Unbeschränkte Registermaschine (URM):

- Hardware: Unendlich viele Register (Variablen) $R_0, R_1, \ldots,$
  jede kann eine (beliebig große) natürliche Zahl speichern.
- Programmiersprache: Basic-artig, nummerierte Programm-Zeilen, jede enthält eines von:

addiere 1 zu R_i	$R_i := R_i + 1,$
subtrahiere 1 (wenn möglich, sonst 0)	$R_i := R_i - 1,$
eine Sprung-Anweisung	`goto` m,
teste auf = 0	`if` $R_i = 0$ `goto` m,
liefere R_0 als Output	`return`

Bevor wir URM-Programme formal definieren, ein einfaches Beispiel:

Beispiel 0.1. Addition zweier Zahlen:

```
0  if R1 = 0 goto 4
1  R1 := R1 - 1
2  R0 := R0 + 1
3  goto 0
4  return R0
```

Berechnung bei Input $(3, 2)$

Zeit:	0	Zeile:	0	R_0 :	3	R_1 :	2	Input $(3, 2)$ ist in R_0 und R_1 gespeichert.
Zeit:	1	Zeile:	1	R_0 :	3	R_1 :	2	
Zeit:	2	Zeile:	2	R_0 :	3	R_1 :	1	
Zeit:	3	Zeile:	3	R_0 :	4	R_1 :	1	
Zeit:	4	Zeile:	0	R_0 :	4	R_1 :	1	
Zeit:	5	Zeile:	1	R_0 :	4	R_1 :	1	
Zeit:	6	Zeile:	2	R_0 :	4	R_1 :	0	
Zeit:	7	Zeile:	3	R_0 :	5	R_1 :	0	
Zeit:	8	Zeile:	0	R_0 :	5	R_1 :	0	
Zeit:	9	Zeile:	4	R_0 :	5	R_1 :	0	
Zeit:	S	Zeile:	S	R_0 :	5	R_1 :	0	Output 5 aus R_0.

Daher: Addition ist berechenbar (nach unserer Definition).

Formal könnten wir das so definieren (das trägt hier nicht zum Verständnis bei, deutet aber an wie wir später ein universelles Programm schreiben könnten):

Definition 0.2. Ein *URM-Programm* ist eine endliche Teilmenge P von $\mathbb{N}^4$ die folgendes erfüllt:

- Wenn $l_1 = (a, b, c, d) \in P$ und $l_2 = (a', b', c', d') \in P$, und $a = a'$, dann ist $(a, b, c, d) = (a', b', c', d')$. (Die erste Komponente ist die eingeutige Zeilennummer.)
- Wenn $l = (a, b, c, d) \in P$ dann ist $b \in \{0, 1, 2, 3, 4\}$. (Wir interpretieren 0 als `return` , 1 als Addition, 2 als Subtraktion, 3 als `goto` und und 4 als Testfunktion.)
- Es gibt ein $l_0 = (0, b, c, d) \in P$. (Startzeile)

In dieser formalen Schreibweise hat unser Beispielprogramm also z.B. Form
$P = \{(0, 4, 1, 4), (1, 2, 1, 0), (2, 1, 0, 0), (3, 3, 0, 0), (4, 0, 0, 0)\}$.

Definition 0.3. Sei P ein URM-Programm. B ist eine k-Berechnungsfolge von P auf Input $(i^0, \ldots, i^n)$, wenn es ein $K > n$ gibt so daß folgendes gilt:

- Wenn $(a, b, c, d) \in P$ dann $d < K$. (D.h. K ist größer als alle Nummern der im Programm P verwendeten Register.)
- $B = (B_0, \ldots, B_k)$ ist eine Folge der Länge $k + 1$.

- $B_i \in \mathbb{N}^{K+1}$, d.h. jedes B_i ist ein $K+1$-Tupel $(r_i^0, \ldots, r_i^K, a_i)$ natürlicher Zahlen. (Das letzte Element ist die aktive Zeilennumer.)
- $B_0 = (i^0, \ldots, i^n, 0)$.
- Angenommen $B_i = (r^0, \ldots, r^K, a)$. Dann gibt es (genau) ein $(a, b, c, d) \in P$.

  Wenn $b = 0$, dann terminiert das Programm, und r^0 ist der Output.

  Ansonsten definieren wir $B_{i+1} = (s^0, \ldots, s^K, a')$ folgendermaßen:
  - Wenn $b = 3$, dann ist $(s^0, \ldots, s^K) := (r^0, \ldots, r^K)$ und $a' = c$.
  - Wenn $b = 4$, dann ist $(s^0, \ldots, s^K) := (r^0, \ldots, r^K)$, und wenn $r^d = 0$, dann ist $a' = c$, ansonsten $a + 1$ (wenn eine solche Zeilennummer existiert, sonst 0).
  - Wenn $b = 1$ (oder 2), dann ist $s^d = r^d + 1$ (oder $\max(0, r^d - 1)$), für die $i < K$ ungleich d gilt $s^i = r^i$. $a' = a + 1$ (wenn eine solche Zeilennummer existiert, sonst 0).