

# 1. Kanonische Ableitung und Reduktion

---

a)

Rechtskanonisch:

Linkskanonisch:

b)

Rechtskanonisch:

$$( ( v + v ) * v / v ) - ( v / *v* ) =>$$

~~$$( ( v + v ) * v / v ) - ( v / *F* ) =>$$~~

~~$$( ( v + v ) * v / v ) - ( F / T ) =>$$~~

$$( ( v + v ) * v / v ) - ( *v* / F ) =>$$

$$( ( v + v ) * v / v ) - ( *F* / F ) =>$$

$$( ( v + v ) * v / v ) - ( *T / F* ) =>$$

$$( ( v + v ) * v / v ) - ( *T* ) =>$$

$$( ( v + v ) * v / v ) - *( E )* =>$$

$$( ( v + v ) * v / v ) - *F* =>$$

$$( ( v + v ) * v / *v* ) - T =>$$

$$( ( v + v ) * *v* / F ) - T =>$$

$$( ( v + *v* ) * F / F ) - T =>$$

$$( ( v + *F* ) * F / F ) - T =>$$

$$( ( *v* + T ) * F / F ) - T =>$$

$$( ( *F* + T ) * F / F ) - T =>$$

$$( ( *T* + T ) * F / F ) - T =>$$

$$( ( *E + T* ) * F / F ) - T =>$$

$$( *( E )* * F / F ) - T =>$$

$$( *F* * F / F ) - T =>$$

$$( *T * F* / F ) - T =>$$

$$( *T / F* ) - T =>$$

$$( *T* ) - T =>$$

$$*( E )* - T =>$$

$$*F* - T =>$$

$$*T* - T =>$$

$$*E - T* =>$$

E

Syntaxbaum:

Linkskanonisch:

$$( ( *v* + v ) * v / v ) - ( v / v ) =>$$

$$( ( *F* + v ) * v / v ) - ( v / v ) =>$$

$$( ( *T* + v ) * v / v ) - ( v / v ) =>$$

$$( ( E + *v* ) * v / v ) - ( v / v ) =>$$

$$( ( E + *F* ) * v / v ) - ( v / v ) =>$$

$$( ( *E + T* ) * v / v ) - ( v / v ) =>$$

$$( *( E )* * v / v ) - ( v / v ) =>$$

$$( *F* * v / v ) - ( v / v ) =>$$

$$( T * *v* / v ) - ( v / v ) =>$$

$$( *T * F* / v ) - ( v / v ) =>$$

$$( T / *v* ) - ( v / v ) =>$$

$$( *T / F* ) - ( v / v ) =>$$

$$( *T* ) - ( v / v ) =>$$

$$*( E )* - ( v / v ) =>$$

$$*F* - ( v / v ) =>$$

$$*T* - ( v / v ) =>$$

$$E - ( *v* / v ) =>$$

$$E - ( *F* / v ) =>$$

$$E - ( T / *v* ) =>$$

$$E \rightarrow ( *T / F* ) =>$$

$$E \rightarrow ( *T* ) =>$$

$$E \rightarrow *( E )* =>$$

$$E \rightarrow *F* =>$$

$$*E \rightarrow T* =>$$

$$E$$

Die Anzahl der Reduktionen ist bei beiden Varianten gleich (25 Ableitungen).

## 2. Mehrdeutigkeit, Beschreibung und Schreibweisen

a)

Bei **frac** ist die Mehrdeutigkeit, da man auf **n** kommen kann, indem man entweder die 1. Alternative verwendet, oder die 2. Alternative und dann **frac** durch  $\epsilon$ .

Beispiel: 6.9

$$\text{real} \Rightarrow \text{mant} \Rightarrow \text{sign int} . \text{frac} \Rightarrow \epsilon \text{ int} . \text{frac} \Rightarrow \epsilon n . \text{frac}$$

option 1	option 2
$\Rightarrow \epsilon n . n$	$\Rightarrow \epsilon n . \text{frac } n$
$\Rightarrow \epsilon 6 . n$	$\Rightarrow \epsilon 6 . \text{frac } n$
$\Rightarrow \epsilon 6 . 9$	$\Rightarrow \epsilon 6 . \epsilon n$
$\Rightarrow \epsilon 6 . 9$	

Änderung: **frac**  $\rightarrow n \mid \text{frac } n \mid \epsilon$  auf **frac**  $\rightarrow \text{frac } n \mid \epsilon$ , rest bleibt gleich.

### b) Äquivalente, eindeutige Grammatik

Möglichst wenige Regeln:

$G(\text{real})$ :

$$\text{real} = [ '+' | '-' ] (0 | \dots | 9) \{0 | \dots | 9\} [ '.' \{0 | \dots | 9\} ] [ 'E' [ '+' | '-' ] (0 | \dots | 9) \{0 | \dots | 9\} ] .$$

"Kürzer":

$$\text{real} = \text{optSign } n \{n\} [ '.' \{n\} ] [ 'E' \text{optSign } n \{n\} ] .$$

$$\text{optSign} = [ '+' | '-' ] .$$

$n = 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 .$

### 3. Reguläre Grammatiken

---

a)

Sätze:

$\epsilon$

ab

abababababab(ab)...

bb

bbbbbbb(b)...

$S \rightarrow A b \mid B b \mid \epsilon . A \rightarrow a \mid C a . C \rightarrow A b . B \rightarrow b \mid B b .$

b)

$ab(ab)^* + b(b)^* + \epsilon$  oder besser  $(ab)^* + (b)^*$  (erspaart das  $\epsilon$ )

### 4. Bezeichner in der Programmiersprache Ada

---

a)

$B \rightarrow l \mid l R$

$R \rightarrow d \mid l \mid \text{'\_'} U \mid l R \mid d R$

$U \rightarrow l \mid d \mid l R \mid d R$

b)

$B \rightarrow l \mid R l \mid R d$

$R \rightarrow U \text{'\_'} \mid R l \mid R d \mid l$

$U \rightarrow R d \mid R l \mid l$

c)

$l ( l + d + \text{'\_'} l + \text{'\_'} d )^*$

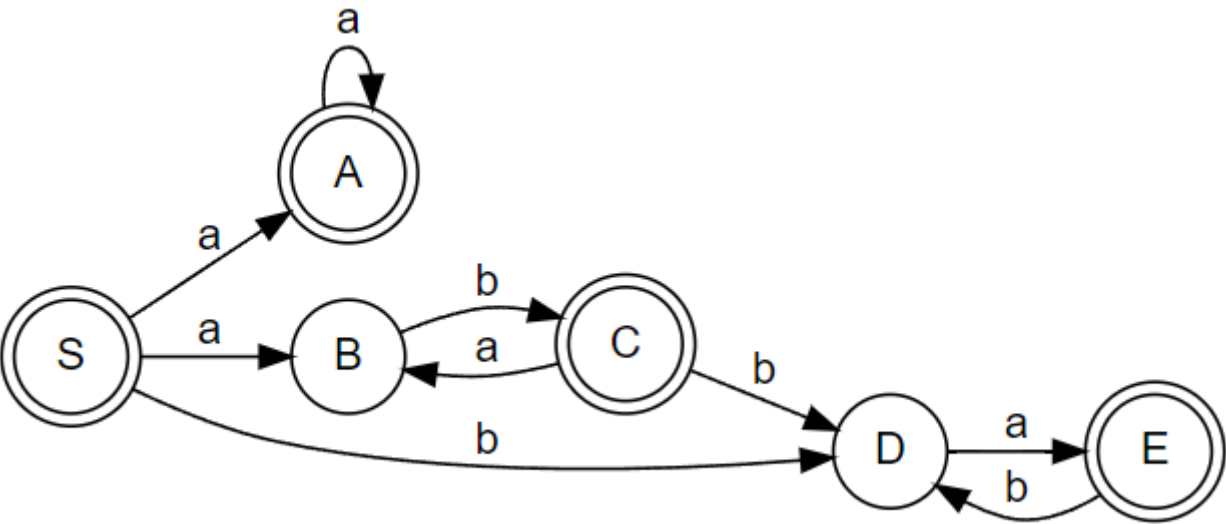
Unix:

$l(l|d|_|_d)^* = ^l(?(l|d))^* = ^l(?[ld])^*$

# 5. Transformation zwischen Darstellungsformen regulärer Sprachen

a)

```
digraph non_deteministic_finite_state_machine {
    fontname="Helvetica,Arial,sans-serif"
    node [fontname="Helvetica,Arial,sans-serif"]
    edge [fontname="Helvetica,Arial,sans-serif"]
    rankdir=LR;
    node [shape = doublecircle]; S A C E;
    node [shape = circle];
    S -> A [label = "a"]
    A -> A [label = "a"]
    S -> B [label = "a"]
    B -> C [label = "b"]
    S -> D [label = "b"]
    C -> D [label = "b"]
    C -> B [label = "a"]
    D -> E [label = "a"]
    E -> D [label = "b"]
}
```



	0	1
-> S	{A, B}	{D}
o A	{A}	-
B	{C}	-

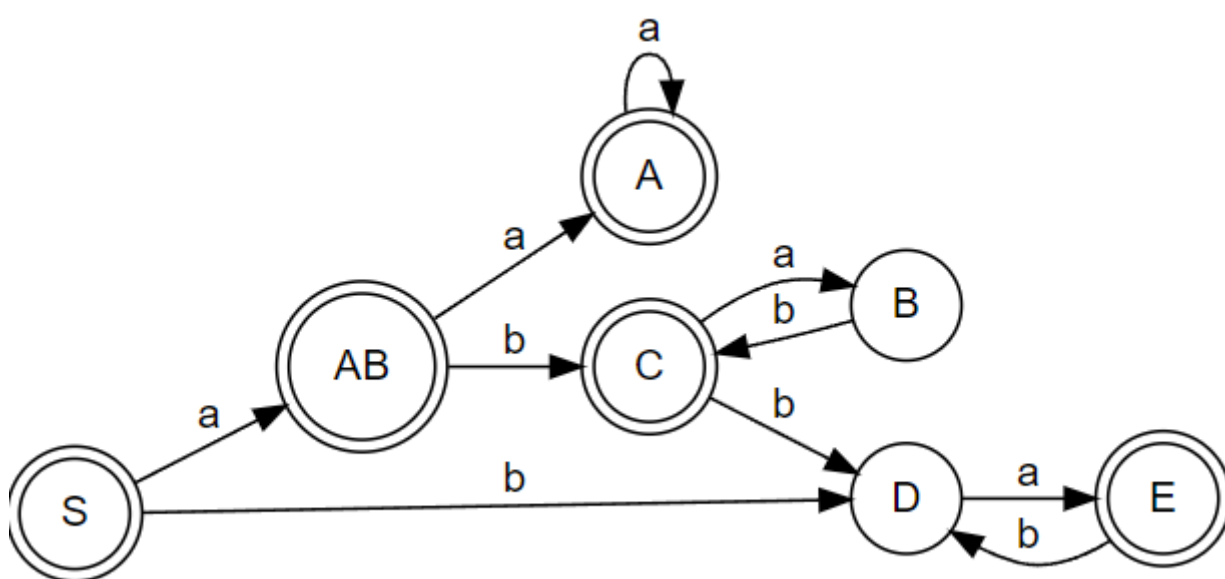
	0	1
o C	{B}	{D}
D	{E}	-
o E	-	{D}
o {A, B}	{A}	{C}

```

digraph deteministic_finite_state_machine {
    fontname="Helvetica,Arial,sans-serif"
    node [fontname="Helvetica,Arial,sans-serif"]
    edge [fontname="Helvetica,Arial,sans-serif"]
    rankdir=LR;
    node [shape = doublecircle]; S A C E AB;
    node [shape = circle];

    S -> AB [label = "a"]
    AB -> A [label = "a"]
    AB -> C [label = "b"]
    A -> A [label = "a"]
    B -> C [label = "b"]
    S -> D [label = "b"]
    C -> D [label = "b"]
    C -> B [label = "a"]
    D -> E [label = "a"]
    E -> D [label = "b"]
}

```



b)

0 anhängen  $\Rightarrow$  \*2 1 anhängen  $\Rightarrow$  \*2 + 1