

Beispiel

Es gibt das bekannte Rätsel:

$$\text{SEND} + \text{MORE} = \text{MONEY}$$

Jeder Buchstabe muß durch eine Ziffer ersetzt werden, so daß die Rechnung stimmt.

Verschiedene Buchstaben müssen dabei verschiedene Ziffern erhalten.

Wie lösen wir solche Rätsel in Prolog?

Nehmen wir uns hier eine weniger bekannte Frage vor:

$$\text{FH} \cdot \text{RWTH} = \text{AACHEN}$$

Beispiel

Wir können das Rätsel lösen, indem wir verlangen, daß $[R, W, T, H, F, A, C, E, N, _]$ eine Permutation der Ziffern $0, \dots, 9$ ist *und* die Rechnung stimmt.

Dafür verwenden wir eine Relation $[num(N, L)]$, die aussagt, daß die Zahl N aus den Ziffern in der Liste L besteht.

$num(0, [\])$.

$num(N, [X|XS]) :- num(N1, XS), N \text{ is } X + 10 * N1$.

$loesung(FH, RWTH) :-$

$perm([R, W, T, H, F, A, C, E, N, _], [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),$

$num(FH, [H, F]),$

$num(RWTH, [H, T, W, R]),$

$num(AACHEN, [N, E, H, C, A, A]),$

$AACHEN \text{ is } FH * RWTH$.

Beispiel – Umdrehen einer Liste

Drehen wir eine Liste naiv um, ist die Laufzeit wieder quadratisch.

Die schnelle Art eine Liste umzudrehen, verwendet ähnlich wie in Haskell einen Akkumulator.

Wir definieren eine Regel, die aussagt, daß „ein Element hinübergeschoben wurde“.

```
hinueber([ ], L, L).
```

```
hinueber([E|XS], YS, L) :- hinueber(XS, [E|YS], L).
```

```
umgedreht(A, B) :- hinueber(A, [ ], B).
```

Was ergibt die Anfrage *umgedreht*(X, X)?