Angewandte Softwareverifikation mit einem interaktiven Theorembeweiser
Version 28. März 2011

Das Lampen–Frosch–Problem

Eine unendliche Schlange von Fröschen steht vor einer unendlichen Kette von Lampen mit Ein-Aus-Tastschaltern. Sowohl Frösche als auch Lampen sind mit den positiven natürlichen Zahlen (1,2,3,...) durchnummeriert.

Zu Beginn der Unix-Epoche (0:00 Uhr am 1. Januar 1970 GMT) springt Frosch Nummer 1 auf jeden Schalter und schaltet damit alle Lampen an. Danach springt Frosch Nummer 2 auf jeden zweiten Schalter, so dass nur die Lampen mit ungeraden Nummern brennen. Frosch Nummer drei springt auf jeden dritten Schalter und schaltet dabei zum Beispiel Lampe 3 aus und Lampe 6 an. Alle weiteren Frösche folgen nach und nach, dabei springt Frosch Nummer n immer auf jeden n-ten Schalter.

Aufgabe

1. Welche Lampen brennen, nachdem alle Frösche gesprungen sind?
2. Formalisieren Sie das Lampen-Frosch-Problem in Pvs und beweisen Sie Ihre Lösung!

Detaillierte Hinweise

- Formalisieren Sie die Lampenreihe als Funktion [`nat` → `bool`]. Die Frösche werden dann Funktionen, die Lampenreihen auf Lampenreihen abbilden.
- Um den Endzustand der Lampenreihe zu bestimmen (nachdem alle Frösche gesprungen sind), ist strenggenommen eine Limesbildung (bezüglich einer entsprechenden Metrik) notwendig. Für dieses Praktikum ist es ausreichend, den finalen Lampenzustand wie folgt zu definieren: Der Zustand der Lampe n ist der, den sie hat, nachdem Frosch Nummer n gesprungen ist.