ETH ZURICH
LEHRDIPLOM INFORMATIK

Algorithmisch lösbare und algorithmisch unlösbare Probleme

Studentin: *Alexandra Maximova*

Fach: *Fachdidaktik 2 (Berechenbarkeit)* – Professor: *Giovanni Serafini, Juraj Hromkovič*
Abgabedatum: *22.04.2020*

Aufgabe zur Reduktion im 'positiven' Sinne

Nehmen wir an, wir haben einen Algorithmus B, um den Logarithmus einer beliebigen Zahl $x \neq 0$ zur Basis b zu berechnen, d.h. $log_b(x)$. Entwickle mittels Reduktion einen Algorithmus zur Berechnung vom Logarithmus zur Basis a.
Heute scheint uns dieses Problem künstlich und konstruiert zu sein, aber vor nicht all zu langer Zeit war das tatsächlich praxisrelevant. Bevor die modernen Taschenrechner erschienen, die Logarithmen in jeder Basis berechnen können, haben Ingenieure täglich mit Tabellen gearbeitet, die die Logarithmen zur Basis 10 für sehr viele Zahlen aufgelistet haben. Manchmal brauchten sie den Logarithmus zu einer anderen Basis, die nicht tabelliert war; Das konnten sie aber einfach aus den Werten aus der Tabelle berechnen.
Zur Erinnerung, der Logarithmus ist bijektiv für $x \neq 0$ und es gelten folgende Rechenregeln:
(a) $log(x \cdot y) = log(x) + log(y)$

(b) $log(x^y) = y \cdot log(x)$

Musterlösung. Bezeichnen wir durch U_B das Problem, den Logarithmus zur Basis b zu berechnen und durch U_A das Problem, den Logarithmus zur Basis a zu berechnen. Wir werden $U_A \leq_{Alg} U_B$ zeigen, d.h. U_A auf U_B reduzieren. Dafür müssen wir und einen Algorithmus A konstruieren, welcher den gegebenen Algorithmus B benutzt und den Logarithmus zur Basis a berechnet.

Wir möchten also $log_a(x)$ berechnen und können dafür den Logarithmus zur Basis b von beliebigen Zahlen mit dem Algorithmus B berechnen. Sei $y = log_a(x)$. Dann, aus der Definition von Logarithmus, muss $a^y = x$ sein. Da nach Annahme $x \neq 0$, können wir auf beiden Seiten der Gleichung die bijektive Funktion log_b anwenden, ohne dass sich die Menge der Resultate ändert.

$$\begin{aligned} a^y &= x \\ log_b(a^y) &= log_b(x) \\ y \cdot log_b(a) &= log_b(x) && \text{(Rechenregeln anwenden)} \\ log_a(x) \cdot \log_b(a) &= log_b(x) && \text{(Definition für } y \text{ einsetzen)} \\ log_a(x) &= \frac{log_b(x)}{log_b(a)} && (a \neq 1) \end{aligned}$$

Im letzten Schritt haben wir beide Seiten der Gleichung durch $log_b(a)$ geteilt. Das können wir genau dann machen, wenn $a \neq 1$. Das können wir hier annehmen, weil es keinen Sinn hat, 1 als die Basis eines Logarithmus zu verwenden.

Den Algorithmus A konstruieren wir folgendermassen: Wir benutzen den Algorithmus B, um $log_b(x)$ und $log_b(a)$ zu berechnen, und dann ermitteln wir die Antwort, indem wir diese zwei Zahlen durcheinander teilen.

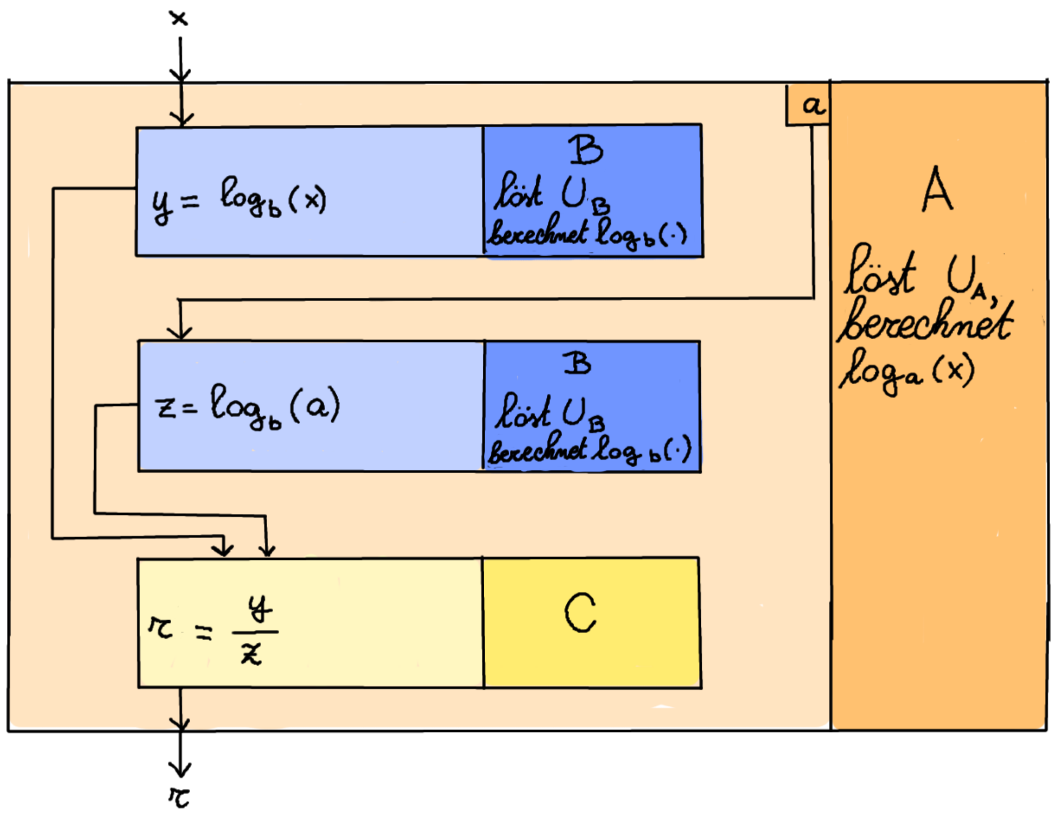

Abbildung 1: Reduktion von U_A auf U_B.

Aufgabe zur Reduktion im 'negativen' Sinne

How much wood would a woodchuck chuck if a woodchuck could chuck wood?

(a) Suppose “chuckïmplies throwing.

(b) Suppose “chuckïmplies vomiting.

Musterlösung.

(*a*) According to the Associated Press (1988), a New York Fish and Wildlife technician named Richard Thomas calculated the volume of dirt in a typical 25–30 foot (7.6–9.1 m) long woodchuck burrow and had determined that if the woodchuck had moved an equivalent volume of wood, it could move "about **700 pounds (320 kg)** on a good day, with the wind at his back".

(*b*) A woodchuck can ingest 361.92 cm^3 (22.09 cu in) of wood per day. Assuming immediate expulsion on ingestion with a 5% retainment rate, a woodchuck could chuck **343.82 cm^3** of wood per day.