

Übersicht

- 1 Einführung
 - 1.1 Organisatorisches
 - 1.2 Motivation
 - 1.3 Empfohlene Literatur
 - 1.4 Alphabete, Wörter, Sprachen

Inhalt der Vorlesung

Verarbeitung von Sprachen

- Charakterisierung von Sprachen
- (Automatische) Erkennung, ob Wörter zur Sprache gehören

Wesentliche Formalismen zur Darstellung von Sprachen:

- Automaten
- Grammatiken

1. einleitendes Beispiel: Automaten

Betrachte folgendes Problem:

Eingabe: Ein String w aus 0en und 1en

Frage: Sind diese beiden Eigenschaften erfüllt?

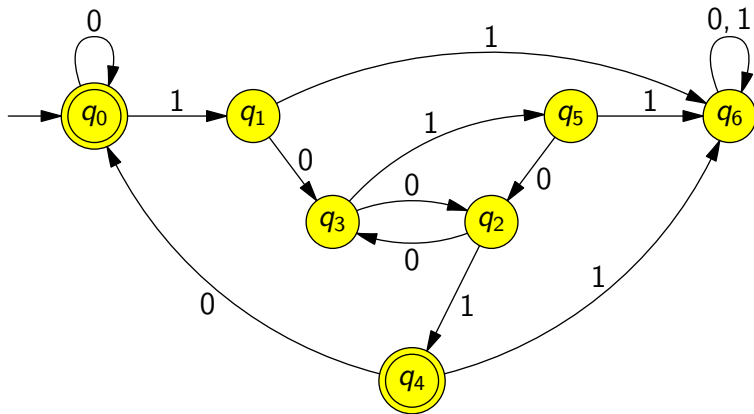
- Es kommt 11 nicht als Unterwort in w vor.
- Als Binärzahl ist w durch drei teilbar.

Beispiele: 0101, 1001, 00110, 0101010

Gesucht:

Ein Programm, das w bekommt und *true* oder *false* zurückgibt.

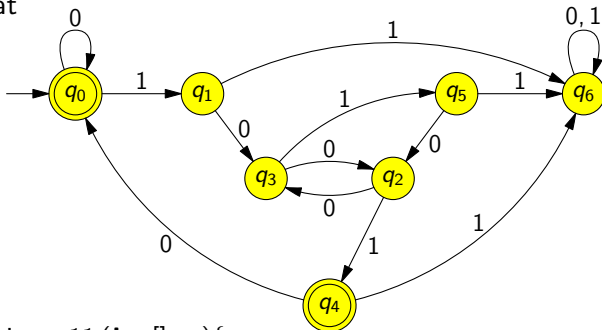
Lösung: Endlicher Automat



1 Einführung

1.2 Motivation

Endlicher Automat
kann leicht durch
Programm
simuliert
werden



```

static boolean drei_not_11 (int [] w){
    boolean[] finalState = {true, false , false , false , true , false , false };
    int [][] delta = {{0,1}, {3,6}, {3,4}, {2,5}, {0,6}, {2,6}, {6,6}};
    int q = 0;

    for (int i = 0; i < w.length; i++) q = delta[q][w[i]];

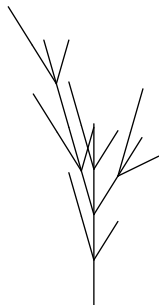
    return finalState [q];
}
  
```

2. einleitendes Beispiel: Grammatiken

Zeichenprogramm, das diese Befehle kennt:

- F: Zeichne eine kurze Linie.
- -: Drehe dich ein wenig nach rechts.
- +: Drehe dich ein wenig nach links.
- [: Merke dir die augenblickliche Position und Richtung.
-]: Kehre zur letzten gemerkten Position und Richtung zurück.

F [+FF] [--F] F [+F [+FF] [--F] FF [+FF]
[--F] F] [--F [+FF] [--F] F] F [+FF] [--F] F



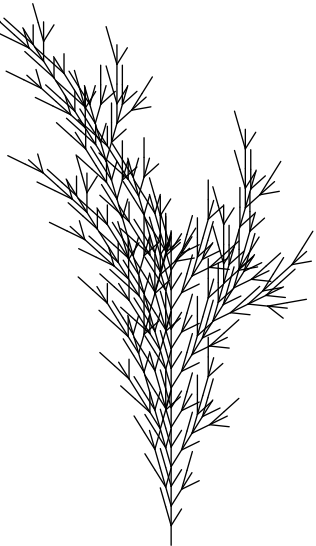
1 Einführung

1.2 Motivation

```

F[+FF] [--F]F[+F[+FF] [--F]FF[+FF] [--F]F] [--F[+FF] [--F]F]F[+FF]
[--F]F[+F[+FF] [--F]F[+F[+FF] [--F]FF[+FF] [--F]F] [--F[+FF] [--
]F]F[+FF] [--F]FF[+FF] [--F]F[+F[+FF] [--F]FF[+FF] [--F]F] [--F[
+FF] [--F]F]F[+FF] [--F]F] [--F[+FF] [--F]F[+F[+FF] [--F]FF[+FF] [
--F]F] [--F[+FF] [--F]F]F[+FF] [--F]F]F[+FF] [--F]F[+F[+FF] [--F
FF[+FF] [--F]F] [--F[+FF] [--F]F]F[+FF] [--F]F[+F[+FF] [--F]F[+F[
+FF] [--F]FF[+FF] [--F]F] [--F[+FF] [--F]F]F[+FF] [--F]F[+F[+FF] [
--F]F[+F[+FF] [--F]FF[+FF] [--F]F] [--F[+FF] [--F]F]F[+FF] [--F]F
F[+FF] [--F]F[+F[+FF] [--F]FF[+FF] [--F]F] [--F[+FF] [--F]F]F[+FF
] [--F]F] [--F[+FF] [--F]F[+F[+FF] [--F]FF[+FF] [--F]F] [--F[+FF] [
--F]F]F[+FF] [--F]F]F[+FF] [--F]F[+F[+FF] [--F]FF[+FF] [--F]F] [
-F[+FF] [--F]F]F[+FF] [--F]FF[+FF] [--F]F[+F[+FF] [--F]FF[+FF] [
--F]F] [--F[+FF] [--F]F]F[+FF] [--F]F[+F[+FF] [--F]F[+F[+FF] [--F]
FF[+FF] [--F]F] [--F[+FF] [--F]F]F[+FF] [--F]F[+F[+FF] [--F]F[+F[+
FF] [--F]FF[+FF] [--F]F]F[+FF] [--F]F[+F[+FF] [--F]FF[+FF] [--F]F
]F[+FF] [--F]F[+F[+FF] [--F]FF[+FF] [--F]F] [--F[+FF] [--F]F]F[+F
F[+FF] [--F]F] [--F[+FF] [--F]F[+F[+FF] [--F]FF[+FF] [--F]F] [--F[+
FF] [--F]F]F[+FF] [--F]F[+F[+FF] [--F]F[+F[+FF] [--F]FF[+FF] [--F]
F]F[+FF] [--F]F]F[+FF] [--F]F[+F[+FF] [--F]FF[+FF] [--F]F]F[+FF]
[--F[+FF] [--F]F]F[+FF] [--F]F]F[+FF] [--F]F[+F[+FF] [--F]FF[+FF]
F]F[+FF] [--F]F]F[+FF] [--F]F]F[+FF] [--F]F]F[+FF]

```



Starte mit F und wende $F \mapsto F[+FF][--F]F$ an!

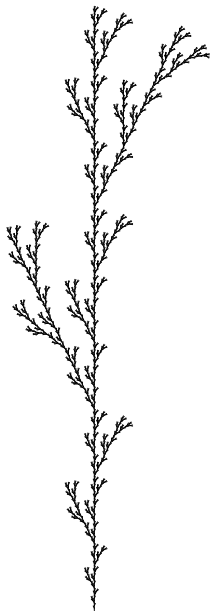
1 Einführung

1.2 Motivation



1 Einführung

1.2 Motivation



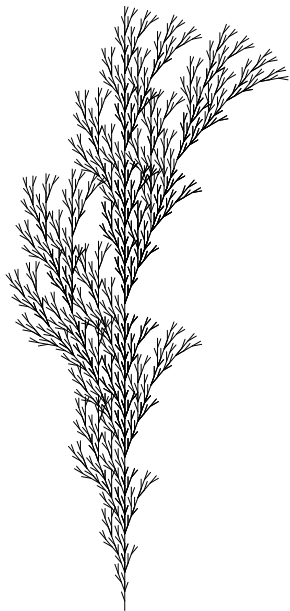
$$n = 5$$

$$\delta = 25.7$$

$$F \mapsto F[+F]F[-F]F$$

1 Einführung

1.2 Motivation



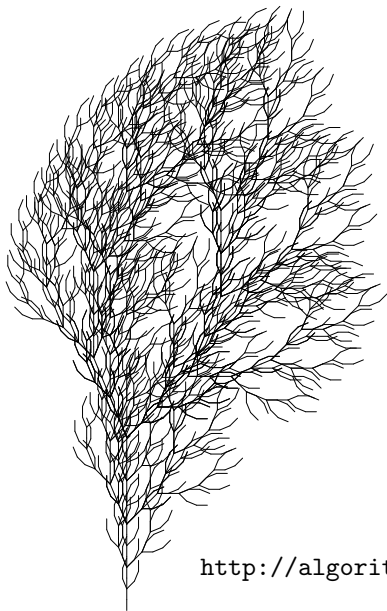
$$n = 5$$

$$\delta = 20$$

$$F \mapsto F[+F]F[-F][F]$$

1 Einführung

1.2 Motivation



$$n = 4$$

$$\delta = 22.5$$

$$F \mapsto$$

$$FF - [-F + F + F] + [+F - F - F]$$

Buch:

P. Prusinkiewicz

A. Lindenmayer

The Algorithmic Beauty of
Plants

<http://algorithmicbotany.org/papers/abop/abop.pdf>