

Praktikum zur Lehrveranstaltung „Künstliche Intelligenz“ Hinweise zum Umgang mit der Visual Prolog IDE¹

!!! Achtung !!! Dieses Dokument enthält lediglich ergänzende Hinweise zur separat erhältlichen Praktikumanleitung. Lesen Sie zunächst die „Aufgabenstellungen und Anleitung“ zum Praktikum.
Dort finden Sie einen Querverweis auf das vorliegende ergänzende Material.

Installation

Zur Vorbereitung des Praktikums laden Sie bitte die beiden Dateien SETUP.EXE und TEST.EXE aus der Rubrik „Lehre“ auf der Internetseite des Fachgebietes „Künstliche Intelligenz“ des Institutes für Theoretische und Technische Informatik der Fakultät für Informatik und Automatisierung herunter².

Systemvoraussetzungen:

- Intel® Pentium™ oder besserer Prozessor der x86-Familie
- 16 MB RAM
- ca. 75 MB freier Festplattenspeicher
- Microsoft® Windows™ 95, 98, 2000 oder NT 4.0

Installation:

Starten Sie zunächst das Programm SETUP.EXE. Dieses installiert Visual Prolog v.5.1 Personal Edition³ auf Ihrem Computer. Führen Sie nun TEST.EXE aus. Bei dieser Datei handelt es sich um ein selbstentpackendes Archiv, das die Testumgebung enthält. Belassen Sie den Installationspfad nach Möglichkeit auf C:\VORFAHRT.
Führen Sie zur Sicherheit einen Neustart des Rechners durch.
Danach starten Sie ViP mit

STARTMENÜ ? PROGRAMME ? VISUAL PROLOG PERSONAL EDITION ? VIP32.

Das Programm begrüßt Sie mit einem Dialog, welcher lizenzrechtliche Hinweise enthält. Dieser Dialog ist Bestandteil eines jeden Programms, das mit der Personal Edition erstellt wurde und muss vor jedem Programmstart bestätigt werden.

Wenn Sie möchten, können Sie nun unter

OPTIONS ? GLOBAL (.INI) ? ENVIRONMENT

die Option AUTO SAVE EDITORS ON CLOSE anwählen. Der Inhalt des Quelltexteditors wird dann automatisch beim Schließen des Fensters gespeichert.
Bitte ändern Sie keine der anderen Optionen, da sonst nicht gewährleistet ist, dass die Testumgebung korrekt arbeitet.

¹ Dieses Dokument bezieht sich auf das Entwicklungswerkzeug „Visual Prolog v.5.1 Personal Edition for Windows 32“, © 1984-1999 Prolog Development Center A/S und ist nicht Bestandteil der offiziellen Dokumentation zu diesem Produkt.

² Adresse: [HTTP://WWW.THEOINF.TU-ILMENAU.DE/KI/KI/AUSBILDUNG.HTML#LEHRE](http://www.theoinf.tu-ilmeneau.de/ki/ki/ausbildung.html#lehre)

³ im folgenden als ViP bezeichnet

Unterschiede zwischen ViP und Standard-Prolog

Die meisten Programmierumgebungen für Prolog enthalten einen Interpreter zur Erzeugung des Maschinencodes. Das hat folgende Konsequenzen:

- Der Typ einer Variablen erschließt sich erst zur Laufzeit unmittelbar aus ihrer Instanz.
- Prädikate werden dem System erst bekannt, wenn das Programm bis zu der Stelle ausgeführt wurde, an welcher der erste Aufruf einer zugehörigen Klausel erfolgt.
- Es liegt keine sogenannte strenge Typbindung vor, somit ist es z. B. möglich, eine Liste zu generieren, deren Elemente Instanzen verschiedener Datentypen (z. B. INTEGER und STRING) sind.

Im Gegensatz dazu handelt es sich bei ViP um einen Compiler, der folgende Besonderheiten aufweist:

- Es herrscht strenge Typbindung, d. h. ohne vorherige Konvertierung sind keine Zuweisungen zwischen Variablen unterschiedlichen Typs möglich. Selbst bei Wertzuweisungen zwischen „typähnlichen“ Variablen (z. B. INTEGER und BYTE) ist äußerste Vorsicht geboten. Des Weiteren sind keine Listen mit Elementen verschiedener Datentypen möglich⁴.
- Ebenso werden für Prädikate Prototypen angelegt. Dies hat den Vorteil, dass im Programm zur Laufzeit bereits alle Prädikate bekannt sind. Somit ist die Möglichkeit eines Prädikataufrufes unabhängig davon, an welcher Stelle im Programm die zugehörigen Klauseln definiert sind.

Vereinfachte Struktur eines Prolog-Programms in ViP:

DOMAINS

```
% IN DIESEM ABSCHNITT WERDEN DIE DOMÄNEN  
% (DATENTYPEN DER ARGUMENTE VON PRÄDIKATEN) DEKLARIERT
```

```
% DOMÄNEN MIT SPEZIELLEN ELEMENTEN:  
% {DAS SEMIKOLON (;) STEHT FÜR DISJUNKTION („ODER“).}  
<NAME DER DOMÄNE> = <WERT 1>; <WERT 2>; ...
```

```
% DOMÄNENDEFINITION ALS AUSWAHL VON ELEMENTEN EINER ANDEREREN DOMÄNE:  
<NAME DER DOMÄNE> = <WERT 1>; <WERT 2>; ... OF <NAME DER REFERENZDOMÄNE>
```

```
% LISTE AUS ELEMENTEN EINER ANDEREN DOMÄNE:  
<NAME DER DOMÄNE> = <NAME DER REFERENZDOMÄNE>*
```

PREDICATES

```
% IN DIESER SEKTION WERDEN DIE PRÄDIKATE DEKLARIERT  
% (VERGLEICHBAR MIT PROTOTYPEN IN C)
```

```
<NAME DES PRÄDIKATES>(<DOMÄNE DES 1. PARAMETERS>,<DOMÄNE DES 2. PARAMETERS>,...)
```

CLAUSES

```
% HIER FOLGEN DIE EIGENTLICHEN KLAUSELN
```

```
<NAME DES PRÄDIKATES> (<VARIABLE AUS DOMÄNE 1>, <VARIABLE AUS DOMÄNE 2>, ...):-  
    <KLAUSELKÖRPER>.
```

⁴ Die geforderte Typenreinheit ist auch der Grund für die „umständliche“ Definition der Domäne KREUZUNG in der Testumgebung. Hier sind die Elemente der Liste vom einheitlichen Typ UMGEBUNG.

Ein kleines Beispiel:

DOMAINS

*% Eine Domäne aus einzelnen Elementen des Referenzdomäne string:
monitor = „Monochrom“; „Color“ OF string*

*% Eine „neue“ Domäne:
rgb = rot; gruen; blau*

*% Eine Liste aus Elementen der Domäne rgb:
farbliste = rgb**

PREDICATES

*% Deklaration des Prädikates hat_farben/2:
hat_farben (monitor,farbliste)*

CLAUSES

*% Die Klauseln des Prädikates hat_farben/2, in diesem einfachen Fall drei Fakten:
hat_farben („Monochrom“, [gruen]):-!.
hat_farben („Color“, [rot, gruen, blau]):-!.
hat_farben (_, []).*

GOAL

*% Das Ziel des Programms (der initiale Prädikataufruf):
hat_farben („Color“, Welche), write (Welche).*

Anmerkung:

Schauen Sie sich bei Gelegenheit auch die einfachen Beispiele an, die sich im Unterverzeichnis EXAMPLES im Programmpfad zu ViP befinden. Sie sind leicht nachzuvollziehen und zeigen die vorgegebene Struktur des Quelltextes besonders gut.

Sollten Sie Interesse an weiteren Informationen zum Umgang mit Visual Prolog haben, so sollten Sie sich auch die Internetseiten des Prolog Development Center unter

[HTTP://WWW.PDC.DK](http://www.pdc.dk)

anschauen.

Bedienung

Die Integrierte Entwicklungsumgebung ist intuitiv erlernbar und bietet unter dem Menüpunkt HELP eine umfangreiche Online-Dokumentation und Befehlsreferenz.

Um sich mit den Funktionen vertraut zu machen, die Sie für das Praktikum benötigen, wählen Sie bitte den Menüpunkt PROJECT ? OPEN PROJECT...

Öffnen Sie die Datei VORFAHRT.PRJ aus dem Verzeichnis, in das Sie die Testumgebung entpackt haben (Standard: C:\VORFAHRT). Im geöffneten Projektfenster sehen Sie links eine senkrechte Buttonleiste, der Button MODULE ist angewählt. Bitte lassen Sie diesen Button stets aktiviert. Die anderen Sektionen des Projekts haben für die Lösung Ihrer Aufgabe keine Relevanz. Wählen Sie im Projektfenster den Eintrag STUDENT.PRO an und klicken Sie rechts daneben auf den EDIT-Button.

Sie sehen nun die vorformatierte Quelltextdatei vor sich, die Sie um Ihre Variante der Lösung der gestellten Aufgabe ergänzen müssen⁵.

Der Editor verfügt über sämtliche Funktionen vergleichbarer Windows-Editoren, wie Kopieren, Ausschneiden, Einfügen usw. Außerdem bietet er eine Syntax-Hervorhebung, wodurch Variablen, Konstanten und Sektionsschlüsselwörter sich farblich vom restlichen Quelltext abheben.

Sollten Sie, wie weiter oben beschrieben, die Funktion AUTO SAVE EDITORS ON CLOSE aktiviert haben, wird der Inhalt des Editorfensters automatisch gespeichert, wenn es geschlossen wird. Ansonsten sollten Sie dies von Zeit zu Zeit „manuell“ mit Hilfe von PROJECT ? SAVE PROJECT tun.

Haben Sie die Datei STUDENT.PRO um den benötigten Code ergänzt, starten Sie den Compiler mit PROJECT ? BUILD

Dies löst zwei Vorgänge aus: Zum einen wird der von Ihnen geänderte Quelltext automatisch abgespeichert, zum anderen wird das Programm in Maschinencode übersetzt.

Sollte der Quelltext Fehler aufweisen, öffnet sich ein Fenster, in welchem die Fehlermeldung, die Fehlernummer und andere Informationen erscheinen. Anhand der Fehlernummer können Sie in der Online-Hilfe eine genauere Beschreibung der betreffenden Fehlermeldung und Ihrer möglichen Ursachen finden.

Nach einem Doppelklick auf eine der angezeigten Fehlermeldungen wird die Quelltextdatei geöffnet und der Cursor auf die Stelle gesetzt, an welcher der Fehler auftrat.

Sind schließlich alle syntaktischen und semantischen Fehler beseitigt, können Sie das Programm mit PROJECT ? RUN starten.

Haben Sie nach einer endlichen, aber ausreichend großen Anzahl Mausklicks auf STARTEN noch keine roten Fahrzeuge gesehen, und wissen außerdem sehr genau, warum das so ist, dann sei Ihnen schon jetzt zum (bald) (ü)be(r)standenen Praktikum gratuliert.

Kopieren Sie die Datei STUDENT.PRO auf eine intakte, virenfreie 3,5“ Diskette und bringen Sie diese zum Praktikumstermin mit.

⁵ Bitte ändern Sie keinesfalls Einträge in anderen eventuell vorhandenen Dateien, die Testumgebung würde dann nicht korrekt funktionieren.