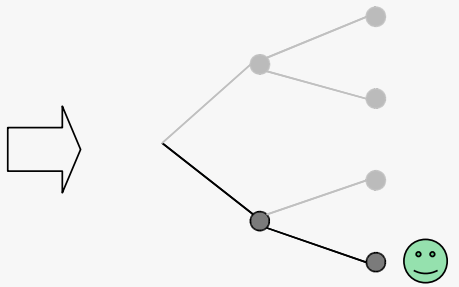
Raten & Verifizieren

Nichtdeterministischer Lösungsansatz

- Es existiert eine akzeptierende Berechnung
- Lösung "erraten" und verifizieren



„Orakel“



Raten & Verifizieren

Die Laufzeit verbessert sich, wie sieht jedoch der Platzbedarf aus?

Platzbedarf einer NTM / DTM

- Ein- Ausgabebänder zählen nicht zum Platzbedarf

Platzbedarf einer NTM / DTM

- Ein- Ausgabebänder zählen nicht zum Platzbedarf

Entscheidungsprobleme:

Platzbedarf einer NTM / DTM

- Ein- Ausgabebänder zählen nicht zum Platzbedarf

Entscheidungsprobleme:

- Exponentiell viele Möglichkeiten zum Beschreiben eines Speichersegmentes
- Jede einzelne Belegung jedoch nur polynomieller Platzbedarf

Platzbedarf einer NTM / DTM

- Ein- Ausgabebänder zählen nicht zum Platzbedarf

Entscheidungsprobleme:

- Exponentiell viele Möglichkeiten zum Beschreiben eines Speichersegmentes
- Jede einzelne Belegung jedoch nur polynomieller Platzbedarf

$$PSPACE = NPSPACE = SPACE(p(n)) \tag{1}$$

Die Komplexitätsklasse NL

Erinnerung an den Bodensee

Anforderungen an eine NTM der Klasse NL:

Die Komplexitätsklasse NL

Erinnerung an den Bodensee

Anforderungen an eine NTM der Klasse NL:
Eine NTM löst ein Entscheidungsproblem auf logarithmischem Platz

$$NL = SPACE(O(log(n))) \tag{2}$$

Erreichbarkeitsproblem in Graphen

REACH Problem:

Gibt es in einem Graphen **G** einen Weg vom Startknoten **s** zum Zielknoten **t**?

Erreichbarkeitsproblem in Graphen

REACH Problem:

Gibt es in einem Graphen **G** einen Weg vom Startknoten **s** zum Zielknoten **t**?

Beispiel: Bahnfahrt

- Nichtakzeptierende Berechnung:

  $\Longrightarrow$ Vom Heimatbahnhof nach Rot a.d. Rot

Erreichbarkeitsproblem in Graphen

REACH Problem:

Gibt es in einem Graphen **G** einen Weg vom Startknoten **s** zum Zielknoten **t**?

Beispiel: Bahnfahrt

- Nichtakzeptierende Berechnung:

  $\Longrightarrow$ Vom Heimatbahnhof nach Rot a.d. Rot

- Akzeptierende Berechnung:

  $\Longrightarrow$ Vom Heimatbahnhof nach Memmingen

Erreichbarkeitsproblem in Graphen

REACH Problem:

Gibt es in einem Graphen **G** einen Weg vom Startknoten **s** zum Zielknoten **t**?

Beispiel: Bahnfahrt

- Nichtakzeptierende Berechnung:

  $\Longrightarrow$ Vom Heimatbahnhof nach Rot a.d. Rot

- Akzeptierende Berechnung:

  $\Longrightarrow$ Vom Heimatbahnhof nach Memmingen

Im Folgenden wird ein akzeptierendes Problem behandelt

Ist REACH NL-Vollständig?

Zwei zu klärende Fragen...

1. $REACH \in NL$
   $\Longrightarrow$ Benötigt man für die Berechnung lediglich logarithmischen Platz?

Ist REACH NL-Vollständig?

Zwei zu klärende Fragen...

1. $REACH \in NL$
   $\Longrightarrow$ Benötigt man für die Berechnung lediglich logarithmischen Platz?

2. *REACH* ist vollständig in *NL*
   $\Longrightarrow$ Ist REACH ein "schweres" Problem und lassen sich weitere (alle) auf REACH reduzieren?

1. $REACH \in NL$

Gegeben sei ein Graph G

- **G** besteht aus den Knoten $x_1 \ldots x_n$
- Es existieren ein Start- (**s**) und ein Zielknoten (**t**)
- Die Knoten sind durch gerichtete Kanten verbunden

1. $REACH \in NL$

Gegeben sei ein Graph G

- **G** besteht aus den Knoten $x_1 ... x_n$
- Es existieren ein Start- (**s**) und ein Zielknoten (**t**)
- Die Knoten sind durch gerichtete Kanten verbunden

"Orakel-Eigenschaft" der NTM nutzen um Weg von **s** nach **t** zu finden

1. $REACH \in NL$

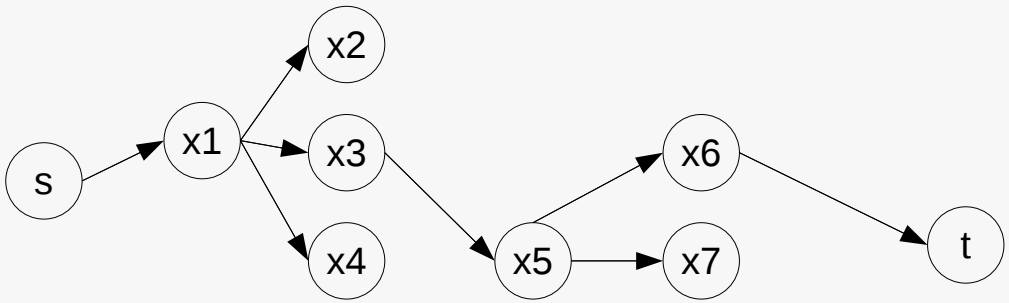
Gegeben sei ein Graph G

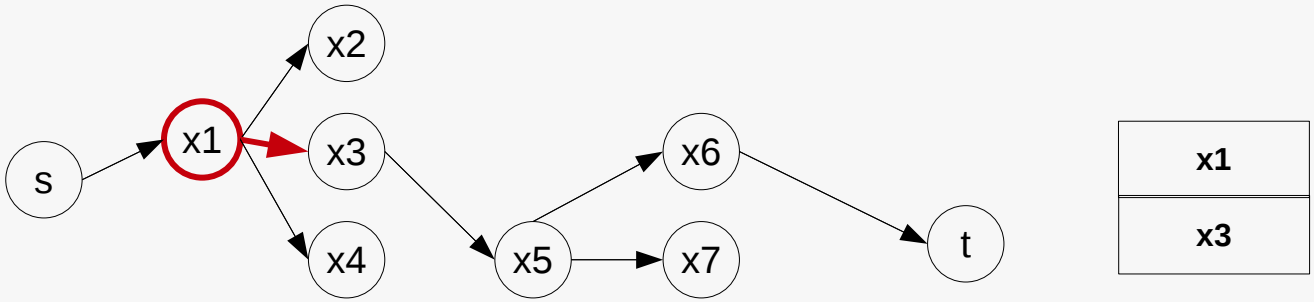
- **G** besteht aus den Knoten $x_1 ... x_n$
- Es existieren ein Start- (**s**) und ein Zielknoten (**t**)
- Die Knoten sind durch gerichtete Kanten verbunden

"Orakel-Eigenschaft" der NTM nutzen um Weg von **s** nach **t** zu finden

Die NTM erhält zwei separate Speicher(bänder):

- Einen für den aktuellen Knoten
- Einen für den nächsten (erratenen) Knoten

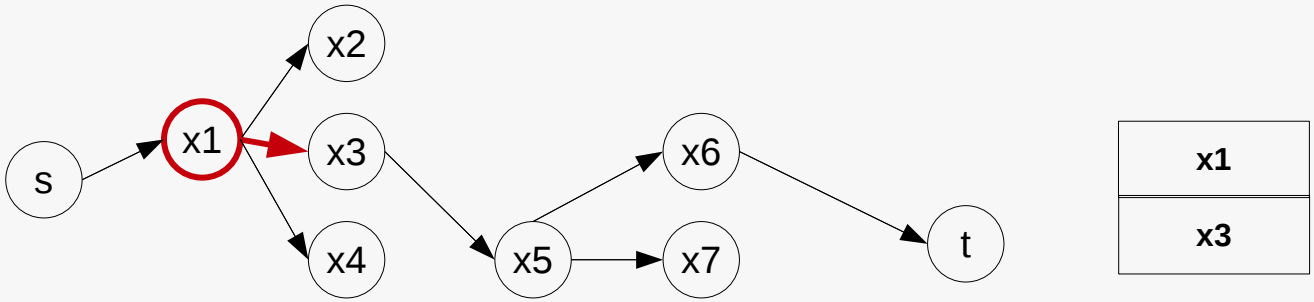




An jedem Knoten wird lediglich...

- der aktuelle
- der "erratene" nächste

Knoten gespeichert!



An jedem Knoten wird lediglich...

- der aktuelle
- der "erratene" nächste

Knoten gespeichert!

Platzbedarf

Jeder Speicher benötigt lediglich $log_2(n)$ an Platz

$$\Longrightarrow REACH \in NL \tag{3}$$

2. *REACH* ist vollständig in *NL*

Lässt sich jede Sprache $A \in NL$ auf REACH reduzieren?

2. *REACH* ist vollständig in *NL*

Lässt sich jede Sprache $A \in NL$ auf REACH reduzieren?

Beweisidee

- Bekannt ist $REACH \in NL$
- Für jede Sprache $A \in NL$ gibt es eine NTM M, welche A akzeptiert

2. *REACH* ist vollständig in *NL*

Lässt sich jede Sprache $A \in NL$ auf REACH reduzieren?

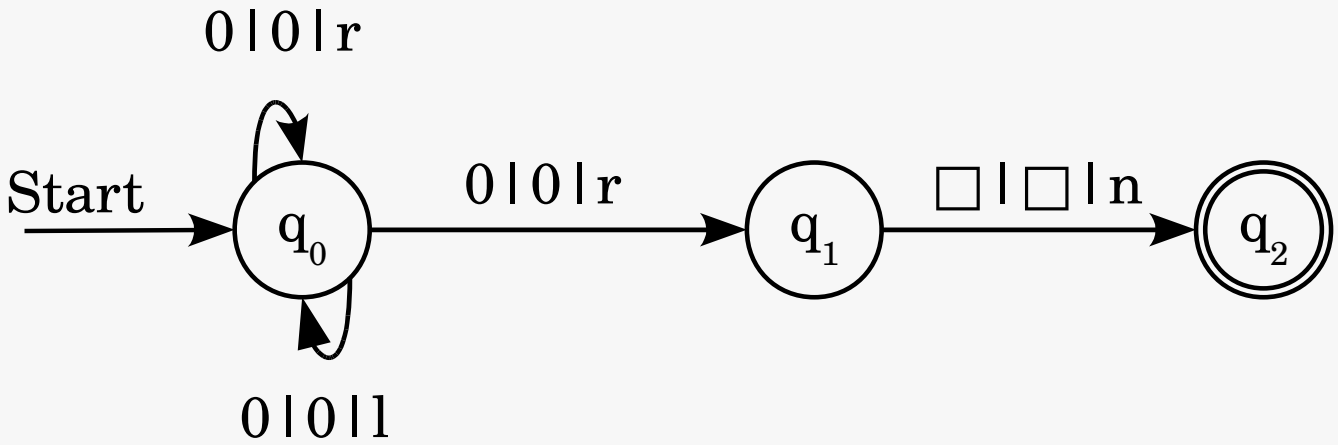
Beweisidee

- Bekannt ist $REACH \in NL$
- Für jede Sprache $A \in NL$ gibt es eine NTM M, welche A akzeptiert

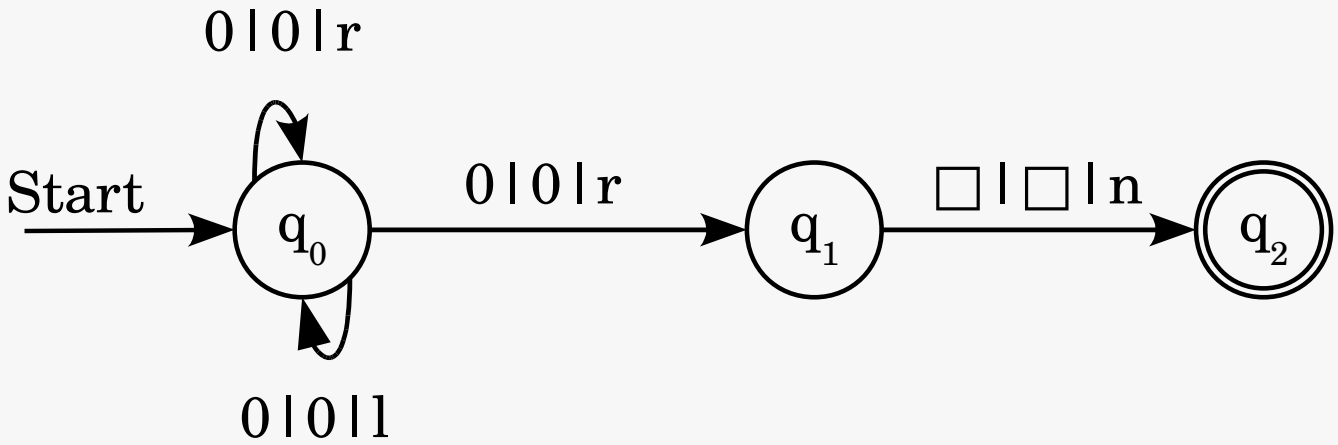
Das Problem wird auf folgende Frage reduziert

Ist ein Eingabewort $x \in A$ bzw. gibt es im Konfigurationsgraphen von M einen Weg von s nach t?

Aufstellen eines Konfigurationsgraphen für eine NTM

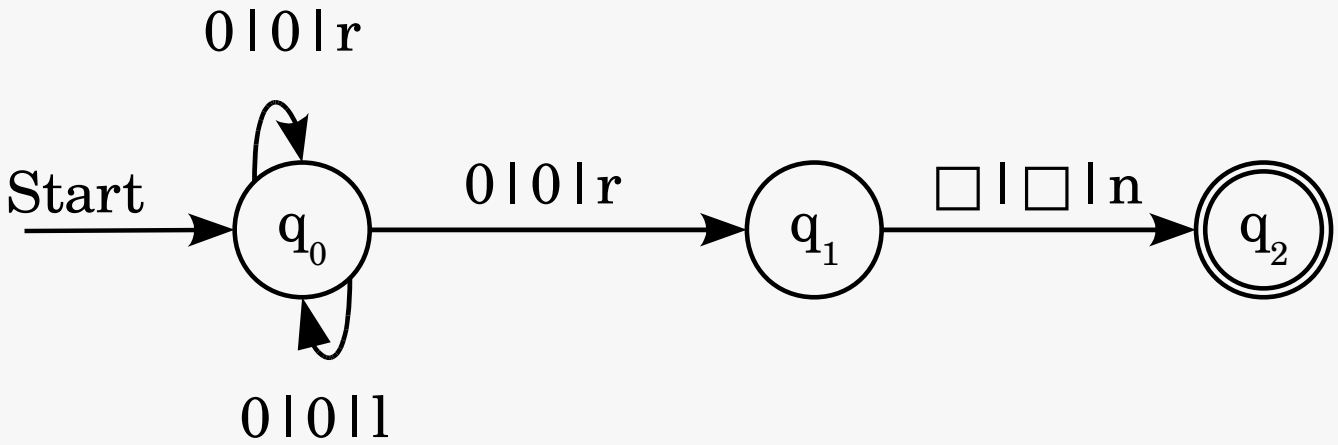


Aufstellen eines Konfigurationsgraphen für eine NTM



Wie sieht der Konfigurationsgraph bei x=00 aus?

Aufstellen eines Konfigurationsgraphen für eine NTM

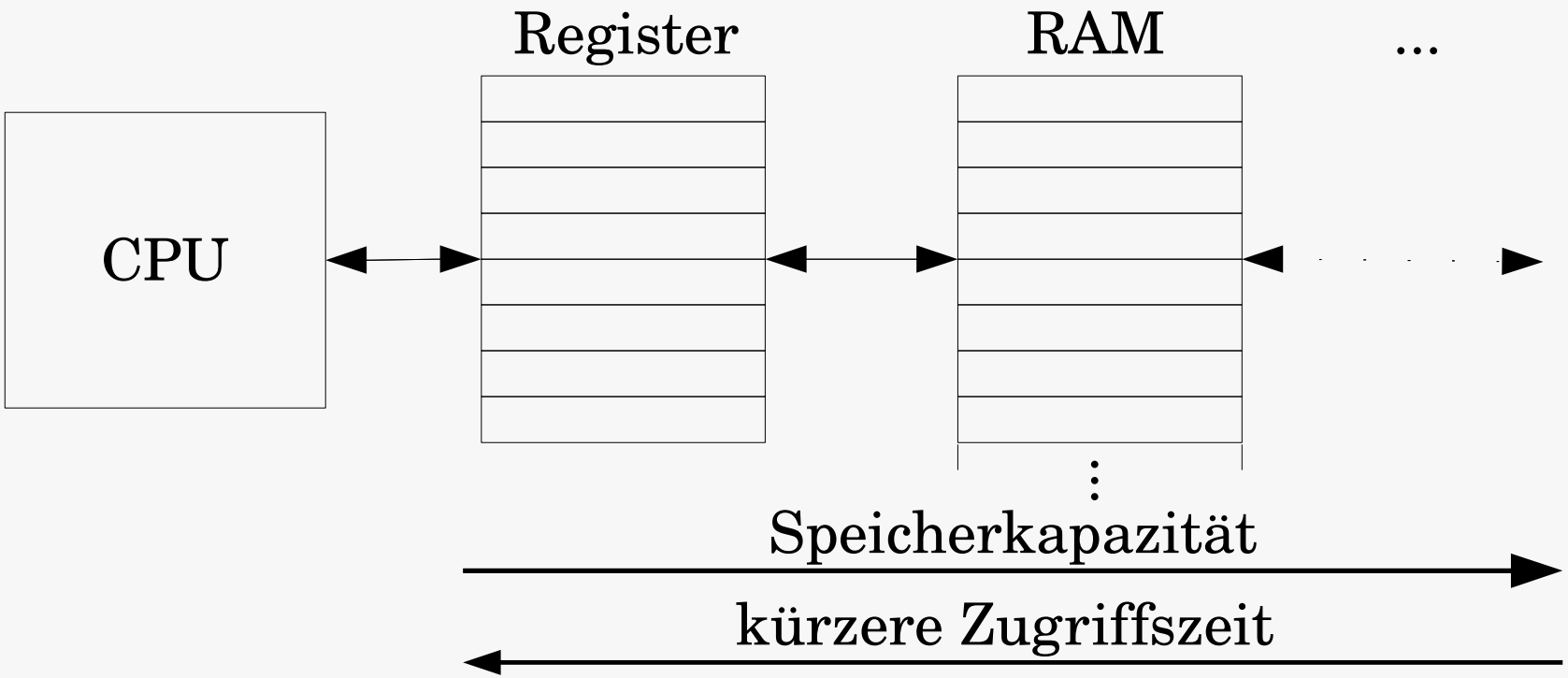


Wie sieht der Konfigurationsgraph bei x=00 aus?

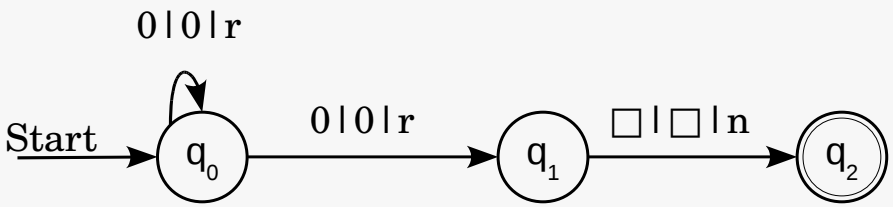
- Wenn $x \in A$ gibt es eine Berechnung die in einer akzeptierenden Konfiguration endet (Pfad von s nach t).
- Erreichbarkeitsproblem (REACH)

Anhang

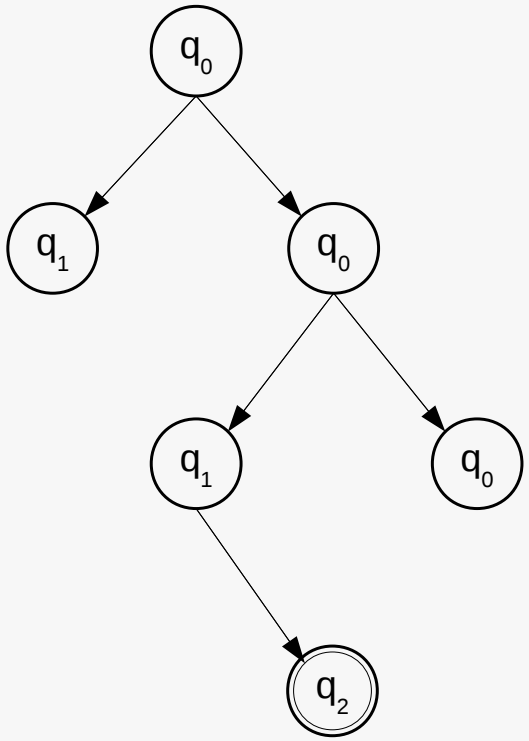
Speicherhirachien im Mikrocontroller:



Berechnungsbaum einer NTM:



Eingabewort: 00

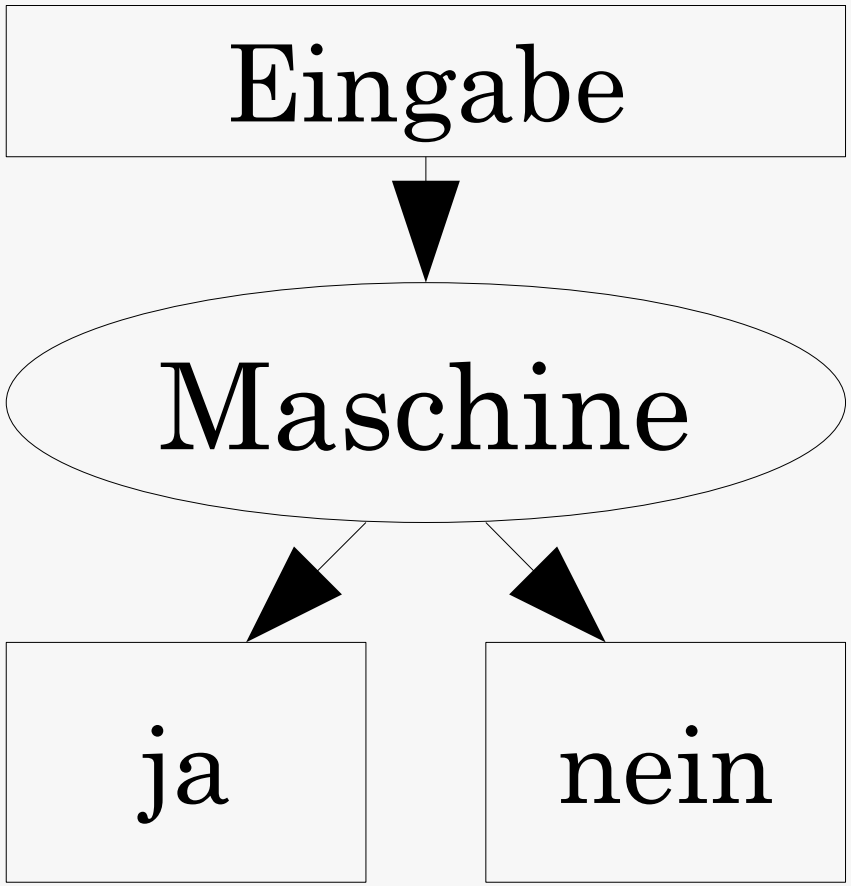


□ ▸ 00□

□0 ▸ 0□

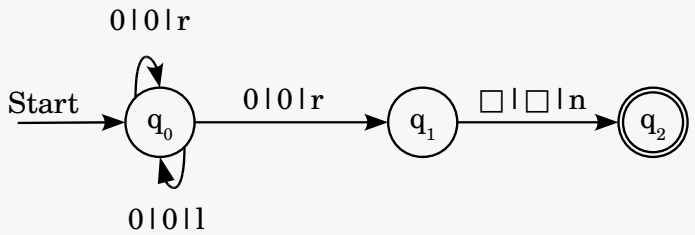
□00 ▸ □

Entscheidungsproblem:



Konfigurationsgraph:

Nichtdeterministische Turing-Maschine M:



Konfigurationsgraph von M bei der Eingabe x=00 :