

Algorithmen und Datenstrukturen

Algorithmen und Datenstrukturen

Foliensatz, 14.11.05

Wintersemester 2005/2006

Algorithmen und Maschinen

Algorithmen und Maschinen

- Algorithmen
 - Rechnen (compute)
 - Änderung der Werte von Attributen eines Modells
 - in mehreren Schritten
 - Anweisungen (*statement*)
 - einzelne Schritte zur Änderung der Werte
 - i.d.R. von maschineller Vorrichtung ausgeführt
 - Berechnung besteht aus mehreren solcher Anweisungen
 - in bestimmter Reihenfolge auszuführen
 - Algorithmus (*algorithm*)
 - Festlegung einer solchen Reihenfolge von Anweisungen
 - bestimmte Wirkung auf Werte der Attribute eines Modells

Algorithmen und Maschinen

- Algorithmus
 - zentrale Begriff in der Informatik
 - Algorithmen können nicht alle Probleme lösen
 - Grenzen des Machbaren
 - Berechenbarkeit
 - mathematische Modelle
 - Beschreiben Anweisungen/Folgen solcher Anweisungen
 - Turingmaschine
 - sehr einfaches, aber dennoch universelles Berechnungsmodell

Algorithmen und Maschinen

- Algorithmus und Berechenbarkeit
 - Algorithmus
 - gibt an, wie Problem gelöst werden kann
 - Berechenbarkeit
 - welche Probleme können überhaupt algorithmisch gelöst werden?

Algorithmen und Maschinen

- Algorithmus
 - Maschinen zum Rechnen
 - genaue Festlegung der jeweiligen elementaren Rechenschritte
 - Anweisungen werden nacheinander ausgeführt
 - Fortgang der Berechnung hängt von Zwischenergebnissen ab
 - eigentliche Information liegt als Datum vor
 - Zahlen- oder Zeichenwerte (Zustand)
 - Bedeutung der Daten ist Maschine nicht bekannt
 - Algorithmus kennt (implizit) Bedeutung der Attribute
 - bestimmt abhängig von Zustand weitere Folge von Rechenschritten
 - Algorithmus
 - macht aus „dummer“ Rechenmaschine „intelligent“ erscheinendes Gerät
 - bringt, salopp formuliert, Geist in die Maschine

Algorithmen und Maschinen

- Algorithmen
 - Berechnungen
 - nach bestimmten Regeln oder Vorschriften
 - in mehreren elementaren Rechenschritten
 - als Algorithmus (*algorithm*) bezeichnet
 - genauere Definition des Begriffs Algorithmus
 - welche Probleme sollen mit Algorithmen gelöst werden?
 - Eingabedaten in Ausgabedaten umwandeln
 - Daten liegen allgemein in Form von Zeichenketten vor
 - auch als mathematische Funktion beschreibbar
 - Algorithmen auch als Abbildungsvorschriften auffassbar
 - gegenüber realen Verhältnissen vereinfachte Auffassung
 - zunächst übernehmen

Algorithmen und Maschinen

- Algorithmen
 - Beispiele
 - Multiplikation ($27 \cdot 33$) ist Abbildung der Zeichenfolgen '2','7' und '3','3' in die Zeichenfolge '8','9','1'.
 - Ein Übersetzer übersetzt die Zeichenfolge 'H', 'U', 'N', 'D' in die Zeichenfolge 'D', 'O', 'G'.
 - Ein Mustererkenner übersetzt die Zeichenfolge 'X', 'X', 'X', 'X', '_', '_', 'X', '_', '_', 'X', '_', '_', 'X', 'X', 'X', '_', '_', '_', '_', 'X', '_', '_', '_', 'X', 'X', 'X', 'X', '_' in die Zeichenfolge '3'. Wieso?

Algorithmen und Maschinen

- Algorithmen
- Beispiele
 - Ein Mustererkenner übersetzt die Zeichenfolge 'X', 'X', 'X', 'X', '_', '_', 'X', '_', '_', 'X', '_', '_', 'X', 'X', 'X', '_', '_', '_', '_', 'X', '_', '_', '_', 'X', 'X', 'X', 'X', '_' in die Zeichenfolge '3'. Wieso?

```
XXXXX
  X
  X
XXX
  X
  X
XXX
```


Algorithmen und Maschinen

- Algorithmen
 - Alphabet
 - jede (geordnete) Menge von Zeichen
 - nicht nur Buchstaben des lateinischen Alphabets: 'A', 'B',...
 - auch z.B.
 - Ziffern: '0', '1', '2', '3', '4', '5', '6', '7', '8', '9',
 - Sonderzeichen: '+', '-', '\$', '°', '±', '(', ')',
 - auch ohne Hochkommata geschrieben, wenn einzelne Zeichen eindeutig
 - Ziffern = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9,},
 - Sonderzeichen = { +, -, \$, °, ±, (,)'}
 - Zeichenfolge in doppelte Hochkommata
 - "27", "33" ≠ "891"
 - "HUND" → "DOG"
 - "XXXX--X--X--XXX----X---XXXX-" → "3".

Algorithmen und Maschinen

- Rechnen
 - Abbildung zwischen Zeichenfolgen
 - klassisches Rechnen Manipulieren von Zeichenfolgen
 - jede Manipulation von Zeichenfolgen heißt Rechnen
 - Rechnen sehr viel allgemeinere Bedeutung als üblich
 - einfach durch Tabelle realisierbar
 - jedem Eingabewert ein Ausgabewert zugeordnet
 - nur für endlich viele Fälle möglich
$$D_1 \rightarrow W_1$$
$$D_2 \rightarrow W_2$$

...

$$D_N \rightarrow W_N$$
 - Nicht für unbeschränkt/sehr viele Eingabewerte

Algorithmen und Maschinen

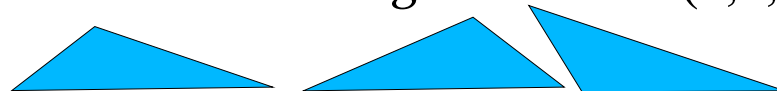
- Rechnen
- Beispiel
 - Tabelle der Produkte aller Zahlen zwischen eins und hundert
 - leicht zu erstellen
 - Wie viele Einträge hätte diese Tabelle?
 - Tabelle der Produkte aller Zahlen zwischen eins und einer Million
 - nicht ebenso erstellt werden?
 - 2 Milliarden Werte (32 Bits)
 - jedes mögliche Produkt verschiedener Zahlen jemals berechnet?
 - Fakultäten
 - 1! → 1
 - 2! → 2
 - 3! → 6
 - 4! → 24
 - ...
 - 20! → 2432902008176640000
 - Fibonacci-Zahlen?

Algorithmen und Maschinen

- Algorithmus heißt
 - abstrakt
 - im Prinzip unbeschränkte Klasse von Problemen
 - statische Finitheit
 - endliche Beschreibung jedes Algorithmus
 - Programm (program)
 - Formulierung eines Algorithmus für spezielle Rechenmaschine
 - dynamische Finitheit
 - während Ablauf nur endliche Menge an Werten (Ressourcen)
 - terminieren
 - Algorithmus endet nach endlicher Anzahl von Schritten
 - nicht terminierende Algorithmen
 - Betriebssysteme
 - Maschinensteuerungen
 - Teilaufgaben

Algorithmen und Maschinen

- Algorithmus heißt
 - determiniert oder funktional
 - bei gleichen Eingabedaten stets gleiche Ergebnisse
 - wird von meisten Algorithmen verlangt
 - entspricht den Vorgaben (Spezifikation)
 - Produkt zweier Zahlen liefert genau einen Wert, auf genau eine Weise dargestellt
 - zufällig oder randomisiert
 - unterschiedliche Ergebnisse
 - Ergebnis eine von mehreren richtigen Lösungen
 - Algorithmus findet zufällig eine Lösung und präsentiert diese
 - Suche Bruder von Claus: Albert oder Berthold
 - Finde eine Quadratwurzel aus 4: +2 oder -2
 - Dreieck aus drei Seiten im Längenverhältnis (2,3,4):



- Zufallszahlengenerator

Algorithmen und Maschinen

- Algorithmus heißt
 - deterministisch, eindeutig
 - im gleichen Zustand immer auf gleiche Weise fortgesetzt
 - liefert natürlich stets die gleichen Ergebnisse → stets determiniert
 - Indeterministisch
 - bei gleichen Datenwerten verschiedene Nachfolgeschritte
 - Auswahl der Schritte
 - zufällig
 - alle möglichen Fortsetzungen des Algorithmus gleichzeitig parallel ausgeführt
 - teilweise für theoretische Untersuchungen benötigt
 - in der Praxis heute keine große Rolle

Algorithmen und Maschinen

- zwei Algorithmen heißen
 - äquivalent
 - bei jeweils gleichen Eingaben stets gleiche Ergebnisse
 - Frage nach besserem/bestem Algorithmus
 - Bewertung von Algorithmen
 - Algorithmus heißt besser, wenn er
 - schneller
 - Rechnung in kürzerer Zeit erledigt
 - kürzer
 - weniger Speicherplatz für Darstellung des Algorithmus
 - weniger Speicherplatz während der Ausführungszeit
 - meist nicht allgemein berechenbar
 - andere quantifizierbare Bedingungen
 - genauer
 - stabil (Sortierverfahren)
 - *Komplexität* eines Algorithmus

Algorithmen und Maschinen

- Berechenbarkeit
 - von großer theoretischer Bedeutung, was man berechnen kann
 - Probleme, für die es i.allg. keine Lösung gibt
 - praktische Bedeutung
 - nicht berechenbares Problem gar nicht erst mit algorithmischen Methoden angehen
 - Berechnungsverfahren entwickeln für möglichst viele Probleme
 - Grenze, was überhaupt noch zu berechnen ist

Algorithmen und Maschinen

- Berechenbarkeit
 - These von Church
 - praktisch nur ein Berechenbarkeitsbegriff
 - alle bisher entwickelten Berechnungsverfahren lösen gleiche Problemklasse
 - es gibt Probleme, die von keinem Algorithmus berechnet werden können
 - alle lösbaren Probleme können mit herkömmlichen Berechnungsverfahren gelöst werden
 - es gibt Probleme, die auch mit den größten Computern nicht behandelt werden können
 - Klasse von Problemen unbekannt, ob berechenbar
 - gewisse Problemklassen nur für sehr kleine Werte lösbar (mit sehr wenigen Parametern/sehr kleinen numerischen Werten)

Algorithmen und Maschinen

- Berechenbarkeit
 - konkreter Berechenbarkeitsbegriff
 - bestimmtes Maschinenmodell
 - Turingmaschine
 - sämtliche berechenbare Probleme Turing-berechenbar
 - anderes Maschinenmodell löst gleiche Problemklasse
 - Maschinenmodell löst sämtliche Turing-berechenbaren Probleme
 - Annahmen
 - bestimmte Ressourcen nicht beschränkt
 - Speicher für Daten und Programme
 - Zeit

Algorithmen und Maschinen

- Bemerkungen zur Berechenbarkeit
 - Andere Verfahren zur Berechnung von Problemen
 - Gödel
 - Regeln zur Konstruktion komplexerer Funktionen aus einfacheren
 - Unvollständigkeitstheorem (1931)
 - (Hilberts Entscheidungsproblem)
 - kein Algorithmus, der für jede Aussage über die natürlichen Zahlen entscheidet, ob diese wahr oder falsch ist.
 - Church
 - funktionales System, Lambda-Kalkül
 - Postsche Ersetzungssystem
 - Turingmaschinen mit mehreren Bändern
 - Alle Verfahren 'äquivalent'
 - Beweis durch 'Emulation' einer Maschine auf anderer

Algorithmen und Maschinen

- Turing-berechenbar
 - Problem heißt Turing-berechenbar, wenn es einen Algorithmus gibt, der eine vorgegebene Zeichenkette in genau eine andere Zeichenkette umwandelt, welche von der vorgegebenen abhängt
 - "123+456" → "579"
 - schrittweise Umwandlung
 - für jeden Schritt eigene Berechenbarkeit definiert
 - Turing-berechenbarer Algorithmus
 - hält nach endlich vielen Schritten an
 - liefert ein richtiges Ergebnis

Algorithmen und Maschinen

- Turing-berechenbar
 - gibt es Algorithmus, der sämtliche Probleme löst?
 - für spezielles Problem kann es keinen Algorithmus geben, der dieses Problem für alle Eingaben lösen kann.
 - Selbstanwendung
 - kann Algorithmus sich selbst analysieren
 - Halteproblem
 - Turingmaschine, die zu jeder vorgegebenen Turingmaschine entscheiden soll, ob sie jemals mit einem Ergebnis anhält.
 - a) analysierende Maschine so konstruieren, dass sie genau dann nicht anhält, wenn die untersuchte Maschine anhält
 - b) untersucht diese Maschine sich selbst, so liefert sie das Ergebnis, dass sie genau dann anhält, wenn sie nicht anhält, also einen Widerspruch.
 - c) Halteproblem nicht allgemein lösbar ist,
 - d) **es gibt Problemklassen, die nicht berechnet werden können.**

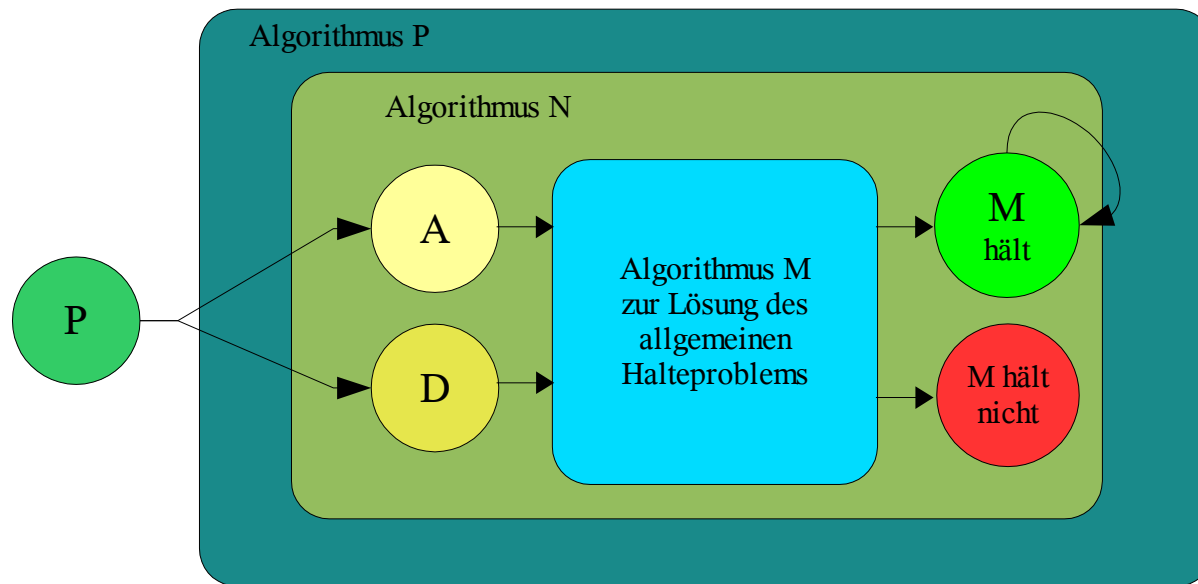
Algorithmen und Maschinen

● Halteproblem

- A ein Algorithmus, D Daten für diesen Algorithmus
 - A sei in Form eines Programms gegeben selbst Zeichenkette
 - ebenso Daten D sind Zeichenkette
- $M[A,D]$ ein Algorithmus, der Halteproblem löst
 - M hält in genau einem von zwei Zuständen, für irgendeinen Algorithmus A und irgendwelche Daten D
 - 1. Zustand: A angesetzt auf D hält (kürzer: $[A, D]$ HÄLT)
 - 2. Zustand: A angesetzt auf D hält nicht ($[A, D]$ HÄLT NICHT)
- Transformieren der Maschine in eine andere
 - $N[A]$ hat nur noch einen Eingabewert: $N[A] \equiv M[A,A]$
 - $P[A]$ hat statt des Haltzustands unendlichen Zyklus: $P[A]$
 - Was geschieht nun mit $P[P]$?
- $P[P]$ hält nicht an $\Leftrightarrow P[P]$ hält an: Widerspruch!
 - $P[...]$ kann es nicht geben kann.
 - $P[..]$ kanonisch aus $N[...]$ konstruiert $\Rightarrow N[...]$ kann es nicht geben
 - $N[...]$ kanonisch aus aus $M[...]$ konstruiert: $\Rightarrow M[...]$ kann es nicht geben,
 - **Es gibt keinen Algorithmus, der das allgemeine Halteproblem löst!**

Algorithmen und Maschinen

- Halteproblem
 - $P[A] = M[A,A]$, d.h. $P[A]$ hält $\Leftrightarrow A[A]$ hält nicht



Algorithmen und Maschinen

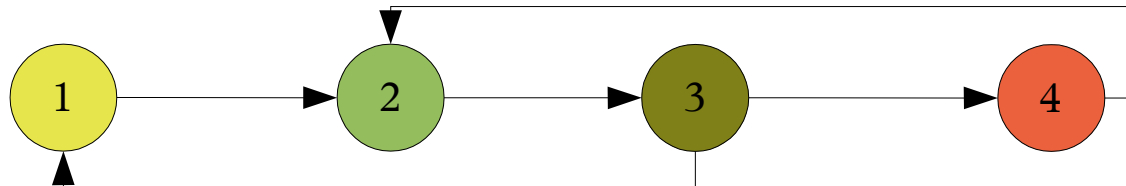
- weitere nicht berechenbare Problemklassen
 - Erreichbarkeitsproblem
 - wird beliebig vorgegebene Stelle beim Ablauf eines Algorithmus jemals erreicht wird
 - Totalitätsproblem
 - hält für alle Eingaben an
 - Äquivalenzproblem
 - gleiche Berechnung zweier beliebiger verschiedene Algorithmen für alle Eingabedaten
 - Probleme nicht allgemein für alle Algorithmen lösbar
 - für Spezialfälle dieser Problemklassen kann es immer noch eine Lösung geben, die auch effektiv realisiert werden kann.

Algorithmen und Maschinen

- Verfeinerung der Berechenbarkeit
- Halteproblem in zwei Teile zerlegbar
 - ein Algorithmus hält an oder
 - Antwort bleibt unbestimmt
 - z.B. weil der Analysealgorithmus niemals zu einem Ende kommt
 - partiell berechenbar
 - weder Totalitätsproblem noch Äquivalenzproblem partiell berechenbar
- wie lange dauert es, bis Problem gelöst ist?
 - einige Probleme im Prinzip berechenbar
 - zu große Laufzeit
 - praktisch überhaupt nicht berechenbar
 - Komplexitätstheorie
 - später genauer untersucht

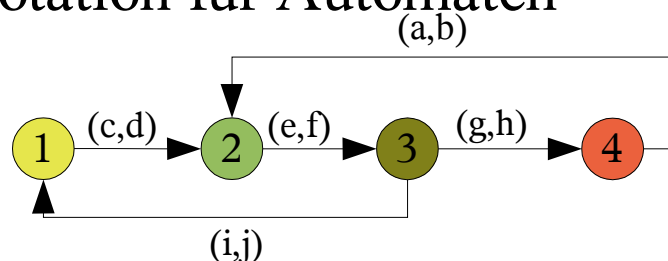
Algorithmen und Maschinen

- Mathematische Berechnungsmodelle
 - Der endliche deterministische Automat
 - ein mathematisches Modell für System
 - Merkmale verschiedene Ausprägungen: Zustände (state)
 - meist nur ein Merkmal
 - Wartende in 'Warteschlange'
 - Betriebsphase einer Waschmaschine
 - Werte durch endliche Menge repräsentiert
 - endliche Teilmenge der natürlichen Zahlen
 - keine Einschränkung gegenüber allgemeineren Modellen
 - zwei Attribute à zwei Merkmalen: (1,1), (1,2), (2,1), (2,2)
 - ein Attribut à vier Merkmalen: (1), (2), (3), (4)
 - äquivalent
 - 2. Form einfacher in Zuständen darstellbar



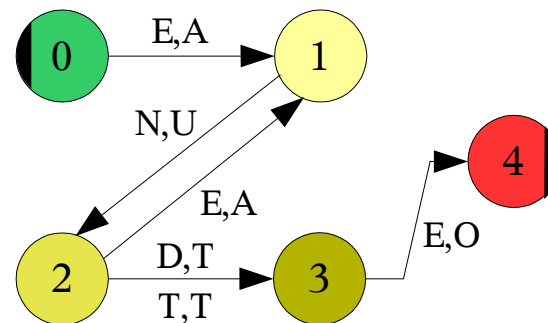
Algorithmen und Maschinen

- Mathematische Berechnungsmodelle
 - Der endliche deterministische Automat
 - Automat 'Modell' eines Systems
 - System reagiert (in bestimmtem Zustand) auf externe Ereignisse
 - Reaktion in Form einer „Ausgabe“
 - Reaktion in Form einer Zustandsänderung
 - weitere Eingaben erzeugen weitere Ausgaben/Zustandswechsel
 - Verhalten hängt ab von
 - augenblicklichem Zustand des Systems
 - Eingabe
 - Graphische Notation für Automaten



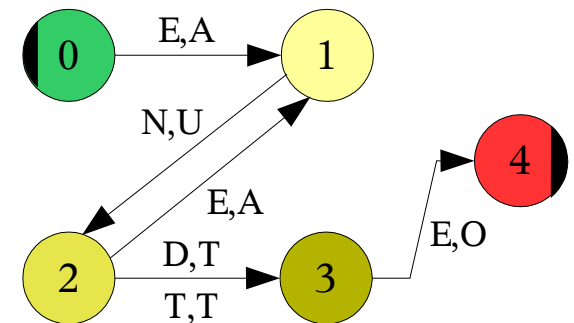
Algorithmen und Maschinen

- Mathematische Berechnungsmodelle
 - Automatentypen
 - Besondere Zustände
 - Anfangszustand
 - Endezustand
 - *erkennende Automaten*
 - Folge eingegebener Zeichen gehört zu vorgegebenen Menge von Zeichenfolgen
 - *Übersetzungsautomat*
 - ordnet erkanntem Wort anderes Wort zu
 - zu jedem Eingabezeichen wird ein Ausgabezeichen erzeugt



Algorithmen und Maschinen

- Mathematische Berechnungsmodelle
 - Anwendungen von Automaten
 - Entwurf von Hardwareschaltungen
 - Ablaufplanung von Programmen
 - Steuerung einer Turingmaschine
 - Formale Spezifikation eines Automaten
 - $A = (\text{Zustände, Eingabe, Ausgabe, Übergänge, Anfang, End})$
 - $A = (Z = \{0,1,2,3,4\},$
Ein = $\{A, \dots, Z\},$
Aus = $\{A, \dots, Z\},$
 $\ddot{U} = \{(0, E, A, 1), (1, N, U, 2), (2, E, A, 1),$
 $(2, D, T, 3), (2, T, T, 3), (3, E, O, 4)\},$
Anf = 0, End = 4).



Algorithmen und Maschinen

- Mathematische Berechnungsmodelle

- Eigenschaften von Automaten

- endlich

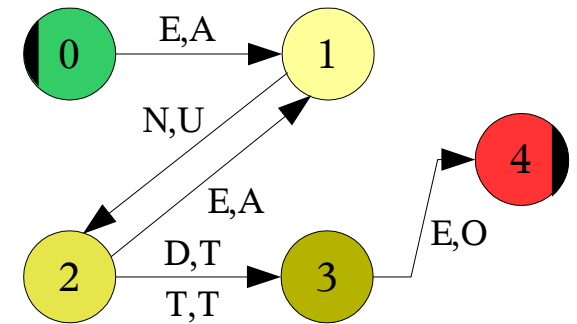
- endlich viele Zustände
- Eingabealphabet endlich

- deterministisch

- in jedem Zustand zu einer Eingabe genau ein Folgezustand festgelegt

- unvollständig

- zu Zustand und möglichem Eingabewert kein Nachfolgezustand definiert
- ggf. führt jeder nicht definierte Übergang in einen speziellen Zustand (Fehler, nicht erkannt)
- implizit alle Automaten zumindest vollständig



Algorithmen und Maschinen

- Mathematische Berechnungsmodelle

- Eigenschaften von Automaten

- indeterministisch

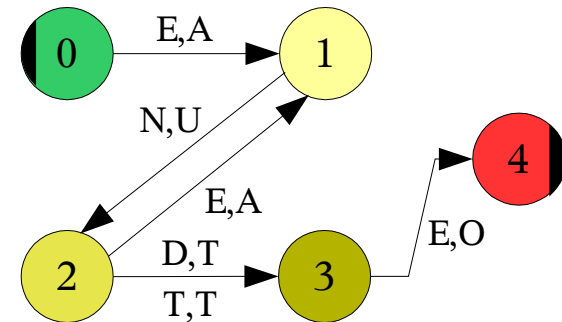
- zu Zustand und möglichem Eingabewert mehrere Nachfolgezustände definiert

- umfangreiche Theorie der Automaten

- äquivalent zu minimalem endlichen deterministischen Automaten

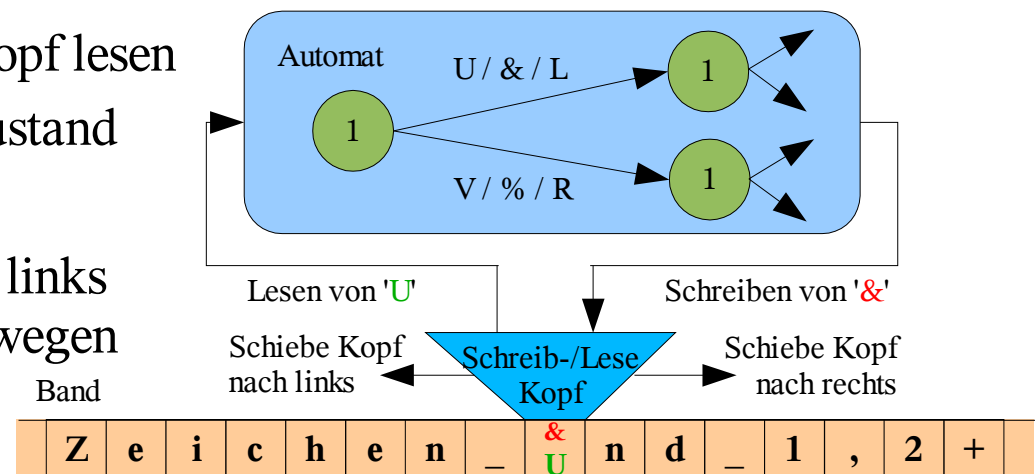
- Klasse der erkennbaren Wörter genau spezifizierbar

- regulären Sprachen



Algorithmen und Maschinen

- Mathematische Berechnungsmodelle
 - Turingmaschine (1936, englischer Mathematiker A.M. Turing)
 - elementares Berechnungsmodell
 - Automaten kontrolliert Funktionalität
 - Speicher in Form eines langen Bandes
 - Schreib/Lese-Kopf befindet sich an einer Stelle des Bandes
 - Berechnungsschritt
 - Wert unter Schreib/Lese-Kopf lesen
 - abhängig von Automatenzustand
 - neuen Wert schreiben
 - Schreib/Lese-Kopf nach links oder rechts auf Band bewegen



Algorithmen und Maschinen

- Mathematische Berechnungsmodelle
 - Turingmaschine zur Addition zweier Dualzahlen
 - Automat

