

Informationen zu Vorlesung und Proseminar

Nichtprozedurale Programmierung:

Logische Programmierung

Elmar Eder

Sommersemester 2021

1 Zeit und Ort

Vorlesung Montag 12:15–14:00

Proseminar Gruppe 1 Montag 14:15–15:00

Proseminar Gruppe 2 Montag 15:15–16:00

<https://edumeeet.geant.org/elmar-eder>

Wenn Edumeeet nicht funktioniert, dann

<https://uni-salzburg.webex.com/meet/elmar.eder>

2 Website von Vorlesung und Proseminar

https://www.cosy.sbg.ac.at/~eder/lehre/Logische_Programmierung/

3 Ablauf und Prüfungen

Vorlesung Prüfung (Klausur) am Donnerstag, 29.7.2021 zwischen 13:00 und 15:00 online auf <https://edumeeet.geant.org/elmar-eder>

4 Arbeiten mit Prolog

Installieren von Prolog In der Vorlesung und im Proseminar verwenden wir die logische Programmiersprache Prolog. Am Fachbereich ist GNU-Prolog <http://gprolog.org/> und SWI-Prolog <https://www.swi-prolog.org/> unter Linux installiert. Beide sowie zahlreiche andere Prolog-Systeme sind im Internet frei erhältlich. Wir werden GNU-Prolog verwenden und ich empfehle Ihnen, es sich auf Ihren eigenen Computer herunterzuladen und zu installieren. Leider ist die aktuelle Version des Debian-Pakets für GNU-Prolog (gprolog) fehlerhaft und der Aufruf von gprolog führt schon für triviale

Prologprogramme zu einer Fehlermeldung. Das betrifft auch Ubuntu und Linux Mint, da die auf Debian basieren. Für diese Linux-Systeme muss man gprolog selbst kompilieren. Auf https://www.cosy.sbg.ac.at/~eder/lehre/Logische_Programmierung/compile_gprolog.html gibt es dazu Informationen. Wenn Sie einen Notebook-Computer haben, können Sie ihn mit installiertem Prolog in die Vorlesung und ins Proseminar mitbringen.

Testen von Prolog Nach dem Installieren probieren Sie Prolog bitte aus! Wenn Sie zum Beispiel GNU-Prolog unter Linux verwenden, starten Sie ein Shell-Fenster (Terminal). Mit dem Shell-Prompt `$` fordert die Shell Sie auf, einen Linux-Befehl einzugeben. Mit dem Befehl `gprolog` rufen Sie GNU-Prolog auf. Mit dem Prompt `?-` fordert Prolog Sie auf, eine Anfrage zu stellen. Ein Dialog mit Prolog kann etwa so ausschauen (Darstellung hier: Benutzereingaben **schwarz**; Rückmeldungen vom Computer **blau**, irrelevante oder systemabhängige Details weggelassen oder durch drei Punkte `...` ersetzt):

```
$ gprolog                                     (Prolog aufrufen)
...                                           (Willkommensmeldung von Prolog)
?- p(a).                                       (Erste Anfrage stellen)
uncaught exception: error(existence_error(procedure,p/1),top_level/0)
?- 2 < 3.                                     (Zweite Anfrage stellen)
yes                                           (Prologs Antwort)
?- 3 < 2.                                     (Dritte Anfrage stellen)
no                                           (Prologs Antwort)
?- halt.                                     (Prolog verlassen)
$                                           (Sie sind wieder in der Shell.)
```

Bitte testen Sie Ihr Prolog mit diesem Dialog! Auf die erste Anfrage `?- p(a).` gibt Prolog eine Fehlermeldung, da wir kein Programm geladen haben und daher das Prädikat `p` undefiniert ist. Um Prolog zu verlassen und in die Shell zurückzukehren, kann man auch einfach nach dem Prolog-Prompt `?-` das Datei-Ende-Steuerzeichen (unter Linux: Control-d oder Strg-d) eingeben.

Probieren Sie Prolog bitte mit einem einfachen Programm aus! Dazu erstellen Sie ein Verzeichnis für Ihre Prolog-Programme, gehen in der Linux-Shell in dieses Verzeichnis und erstellen darin mit Ihrem Lieblings-Editor eine Datei `lebt.pl` mit dem Inhalt

```
ist_tier(steinbock).
ist_tier(gemse).
ist_tier(hai).

ist_pflanze(edelweiss).
ist_pflanze(enzian).
ist_pflanze(seetang).

lebt_in(steinbock,gebirge).
```

```
lebt_in(gemse,gebirge).
lebt_in(hai,meer).
lebt_in(edelweiss,gebirge).
lebt_in(enzian,gebirge).
lebt_in(seetang,meer).
```

Dann laden Sie diese in Prolog und stellen Anfragen an Prolog. Der Dialog mit dem Computer könnte etwa bei Benutzung des Editors Emacs folgendermaßen ausschauen:

```
$ mkdir prolog-programme                (Verzeichnis erstellen)
$ cd prolog-programme/                  (Ins Verzeichnis gehen)
$ emacs lebt.pl                          (Datei editieren)
$ gprolog                                (Prolog aufrufen)
?- [lebt].                               (Datei laden/kompilieren)
...                                     (Computer meldet Kompilierung)
?- ist_pflanze(Pflanze), ist_tier(Tier),
lebt_in(Pflanze,Lebensraum), lebt_in(Tier,Lebensraum).    (Anfrage stellen)

Lebensraum = gebirge                    (Prolog gibt eine erste Antwort. Wenn
Pflanze = edelweiss                      Sie weitere Antworten sehen wollen,
Tier = steinbock ;                       geben Sie einen Strichpunkt ; ein!)

Lebensraum = gebirge                    (Prolog gibt die zweite Antwort. Wenn
Pflanze = edelweiss                      Sie weitere Antworten sehen wollen,
Tier = gemse ;                           geben Sie einen Strichpunkt ; ein!)

Lebensraum = gebirge                    (Prolog gibt die dritte Antwort. Wenn Sie
Pflanze = enzian                          keine weiteren Antworten sehen wollen,
Tier = steinbock                          drücken Sie die RETURN-Taste!)

?-                                       (Nun können Sie eine neue Anfrage eingeben.)
```

Mit der Anfrage in diesem Beispiel haben Sie Prolog nach einer Pflanze und einem Tier gefragt, die sich denselben Lebensraum teilen. Prolog gibt Ihnen eine Antwort

```
Lebensraum = gebirge
Pflanze = edelweiss
Tier = steinbock
```

Aber es gibt weitere Antworten. Wenn Sie die nächste Antwort sehen wollen, geben Sie bitte einen Strichpunkt (;) ein! Wenn Sie keine weiteren Antworten sehen wollen, betätigen Sie die RETURN-Taste! Dann kommt wieder der Prolog-Prompt `?-` und Sie können eine neue Anfrage stellen. Im Beispiel oben haben wir Prolog durch zweimalige Eingabe eines Strichpunkts veranlasst, uns die ersten drei von insgesamt fünf Antworten zu zeigen. Durch Drücken der RETURN-Taste haben wir wieder den normalen Prolog-Prompt `?-` bekommen und können eine neue Anfrage stellen oder Prolog verlassen.

Hinweis: Verwenden Sie bitte in Ihren Programmen, wenn nicht explizit anders verlangt, nicht Umlaute oder scharfes s (ä, ö, ü, ß, Ä, Ö, Ü), sondern stattdessen ae, oe, ue, ss, Ae, Oe, Ue!

Noch ein Hinweis zur Benutzung des Editors GNU-Emacs zum Editieren von Prolog-Dateien: Um zu verhindern, dass Emacs fuer Prolog-Dateien mit Datei-Endung `pl` den Perl-Mode verwendet, fügen Sie in Ihrem Home Directory in der Datei `.emacs` die Zeile `(add-to-list 'auto-mode-alist ('("\\.pl\\'" . prolog-mode))` ein!

5 Literatur

William F. Clocksin and Christopher S. Mellish: Programming in Prolog, Fifth Edition. Springer Verlag, 2003.

P. Deransart, A. Ed-Dbali, L. Cervoni: Prolog: The Standard, Reference Manual. Springer-Verlag