Da Machine für Primzahlen

Thomas Pajor
thomas.pajor@logn.de

17. Februar 2005

Übersicht

Definition der Primzahlen

Definition

Sei $\mathbb{N}$ die Menge der natürlichen Zahlen.
Eine Zahl $n \in \mathbb{N}$ mit $n > 1$ heißt *prim*, wenn es keine Zahlen $p, q \in \mathbb{N}$ gibt mit $p, q > 1$ so, dass gilt:

$$n = p \cdot q$$

Formulierung als Entscheidungsproblem

- Gegeben: Eine natürliche Zahl $n \in \mathbb{N}$ mit $n > 1$.

Formulierung als Entscheidungsproblem

- Gegeben: Eine natürliche Zahl $n \in \mathbb{N}$ mit $n > 1$.
- Frage: Ist n eine Primzahl?

Formulierung als Entscheidungsproblem

- Gegeben: Eine natürliche Zahl $n \in \mathbb{N}$ mit $n > 1$.
- Frage: Ist n eine Primzahl?
- Sprache: $\mathcal{L} := \{n \mid n \text{ prim}\}$.

Funktionsweise der TM

- Deterministisch

Funktionsweise der TM

- Deterministisch
- Eingabe w soll in binärer Kodierung vorliegen, also $\Sigma = \{0, 1\}$

Funktionsweise der TM

- Deterministisch
- Eingabe w soll in binärer Kodierung vorliegen, also $\Sigma = \{0, 1\}$
- Akzeptierende Endzustände:
  1. q_J, falls w eine Primzahl ist
  2. q_N, sonst

Algorithmus in Pseudocode

```
w = leseZahl(); divisor = 2;

while (divisor < w) {
  k = w;
  while(k > 0)
    k = k - divisor;
  if (k == 0)
    halte(w ist nicht prim);
  divisor = divisor + 1;
}

halte(w ist prim);
```

Algorithmus in Pseudocode

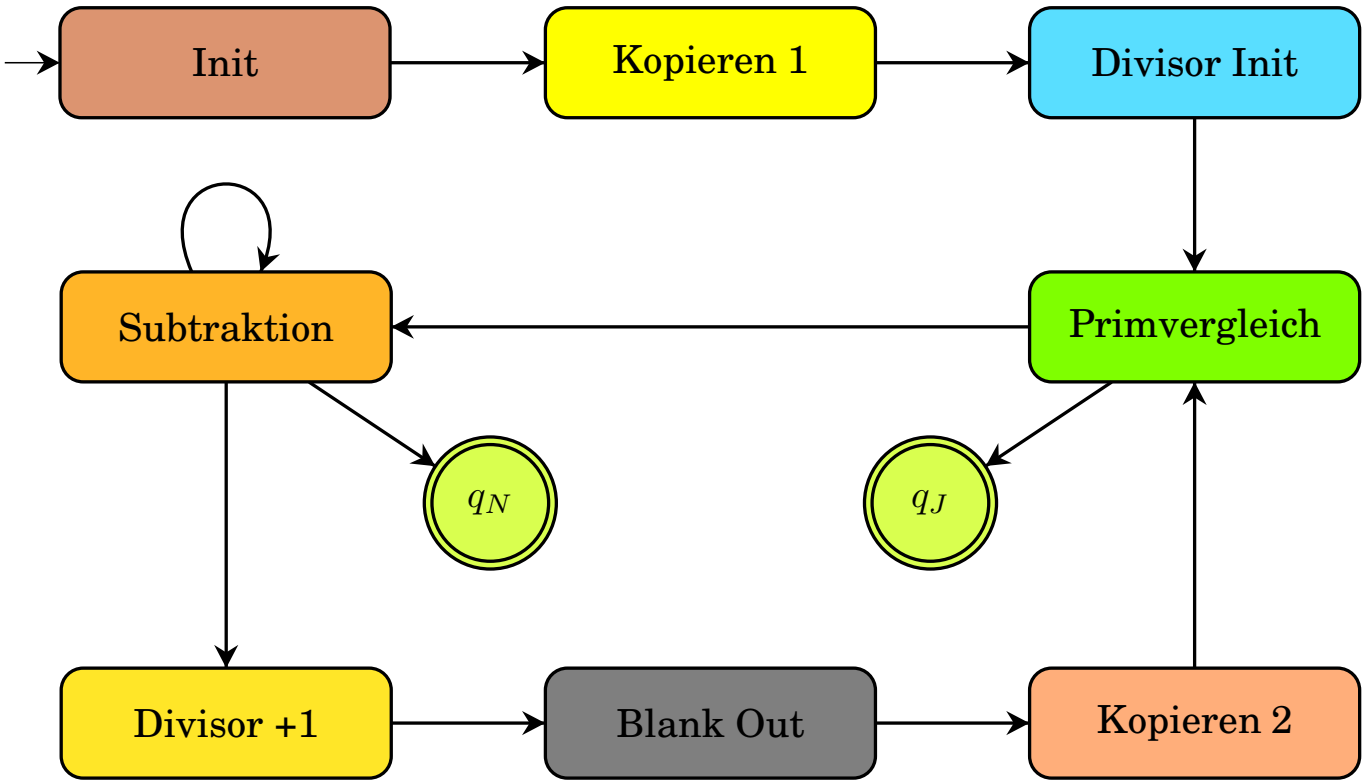
```
w = leseZahl(); divisor = 2;

while (divisor < w) {
  k = w;
  while(k > 0)
    k = k - divisor;
  if (k == 0)
    halte(w ist nicht prim);
  divisor = divisor + 1;
}

halte(w ist prim);
```

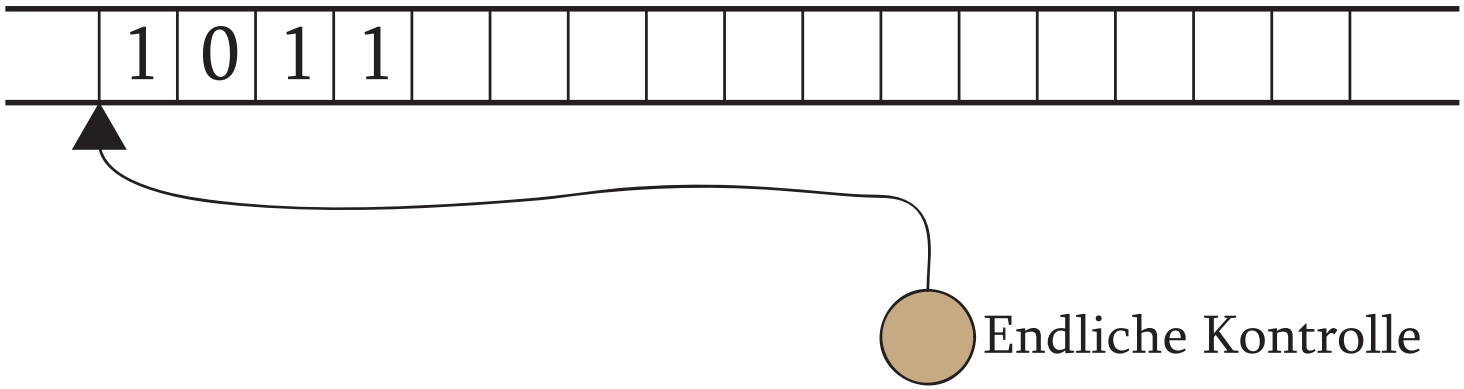
Aufwand leider **exponentiell**.

Zusammenhang der Unterprogramme



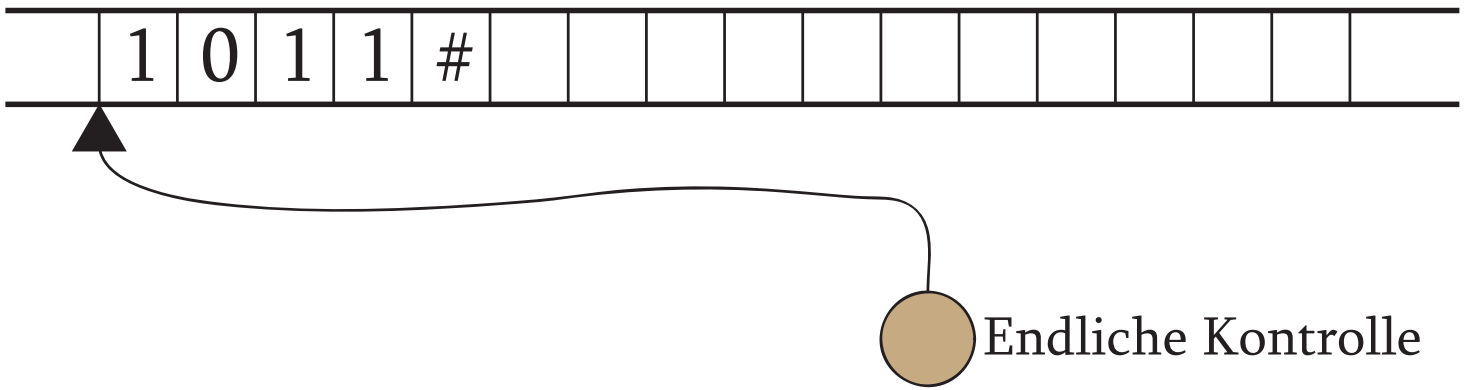
Ausganssituation

Funktionsweise am Beispiel von $1011_2 = 11_{10}$



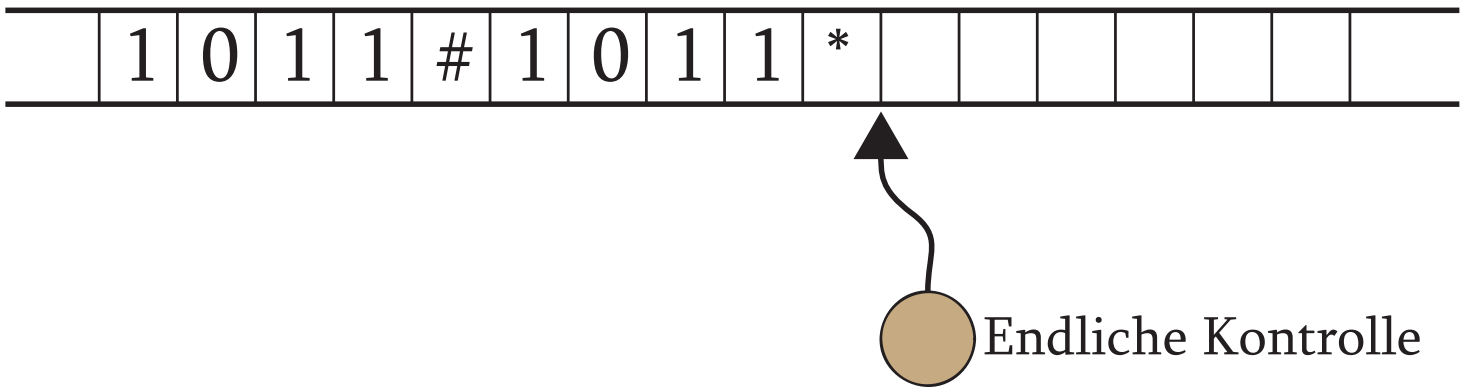
Unterprogramm „Init“

Schreibe # hinter das Eingabewort.



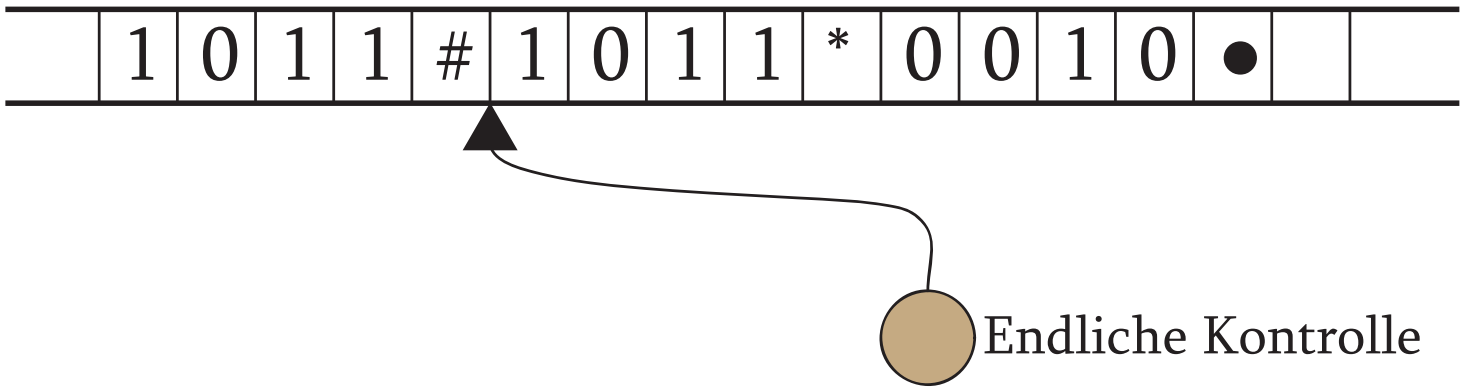
Unterprogramm „Kopieren 1“

Kopiere das Eingabewort w und setzte ein $*$ an das Ende. Die Kopie sei mit k bezeichnet.



Unterprogramm „Divisor Init“

Setze die Zahl $d = 10_2$ an das Ende und fülle mit so viel Nullen auf, dass $|d| = |w|$.



Unterprogramm „Primvergleich“

Vergleiche d mit k und führe folgende Aktion aus:

Unterprogramm „Primvergleich“

Vergleiche d mit k und führe folgende Aktion aus:

- $d = k$? Dann gehe in Zustand q_J und stoppe.

Unterprogramm „Primvergleich“

Vergleiche d mit k und führe folgende Aktion aus:

- $d = k$? Dann gehe in Zustand q_J und stoppe.
- $d \neq k$? **Führe nächstes Unterprogramm aus.**

Unterprogramm „Subtraktion“

Ziehe Divisor d von k ab. Folgende Fälle können nach der Subtraktion eintreten:

Unterprogramm „Subtraktion“

Ziehe Divisor d von k ab. Folgende Fälle können nach der Subtraktion eintreten:

- $k < 0$. Die „Division“ ist beendet. TM geht zum nächsten Unterprogramm.

Unterprogramm „Subtraktion“

Ziehe Divisor d von k ab. Folgende Fälle können nach der Subtraktion eintreten:

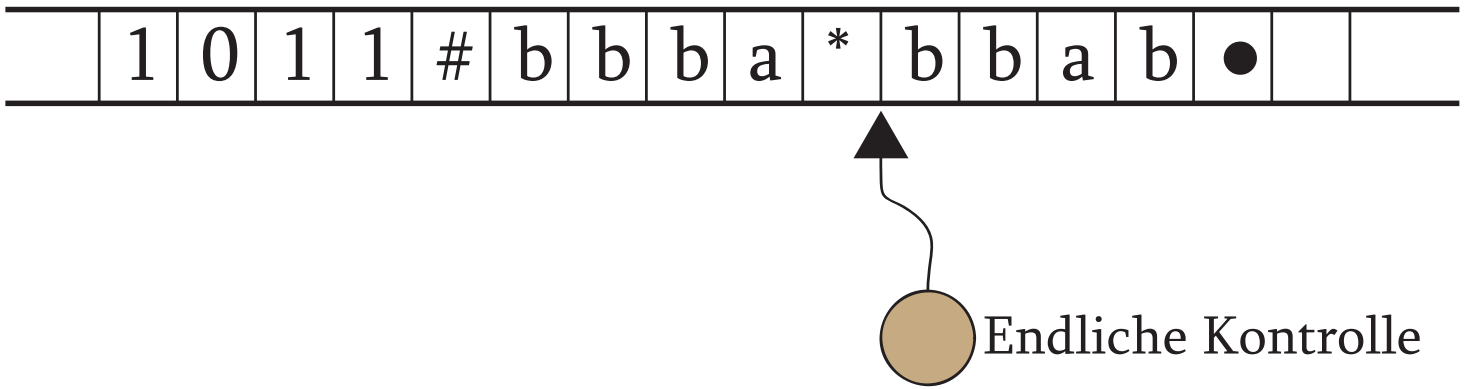
- $k < 0$. Die „Division“ ist beendet. TM geht zum nächsten Unterprogramm.
- $k > 0$. Die Subtraktion wird wiederholt.

Unterprogramm „Subtraktion“

Ziehe Divisor d von k ab. Folgende Fälle können nach der Subtraktion eintreten:

- $k < 0$. Die „Division“ ist beendet. TM geht zum nächsten Unterprogramm.
- $k > 0$. Die Subtraktion wird wiederholt.
- $k = 0$. Die „Division“ ging auf, w ist nicht prim. Wechsle also in q_N und stoppe.

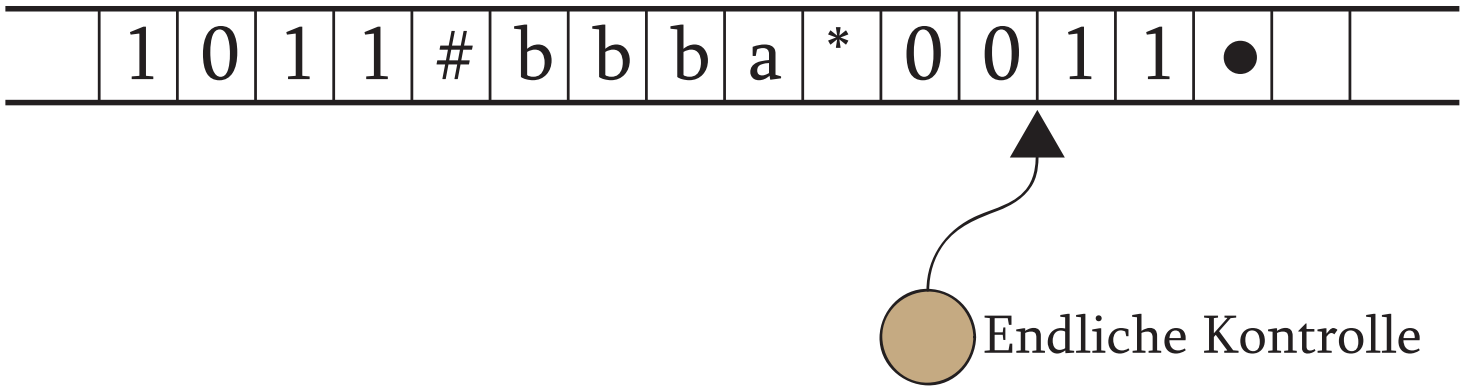
Fall $k < 0$



- $a, b \in \Gamma$, wobei $a \widehat{=} 1$ und $b \widehat{=} 0$.
- Die nächste Subtraktion würde einen Unterlauf produzieren!

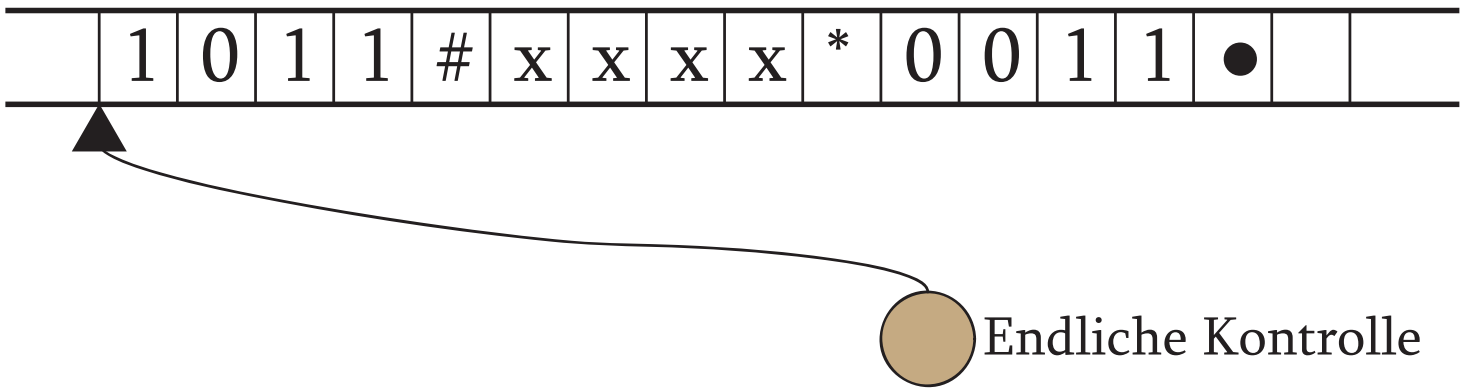
Unterprogramm „Divisor +1“

Der Divisior d wird binär um den Wert 1 inkrementiert.



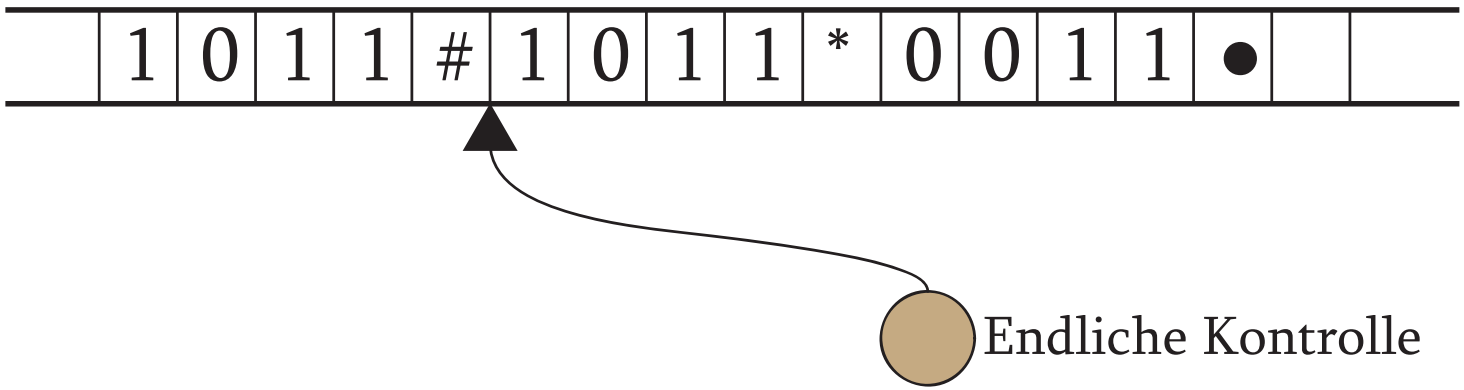
Unterprogramm „Blank Out“

Überschreibe k durch ein Hilfssymbol aus Γ, z.B. X.



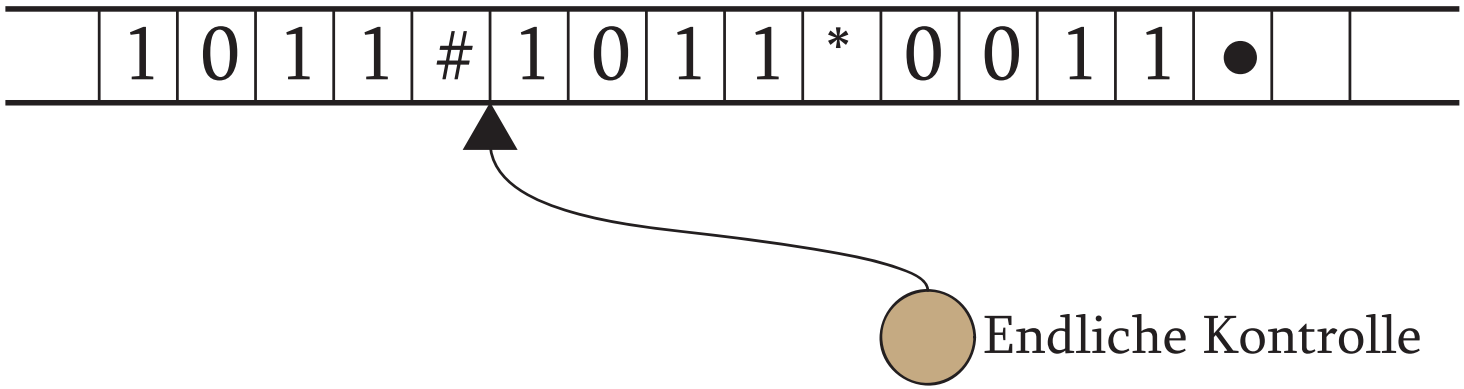
Unterprogramm „Kopieren 2“

Kopiere Eingabewort w an die Stelle der Xs.



Unterprogramm „Kopieren 2“

Kopiere Eingabewort w an die Stelle der Xs.



Fahre fort mit Unterprogramm „Primvergleich“.

Formale Definition der TM

$\mathcal{M} := (Q, \Sigma, \Gamma, \sqcup, \delta, s, F)$ wobei:

Formale Definition der TM

$\mathcal{M} := (Q, \Sigma, \Gamma, \sqcup, \delta, s, F)$ wobei:

- $\Sigma := \{0, 1\}$

Formale Definition der TM

$\mathcal{M} := (Q, \Sigma, \Gamma, \sqcup, \delta, s, F)$ wobei:

- $\Sigma := \{0, 1\}$
- $\Gamma := \{X, a, b, \#, *, \bullet\}$

Formale Definition der TM

$\mathcal{M} := (Q, \Sigma, \Gamma, \sqcup, \delta, s, F)$ wobei:

- $\Sigma := \{0, 1\}$
- $\Gamma := \{X, a, b, \#, *, \bullet\}$
- $F := \{q_J, q_N\}$

Formale Definition der TM

$\mathcal{M} := (Q, \Sigma, \Gamma, \sqcup, \delta, s, F)$ wobei:

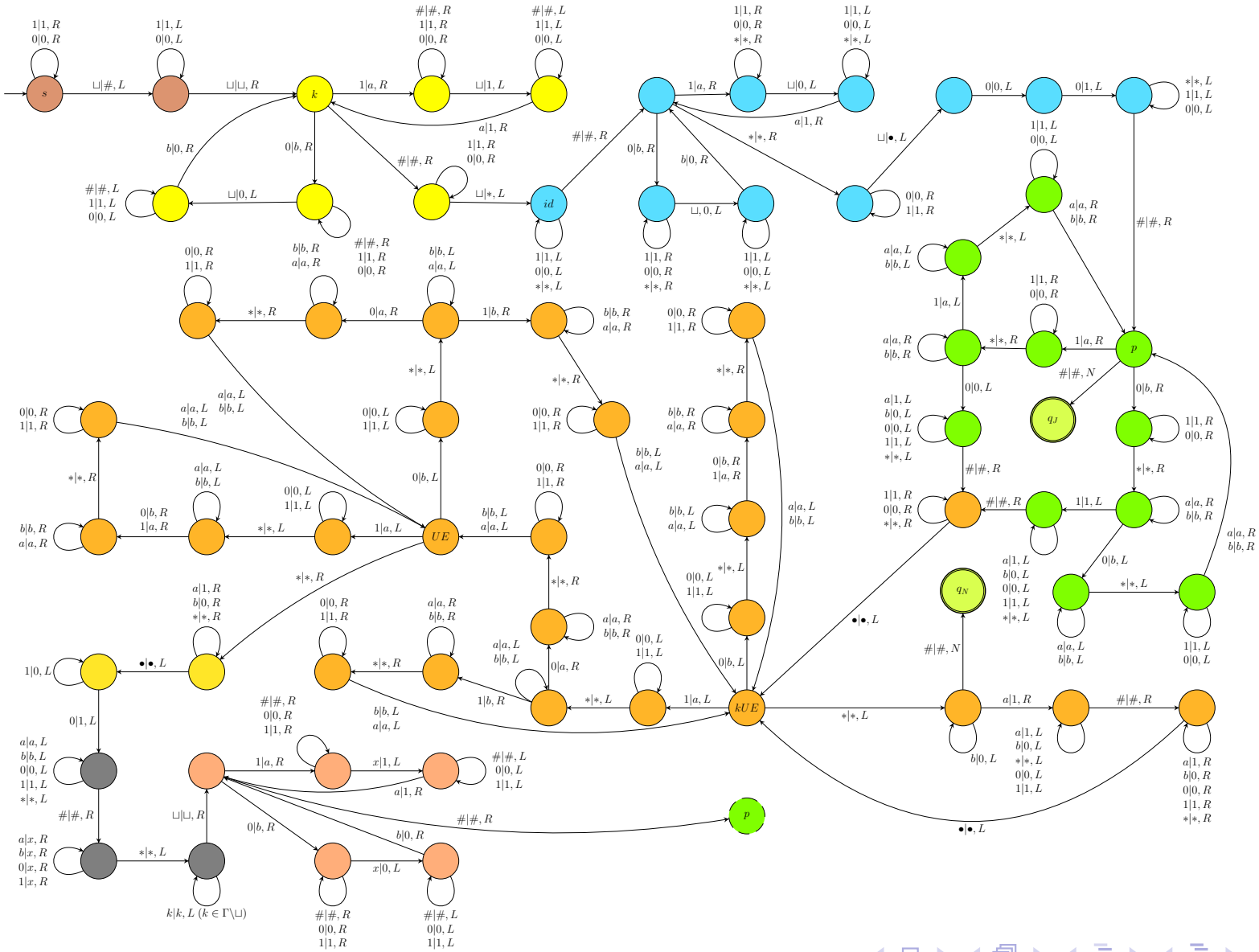
- $\Sigma := \{0, 1\}$
- $\Gamma := \{X, a, b, \#, *, \bullet\}$
- $F := \{q_J, q_N\}$
- $|Q| = 68$

Formale Definition der TM

$\mathcal{M} := (Q, \Sigma, \Gamma, \sqcup, \delta, s, F)$ wobei:

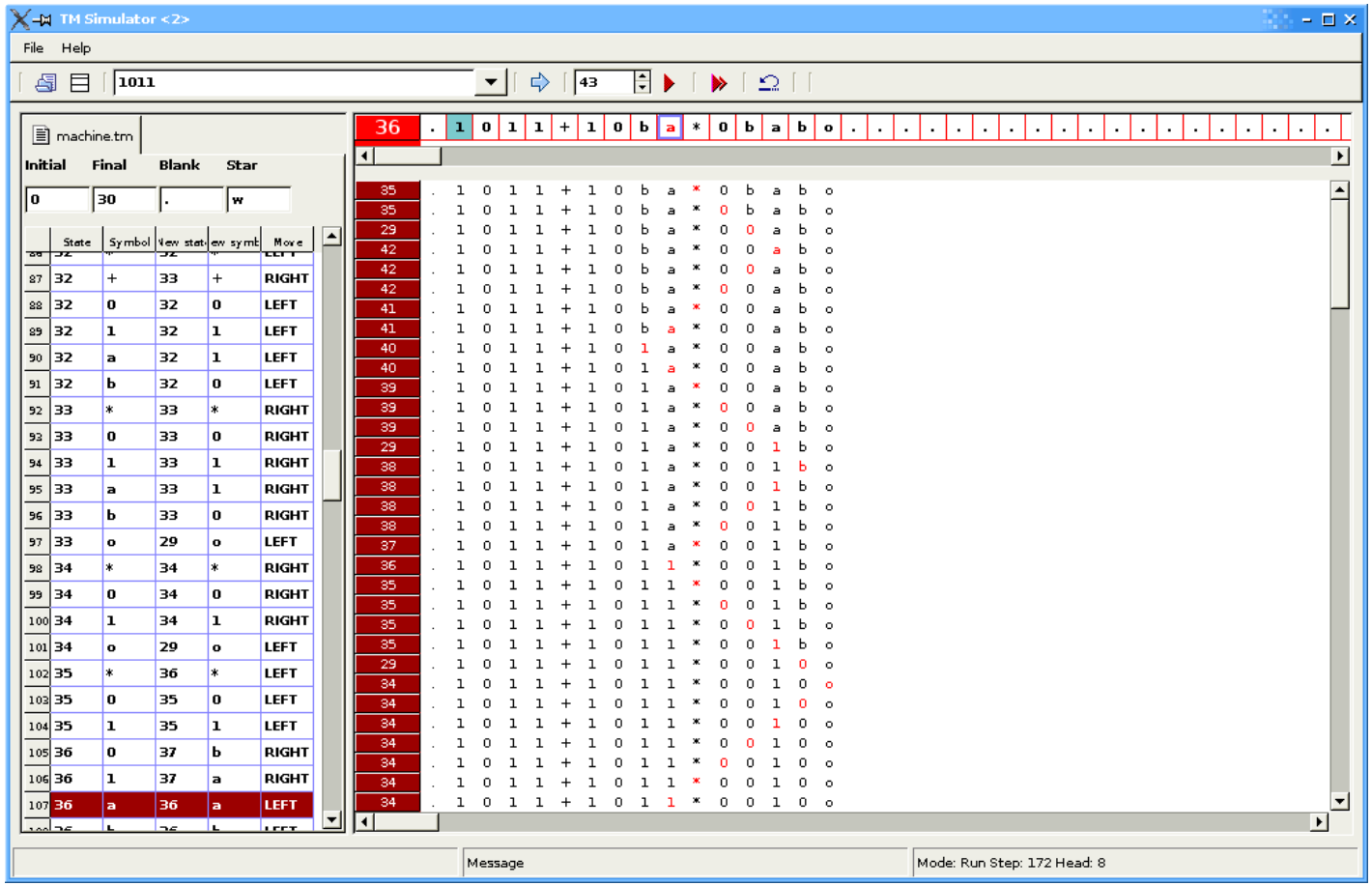
- $\Sigma := \{0, 1\}$
- $\Gamma := \{X, a, b, \#, *, \bullet\}$
- $F := \{q_J, q_N\}$
- $|Q| = 68$
- $|\delta| = 243$

Übergangsgraph



Simulation

TM Simulator: http://www.ncc.up.pt/fado/node2.html



Ergebnisse

Simulation mit verschiedenen Eingaben w:

- $w = 1011_2 = 11_{10} \rightarrow$ **2586** Abarbeitungsschritte.

Ergebnisse

Simulation mit verschiedenen Eingaben w:

- $w = 1011_2 = 11_{10} \rightarrow$ **2586** Abarbeitungsschritte.
- $w = 1100001_2 = 97_{10} \rightarrow$ bereits **87274** Schritte.

Ergebnisse

Simulation mit verschiedenen Eingaben w:

- $w = 1011_2 = 11_{10} \rightarrow$ **2586** Abarbeitungsschritte.
- $w = 1100001_2 = 97_{10} \rightarrow$ bereits **87274** Schritte.
- $w = 1111100101_2 = 997_{10} \rightarrow$ Überlauf des Simulators...

Ergebnisse

Simulation mit verschiedenen Eingaben w:

- $w = 1011_2 = 11_{10} \rightarrow$ **2586** Abarbeitungsschritte.
- $w = 1100001_2 = 97_{10} \rightarrow$ bereits **87274** Schritte.
- $w = 1111100101_2 = 997_{10} \rightarrow$ Überlauf des Simulators...

PS: Die Gödelnummer hat **21461** Stellen!

Ende...

Folien online unter: www.logn.de