Heutige Doppelstunde

Erste Stunde:

- Abschluss Kryptografie. Warum RSA funktioniert
- Endliche Automaten - Theorie

Kapitel 10.6 in Tigerjython: *Effizienz & Grenzen/Endliche Automaten* *(http://www.tigerjython.ch)*

Zweite Stunde

2 Übungsteile

AUFGABEN

1. Versuch den Geheimtext
   AV SFX EXSWAXSXAFDTWMFZSZXJFXSTF
   CTFFSTUXJSXJKLSMES TDUFXMFSCGEEXF
   YJXMXSEAV
   ETP
   zu entschlüsseln. Es handelt sich um eine Caesar-Verschlüsselung.

   Bemerkung: Eine Lösungsmöglichkeit besteht darin, davon auszugehen, dass der Buchstabe E in deutschen Texten weitaus am häufigsten vorkommt. Du kannst aber auch alle Verschiebungsmöglichkeiten durchprobieren.

2. Orientiere dich auf dem Internet, was man unter dem Verschlüsselungsverfahren mit einer Skytale versteht und implementiere einen Encoder/Decoder nach diesem Prinzip.

3. Begründe, warum die Caesar-Verschlüsselung ein Spezialfall des Vigenère-Verfahrens ist.

4. Erzeuge mit zwei Primzahlen p und q (beide kleiner als 100) einen öffentlichen und privaten Schlüssel gemäss dem RSA-Verfahren und verschlüssle/entschlüssle damit einen Text.

Zusatzaufgaben zu 10.5

Aufgabe 5: Erstelle ein Programm, das die Häufigkeiten der Buchstaben A-Z in einem Text (Textdatei) zählt und als Liste ausgibt. Es kann angenommen werden, dass der Text nur Grossbuchstaben und das Leerzeichen enthält.

Aufgabe 6: Erstelle ein Programm, welches aus den Primzahlen p und q und dem öffentlichen Schlüssel [m,e] den privaten Schlüssel [m, d] berechnet.
Hinweis: Lemma von Bézout.

Satz von Euler: $a^{\phi(m)} = 1$ (modulo m) wenn ggT(a,m) = 1

Allgemeiner gilt: $a^b = a^{b \bmod \phi(m)}$ (modulo m)

Daraus folgt sofort: $a * a^{\phi(m) - 1} = 1$ (modulo m)

Beispiel: wähle m = 10 = 2 * 5. Es folgt $\phi(10) = 4$. Für die Zahl a=7 gilt ggT(7,10) = 1.
Der Satz von Euler ist also anwendbar und ergibt: $7^{\phi(10)} = 7^4 = 1$ (modulo 10).

Eine direkte Rechung bestätigt dies: $7^4 = 49 * 49 = 2401 = 1$ (modulo 10).

Beispiel: Man berechne 7^{222} mod 10 mit dem Satz von Euler.

Satz von Euler: $a^{\phi(m)} = 1$ (modulo m) wenn ggT(a,m) = 1

Allgemeiner gilt: $a^b = a^{b \bmod \phi(m)}$ (modulo m)

Daraus folgt sofort: $a * a^{\phi(m) - 1} = 1$ (modulo m)

Beispiel: wähle m = 10 = 2 * 5. Es folgt $\phi(10) = 4$. Für die Zahl a=7 gilt ggT(7,10) = 1.
Der Satz von Euler ist also anwendbar und ergibt: $7^{\phi(10)} = 7^4 = 1$ (modulo 10).

Eine direkte Rechung bestätigt dies: $7^4 = 49 * 49 = 2401 = 1$ (modulo 10).

Beispiel: Man berechne 7^{222} mod 10 mit dem Satz von Euler.
7^{222} mod 10 = $(7^4) * (7^4) * (7^4)$ …..* 7^2 = 1 * 1* …. * 7^2 = 49 mod 10 = 9.

Ende Zahlentheorie

Puu, geschafft

Endliche Automaten

EINFÜHRUNG

Will man untersuchen, welche Probleme ein Computer grundsätzlich lösen kann und welches seine Grenzen sind, so muss man zuerst exakt definieren, was man unter einer Rechenmaschine versteht. Der berühmte Mathematiker und Informatiker Alan Turing veröffentlichte bereits 1936 eine Untersuchung zu diesem Thema, lange bevor es überhaupt einen programmierbaren Digitalrechner gab. Die nach ihm benannte **Turingmaschine** durchläuft programmgesteuert und auf Grund von Eingabewerten, die sie von einem Band liest, schrittweise einzelne Zustände und schreibt dabei Ausgabewerte auf das Band. Diese grundsätzliche Vorstellung über die Funktionsweise des Computers ist auch heute noch gültig, denn jeder Prozessor ist eigentlich eine Turingmaschine, die im Takt einer Clock Zustand um Zustand durchläuft. Besser an die Praxis angepasst sind allerdings Zustandsautomaten, die sich mit Zustandsgraphen modellieren lassen. Man nennt sie **Endliche Automaten**.

PROGRAMMIERKONZEPTE: *Turingmaschine, Endlicher Automat, Mealy-Automat, Automatengraph, Formale Sprache*

Kapitel 10.6 in Tigerjython: *Effizienz & Grenzen/Endliche Automaten* *(http://www.tigerjython.ch)*

Turing Maschine

- https://www.youtube.com/watch?v=E3keLeMwfHY

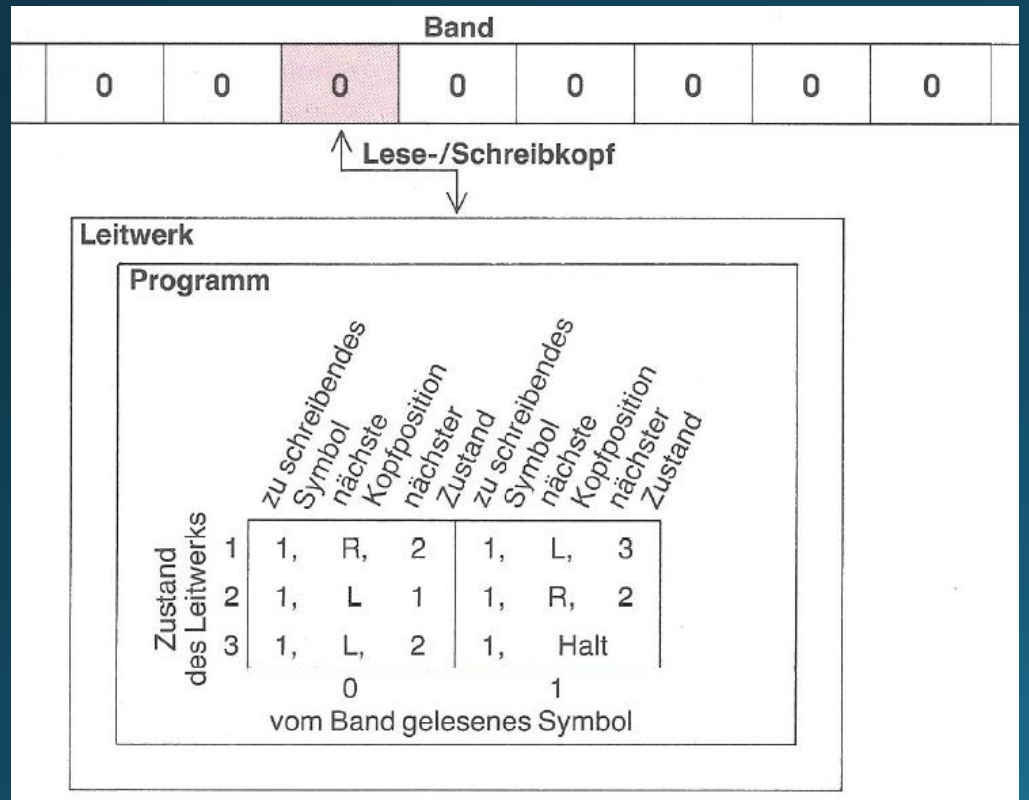
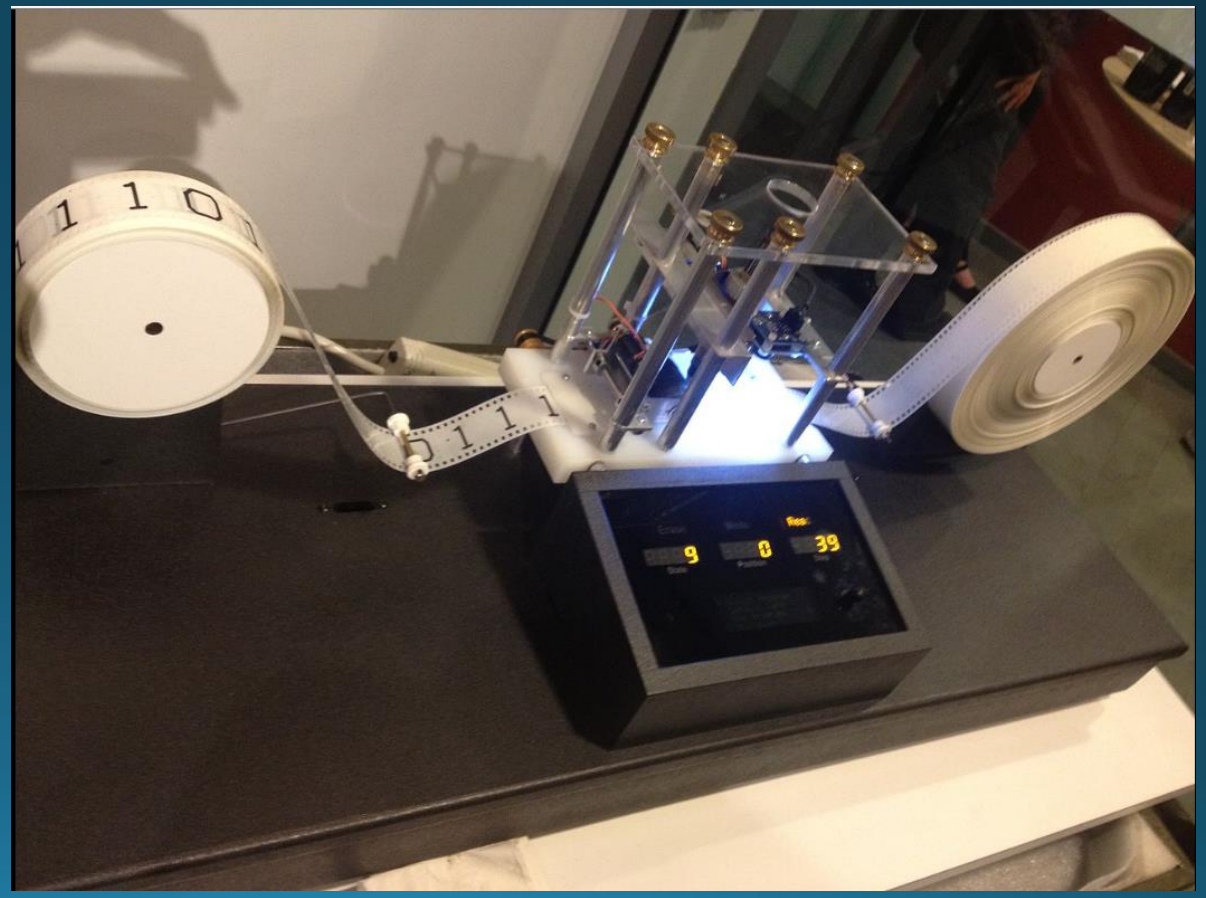


Bild 1: Eine Turingmaschine und ihr Programm.



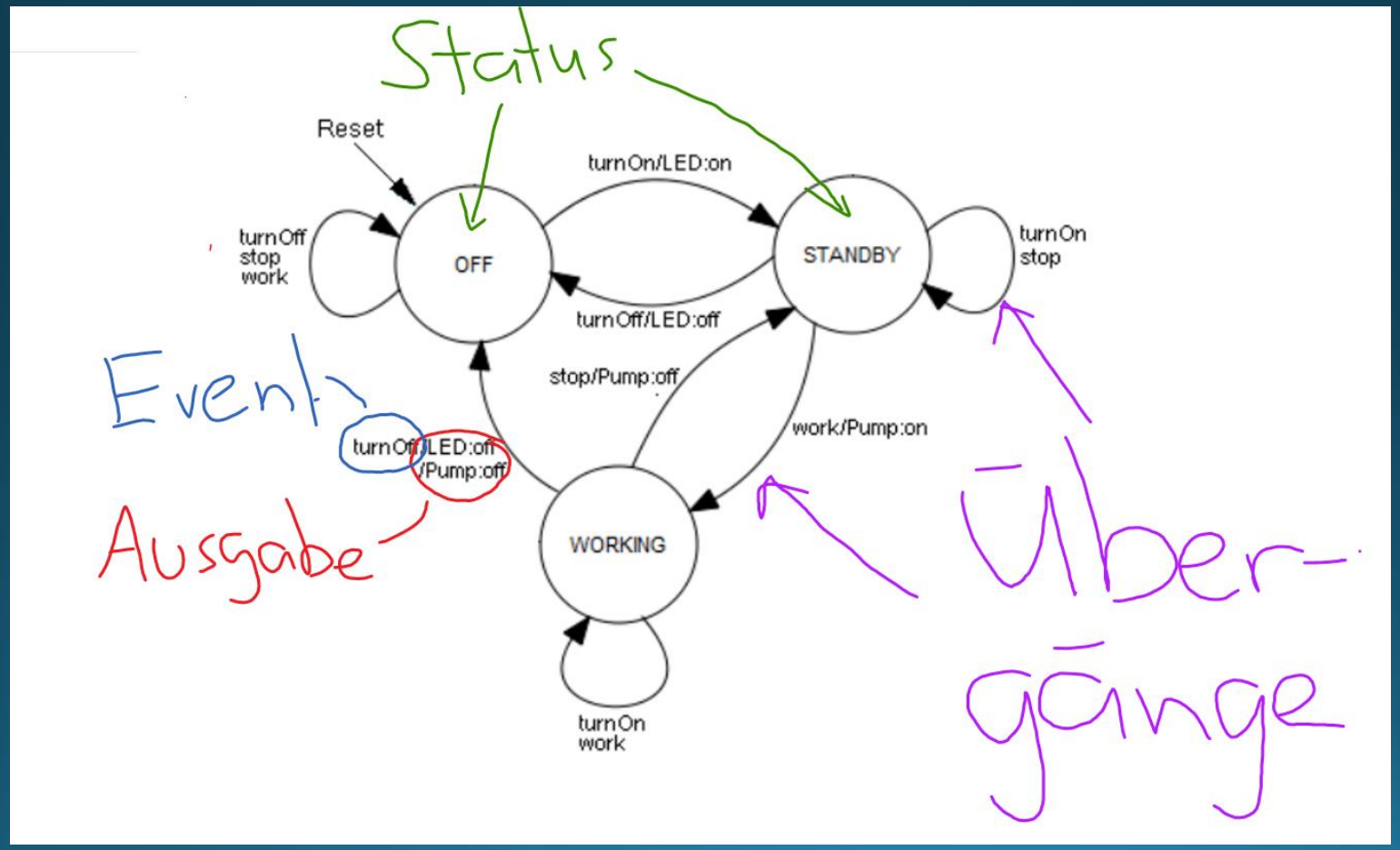
Alan Turing

https://de.wikipedia.org/wiki/Alan_Turing
https://www.youtube.com/watch?v=xj2LTM-Dr8g

Endlicher Automat

Turing Maschine mit endlich langem Band

meistens Beschreibung mit Automatengraph statt Tabelle



Übergangstabelle

Aufgabe: Finde den Fehler

Du kannst das Verhalten auch in einer Tabelle festhalten, in der du zu jedem Zustand s und jeder Eingabe t den Nachfolgezustand s' angibst. Mit einem Stern bezeichnest du den Anfangszustand.

Übergangstabelle:

t = **s =**	**OFF(*)**	**STANDBY**	**WORKING**
turnOff	OFF	OFF	OFF
turnOn	STANDBY	STANDBY	WORKING
stop	OFF	STANDBY	STANDBY
work	OFF	STANDBY	WORKING

Mathematisch ausgedrückt kannst du sagen, dass der Nachfolgezustand s' eine Funktion des aktuellen Zustands *s* und der Eingabe t ist: $s' = F(s, t)$. Du nennst F die **Übergangsfunktion**.

Ausgabetabelle

Ausgabe = Anzeige, Motor, Steuerung, Schalter…

Die Ausgaben, die zu jedem Zustand und einer Eingabe gehören, kannst du ebenfalls tabellarisch festhalten:

Ausgabetabelle:

t = s =	**OFF(*)**	**STANDBY**	**WORKING**
turnOff	-	LED off	LED off, Pump off
turnOn	LED on	-	-
stop	-	-	Pump off
work	-	Pump on	-

Mathematisch kannst du auch hier sagen, dass die Ausgabe g eine Funktion des aktuellen Zustands s und der Eingabe ist: $g = G(s, t)$. Du nennst G die **Ausgabefunktion**.

Übung Espresso Maschine 15 Min.

a) Spielen Sie das Programm durch (Variante enumeration)

b) Starten Sie auch die Maus-Variante und vergleichen Sie die Logik

c) Ändern sie es eines so ab, dass sie während dem Kaffee-Zubereiten NICHT ausgeschaltet werden kann

Zusatzaufgaben zu 10.6

Aufgabe 2: Verifiziere, dass das Programm «Espresso Maschine mit enumeration» korrekt gemäss Automatengraphen funktioniert. Führe dazu Tastendrücke aus, sodass jeder Übergang (jede Kante im Automatengraphen) mindestens einmal durchlaufen wurde.

Zusatzfrage: wieviel Tastendrücke wurden dazu gebraucht?

Aufgabe Parkscheinautomat

AUFGABEN

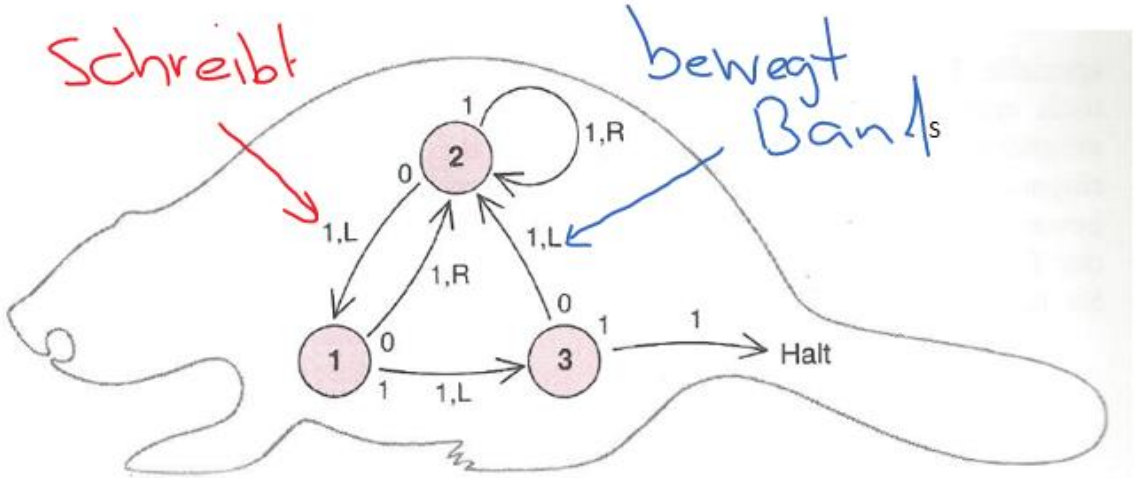
1. Ein Parkscheinautomat akzeptiert nur 1 € und 2 € Geldstücke, die einzeln hintereinander eingeworfen werden. Sobald der Automat mindestens die Parkgebühr erhalten hat, gibt er den Parkschein und das Restgeld aus. Die Parkgebühr betrage 3 €.

   Der Automat durchlaufe ausgehend vom Startzustand S0 die Zustände S1 oder S2, je nachdem ob 1 € oder 2 € eingeworfen werden. Seine Ausgabewerte sind - (nichts), K (Karte) oder K,R (Karte und Rückgeld).

   a. Erstelle die Übergangs- und Ausgabetabellen
   b. Zeichne den Automatengraphen
   c. Erstelle ein Programm mit GConsole, welches das Drücken der Zahlentaste 1 als Einwurf von 1 € und das Drücken der Zahlentaste 2 als Einwurf von 2 € interpretiert und den Folgezustand, sowie die Ausgabewerte in das Konsolenfenster ausschreibt.

Hausaufgabe «Biber»

Aufgabe 2: Erstelle Sie ein Programm, das einen endlichen Automaten gemäss folgendem Automatengraphen darstellt. Es handelt sich um ein Programm, das auf einer Turing Maschine läuft.
Tipp: orientieren Sie sich an der Espresso Maschine mit enumeration. Es gibt keine Tasteneingaben. Verwenden Sie eine Liste als «Band».



Fleissiger Biber (Busy Beaver):

https://de.wikipedia.org/wiki/Flei%C3%9Figer_Biber

Zustände n	Turingmaschinen	$\Sigma(n)$	Quelle
1	64	1	(1962; Rado)
2	20736	4	(1962; Rado)
3	16777216	6	(1965; Lin, Rado)
4	$2{,}56\times10^{10}$	13	(1972; Weimann, Casper, Fenzel)
5	$\approx 6{,}34\times10^{13}$	≥ 240	(1983; Jochen Ludewig)
		≥ 501	(1983; Uwe Schult)
		≥ 1915	(1984; George Uhing)
		≥ 4098	(1989; Jürgen Buntrock und Heiner Marxen)
6	$\approx 2{,}32\times10^{17}$	$> 3{,}514\times10^{18267}$	(2010; Pavel Kropitz)
7	$\approx 1{,}18\times10^{21}$	Abschätzung unrealistisch	

Anhang

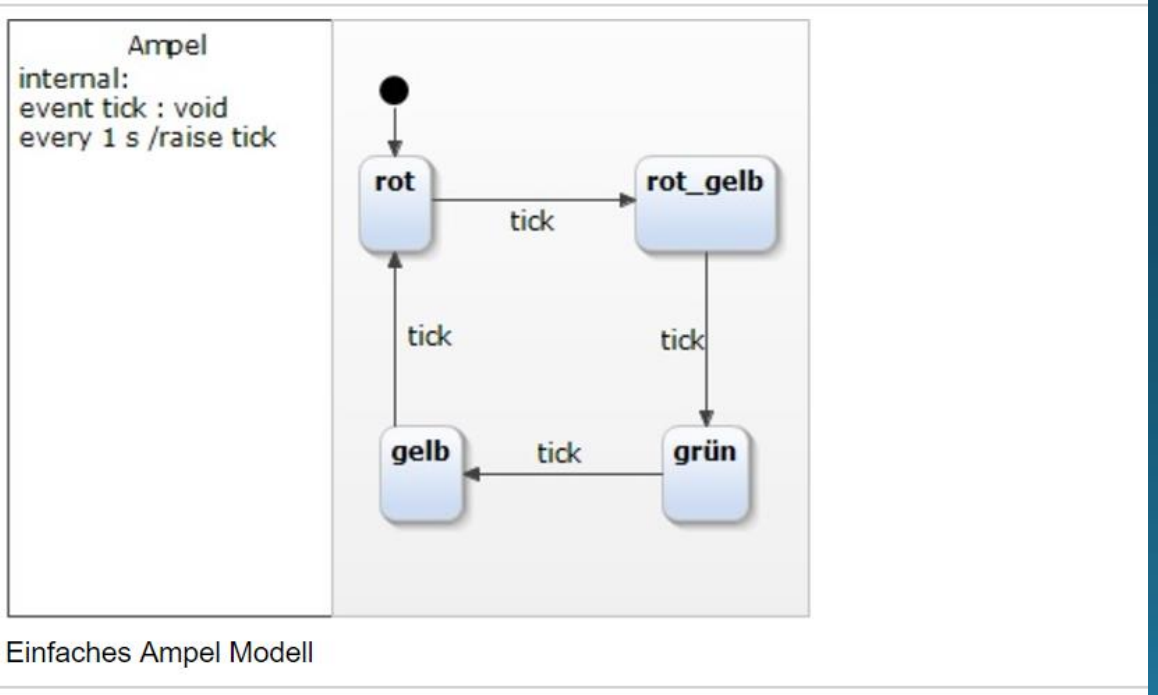
- Fleissiger Biber (Busy Beaver):
- https://de.wikipedia.org/wiki/Flei%C3%9Figer_Biber

- Sehr schöne Erklärung und Beispiele, Programmcode in C
- https://www.mikrocontroller.net/articles/Statemachine
- Beispiel Software:
- https://www.itemis.com/en/yakindu/state-machine/
- https://state-machine.com (Profi Hardware Programmierer)
- http://smc.sourceforge.net/ State machine Compiler, auch Python

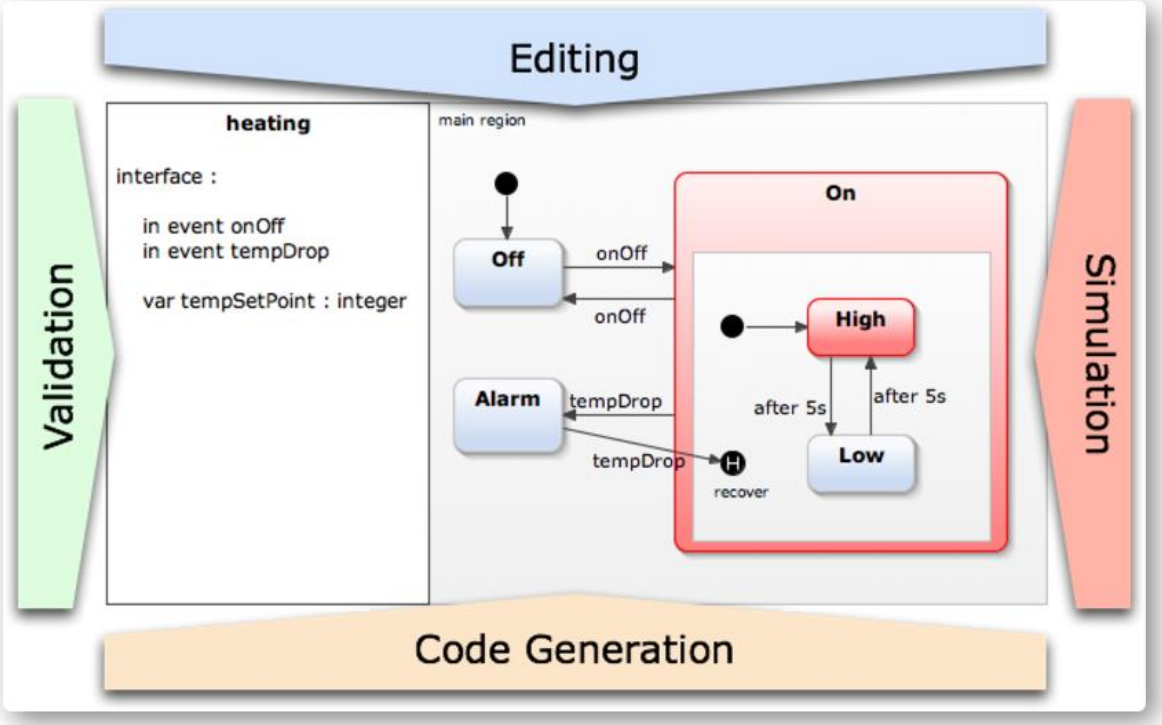
Yakindu

Einfaches Ampel Modell

Das Modell einer einfachen Ampel wie oben beschrieben sieht in Yakindu SCT wie folgt aus:



Einfaches Ampel Modell



Features of YAKINDU Statechart Tools