Prolog-Strukturen

Übersicht

- Woraus besteht ein Prolog-Programm?
  - Fakten, Regeln und Anfragen
  - Kommentare
- Einführen wichtiger Begriffe
- Verschiedene Arten von Termen
- Unifikation
  - Was ist Unifikation?
  - Unifikation in Prolog

Konstrukte in Prolog

Ein Prolog-Programm besteht aus:

- **Fakten:**

  Fido ist ein Hund. `hund(fido).`

  Hans ist Sabrinas Vater. `vater(hans, sabrina).`

- **Regeln:**

  Hunde sind bissig. `bissig(X) :- hund(X).`

An ein solches Programm werden Anfragen gestellt:

- **Anfragen:**

  Ist Fido bissig? `?- bissig(fido).`

  Welche Kinder hat Hans? `?- vater(hans, K).`

Kommentare

```
/* Bla bla.
   Bla bla bla. */
```

Zwischen / und */*

```
% Bla bla.
% Bla bla bla.
```

Ab % bis Zeilenende

Kommentare erhöhen die Verständlichkeit für Menschen

- Prolog-Interpreter ignorieren Kommentare
- ein Programm ohne Kommentare ist nur sehr schwer verständlich

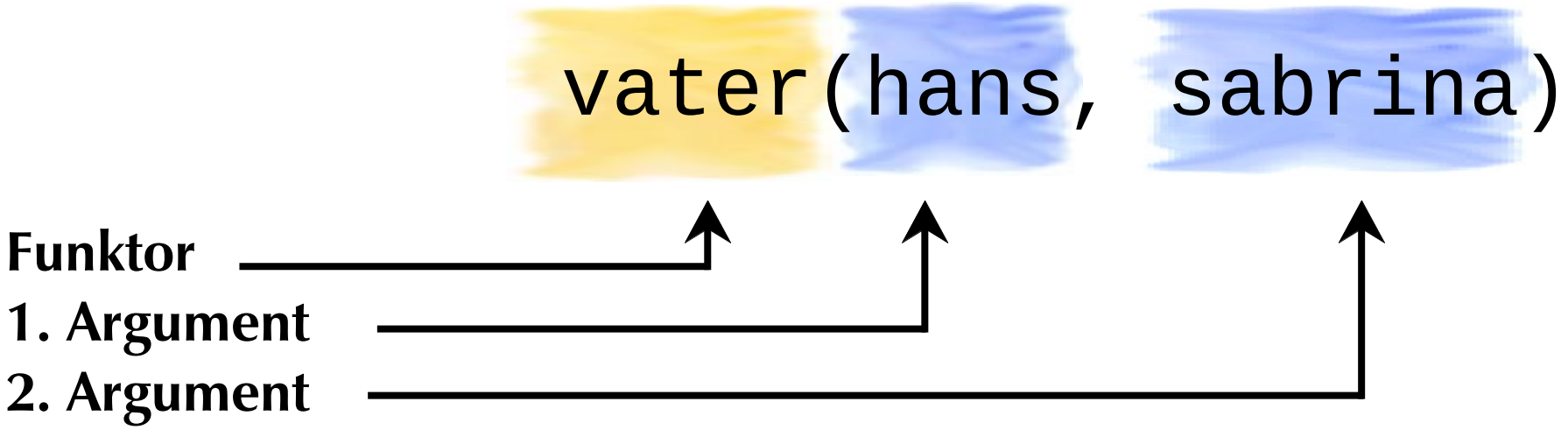
  … vor allem für andere!
- besser zu viel als zu wenig kommentieren

Terme

Ausdrücke wie

- `klara`
- `vater(hans, sabrina)`
- `bissig(X)`

heissen Terme.



Stelligkeit

Die Anzahl der Argumente eines Terms heisst Stelligkeit (manchmal: »Arität«, von engl. *arity*) des Terms.

n	n-stelliger Term
0	`fido`
1	`hund(fido)`
2	`frisst(fido, X)`
3	`gibt(hans, fido, fleisch)`

Stelligkeit

Kurznotation im Prolog-Slang:

- Funktor und Stelligkeit durch Schrägstrich getrennt
- ohne die Argumente zu nennen

n	n-stelliger Term	Funktor/Stelligkeit
0	`fido`	`fido/0`
1	`hund(fido)`	`hund/1`
2	`frisst(fido, X)`	`frisst/2`
3	`gibt(hans, fido, fleisch)`	`gibt/3`

Atomare Terme

Atomare Terme *(atomic terms)* sind nicht aufteilbar.

Schreibweisen in Prolog:

- ein einziges Wort, das mit einem Kleinbuchstaben anfängt
- beliebiger Text zwischen zwei Apostrophen
- Zahl

Beispiele für atomare Terme:

- `klara`
- `a4711_b23`
- `'Dies hier ist ein einziges Atom.'`
- `4711`

Komplexe Terme

Komplexe Terme bestehen aus:

- einem Funktor
- beliebig vielen Argumenten, die atomar oder komplex sein können

Beispiele für komplexe Terme:

- `weiblich(klara)`
- `liebt(hans, klara)`
- `bla(ble, bli, blo, blu, 3, 1, 4, 1, 5, 9, 2, 7)`

Terme dürfen beliebig tief verschachtelt sein:

- `a(b, c, d(e, f(g), h))`

Regeln

muede **:–** ***abends*** **,** ***dunkel*** **.**

Konklusion ← *Prämisse* ∧ *Prämisse*

Wenn *es abends* ***und*** *dunkel ist,* ***dann*** *bin ich müde.*

Eine Regel besteht aus zwei Teilen:

- Unter welchen Voraussetzungen …
  - beliebig viele Terme auf der rechten Seite
- … ein Schluss gezogen werden kann
  - *ein* Term auf der linken Seite

Achtung: Diese Regel sagt nicht, dass es nicht noch andere Gründe für Müdigkeit geben könnte!

Variablen

Variablen sind Platzhalter für Terme.

```
frau(X) :- person(X), weiblich(X).
?- frisst(fido, Was).
```

Schreibweisen in Prolog:

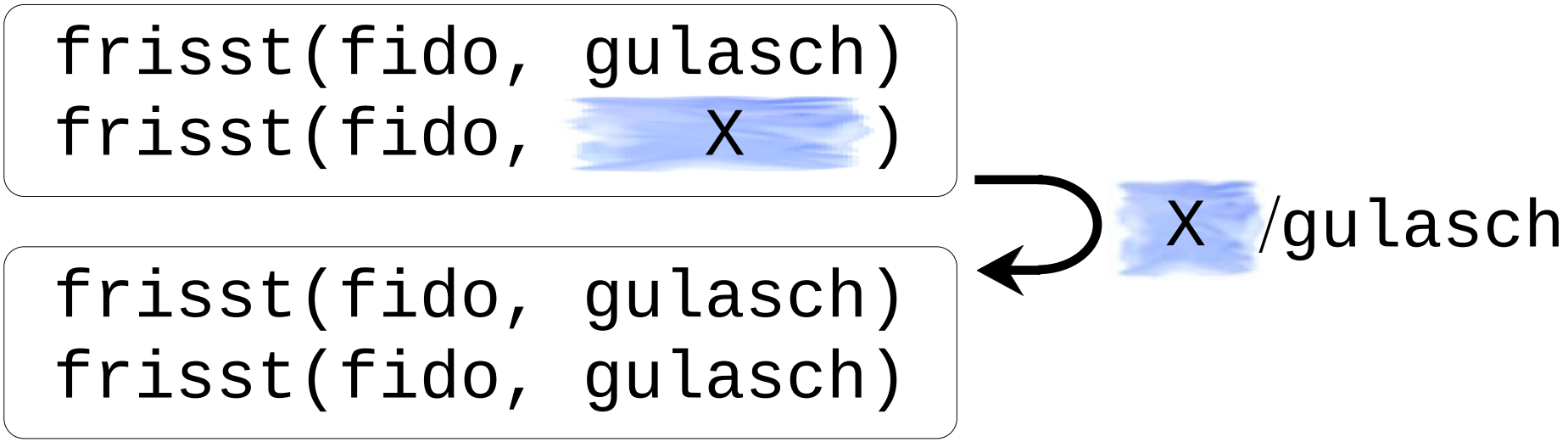
- ein einziges Wort, das mit einem Grossbuchstaben oder _ beginnt
- _ (= **anonyme** Variable)

Beispiele für Variablen:

- `Futter`
- `_futterFuerFido`
- `_`

Unifikation

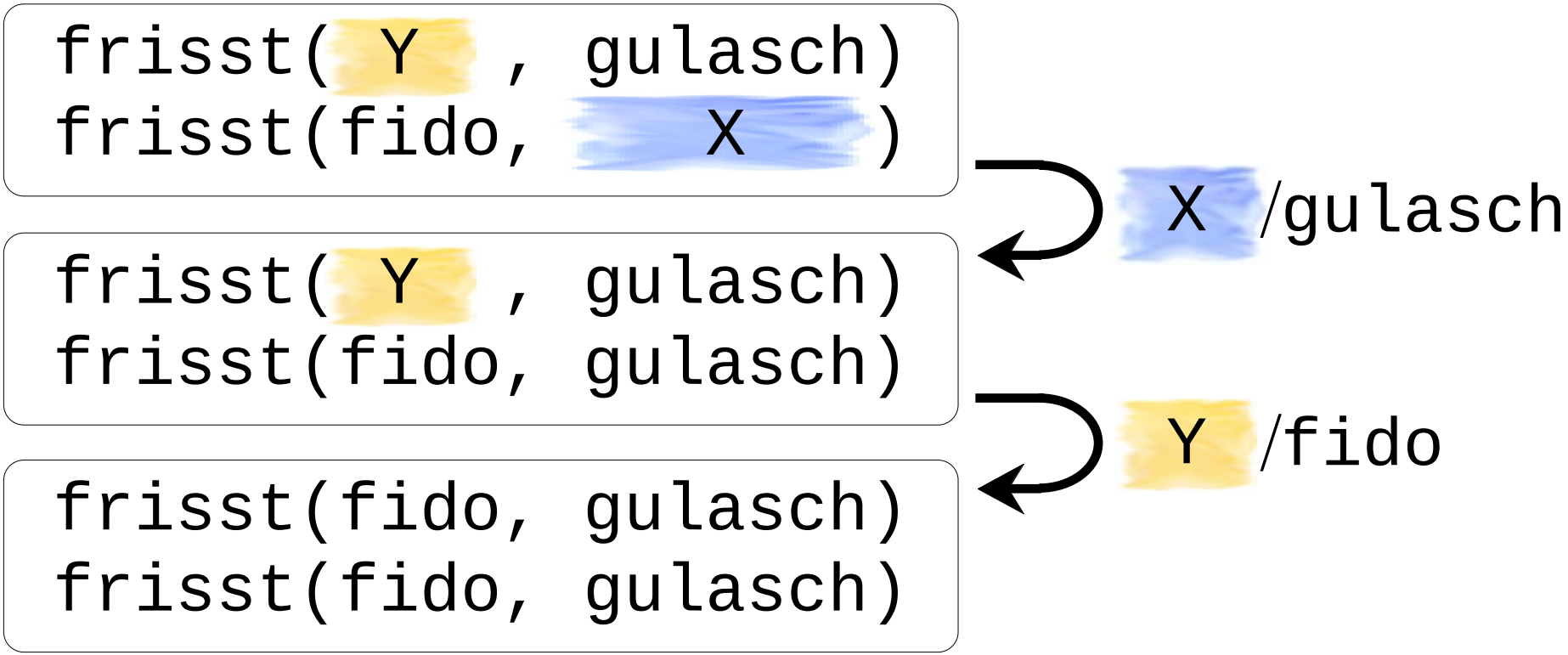
Unifikation versucht, zwei Terme gleich zu machen, indem Variablen (so weit wie nötig) ersetzt werden.



Bei **Substitution** von `X` durch `gulasch` werden die beiden Terme gleich.

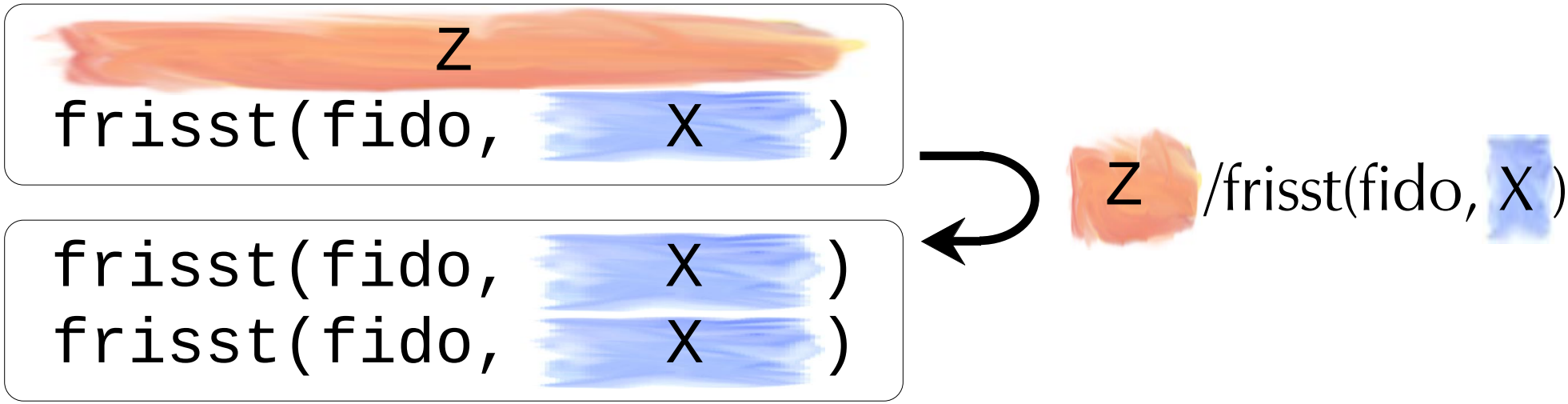
Unifikation

Manchmal müssen mehrere Variablen substituiert werden, um die beiden Terme identisch zu machen:



Unifikation

Es brauchen nicht alle Variablen substituiert zu werden:



Nicht unifizierbare Terme

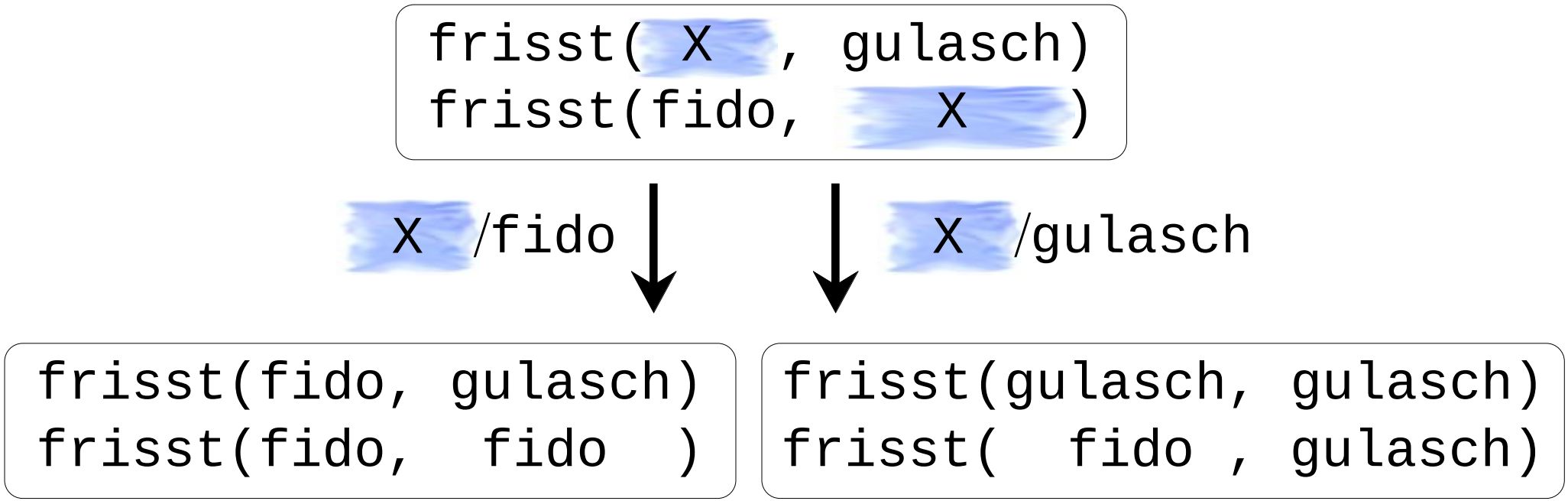
Manchmal gibt es keine Möglichkeit, Variablen zu ersetzen, damit die Terme identisch werden:

```
frisst(fido, gulasch)
frisst(fido, sellerie)
```

Die Unifikation **schlägt fehl** *(unification failure)*.

Nicht unifizierbare Terme

Manchmal gibt es keine Möglichkeit, Variablen zu ersetzen, damit die Terme identisch werden:



Wann sind zwei Terme unifizierbar?

Zwei Terme *S* und *T* sind unifizierbar, genau dann wenn:

- *S* und *T* sind dasselbe Atom,
- oder einer von ihnen ist eine freie Variable
  (in diesem Fall wird die Variable gebunden),
- oder *S* und *T* sind komplexe Terme, wobei
  - *S* und *T* haben denselben Funktor,
  - und *S* und *T* haben dieselbe Stelligkeit,
  - und die einzelnen Argumente sind jeweils paarweise unifizierbar.

Wann sind zwei Terme unifizierbar?

p4711 = p4711

- ja: dasselbe Atom

X = fido

- ja: X ist Variable

X = Y

- ja: X ist Variable (wobei X/Y) · ***oder:*** ja: Y ist Variable (wobei Y/X)

frisst(fido, X) = frisst(Y, gulasch)

- derselbe Funktor, dieselbe Stelligkeit
- 1. Argument: fido und Y sind unifizierbar (wobei Y/fido)
- 2. Argument: X und gulasch sind unifizierbar (wobei X/gulasch)

Aufruf des Prolog-Unifikators

Der Unifikator *(unifier)* ist ein wichtiger Teil jedes Prolog-Interpreters.

```
?- gulasch = gemuese.
no
```

nicht unifizierbar

```
?- gulasch = gulasch.
yes
```

unifizierbar, ohne Variablenbindung

```
?- frisst(fido, X) =
   frisst(Y, gulasch).
X = gulasch, Y = fido
```

unifizierbar, mit Variablenbindung