

Occur Check

Übersicht

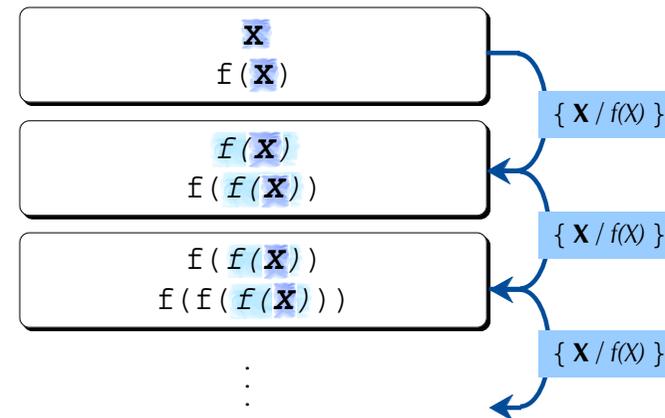
- ◆ Beim Unifizieren kann ein Problem entstehen...

```
?- X = f(X).
```

- ◆ Verbot: Occurs Check
- ◆ Alternative: Zyklische Terme

Occur Check - 1

Substitutionen ad infinitum...



Occur Check - 2

Verbot: Occur Check

Eine Variable darf nicht an einen Term gebunden werden, in der dieselbe Variable vorkommt.

- ◆ Dieser sog. **Occur Check** (Vorkommenstest) wird aus Effizienzgründen von Prolog-Interpretern nicht gemacht.
 - ▶ Programmierende müssen sich eigenhändig um diese (seltenen) Fälle kümmern.
- ◆ Prolog-Implementationen, die den ISO-Prolog-Standard erfüllen, bieten ein besonderes Prädikat für die Unifikation mit *Occur Check* an: **unify_with_occure_check/2**.

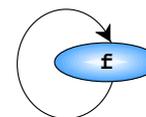
```
?- use_module(library(terms)). % Konsultiere Bibliothek
yes
?- unify_with_occure_check(X,f(X)).
no
```

Occur Check - 3

Alternative: Zyklische Terme

Zyklische Terme

- ◆ sind in SICStus Prolog zugelassen.
- ◆ entstehen durch Unifikationen, die den *Occur Check* verletzen.
 - ▶ Das ergibt Terme, die nicht mehr als Bäume darstellbar sind!
- ◆ erfordern etwas vorsichtigen Umgang, damit man sich beim Verarbeiten dieser Strukturen nicht unendlich tief verliert.



```
?- X = f(X).
X = f(f(f(f(f(f(f(f(...))))))))
yes
```

Occur Check - 4