

1. Praktische Übung zur

Logische und funktionale Programmierung

GRUPPENÜBUNGEN:

(G 1) Gelöste Aufgabe

Man bestimme, ob eine natürliche Zahl eine Primzahl ist.

Beispiel:

```
?- is_prime(7).
```

Yes

LÖSUNG:

```
% is_prime(P) :- P ist prim
```

```
% (integer) (+)
```

```
is_prime(2).
```

```
is_prime(3).
```

```
is_prime(P) :- integer(P), P > 3, P mod 2 =\= 0, \+ has_factor(P,3).
```

```
% has_factor(N,L) :- N has an odd factor F >= L.
```

```
% (integer, integer) (+,+)
```

```
has_factor(N,L) :- N mod L =:= 0.
```

```
has_factor(N,L) :- L * L < N, L2 is L + 2, has_factor(N,L2).
```

(G 2) Queries

Gegeben sei das folgende Prolog Programm:

```
direct(frankfurt,san_francisco).
```

```
direct(frankfurt,chicago).
```

```
direct(san_francisco,honolulu).
```

```
direct(honolulu,maui).
```

```
connection(X, Y) :- direct(X, Y).
```

```
connection(X, Y) :- direct(X, Z), connection(Z, Y).
```

Man beantworte folgende Queries:

```
?- connection(maui, X).
```

```
?- connection(frankfurt,X).
```

```
?- connection(X, maui).
```

(G 3) Multiplikation in Prolog

Man definiere in Prolog ein Prädikat für die Multiplikation. Sie können dafür folgendes Prädikat benutzen:

```
add(X,0,X).    %  $x + 0 = x$ 
```

```
add(X,s(Y),s(Z)) :- add(X,Y,Z).    %  $x + y = z \quad x + s(y) = s(z)$ 
```

Geben Sie den Output folgender Queries:

```
?-mul(s(s(0)),s(s(s(0))),Z).
```

```
?-mul(s(s(0)),s(s(0)),s(s(s(s(s(0)))))).
```

(G 4)

Benutzen Sie das obige Multiplikationsprädikat um die Funktion *Fakultät* zu definieren ($n!$).

Beispiel: ? - fact(s(s(s(0))), F) ergibt $F = s(s(s(s(s(s(0))))))$.

(G 5)

Das Quadrat von 45 ist 2025 und $20 + 25 = 45$. Man finde weitere Paare von Zahlen, die diese Eigenschaft aufweisen.

(G 6)

Definiere ein Prädikat `Palindrom(L)`, welches überprüft, ob ein Wort L ein Palindrom ist oder nicht.