Fachbereich Informatik
Universität Kaiserslautern
Postfach 3049
D-6750 Kaiserslautern 1, W. Germany

SEKI-Working Paper

micro-UNIXPERT
Ein wissensbasiertes System zur Behandlung von Problemen bei UNIX-Druckauftraegen

Michael Lessel

Mai 1986 SWP-86-04

micro-UNIXPERT: EIN WISSENSBASIERTES SYSTEM ZUR BEHANDLUNG VON PROBLEMEN BEI UNIX-DRUCKAUFTRAEGEN

Michael Lessel

Fachbereich Informatik
Universitaet Kaiserslautern
Postfach 3049
D-6750 Kaiserslautern 1
W. Germany

uucp: unido!uklirb!lessel
- oder -
lessel@uklirb.UUCP

Projektarbeit
(Betreuer : Harold Boley)

Mai 1986

A B S T R A C T

micro-UNIXPERT (UNIX^ - PERiphery - Tester) is an interactive, knowledge-based diagnosis system for the treatment of printer hardware faults and of other problems which can appear in connection with printing (user errors and software faults).
If possible, the system both generates diagnoses and gives proposals for clearing. The entire knowledge of micro-UNIXPERT consists of a set of PROLOG-like facts and rules.
The system is implemented in LISPLOG, a LISP/PROLOG integration running in FRANZ LISP under the UNIX 4.2 BSD operating system on a VAX 11/750 .
After a general introduction into the problems of diagnosis systems and a discussion of the most important micro-UNIXPERT features (knowledge acquisition, positive and negative facts, user acceptance, inference mechanism), the method of operation of the system is explained by giving full details of the LISPLOG implementation.

---

^ UNIX is a registered trade-mark of Bell Laboratories

This research was supported by the
Deutsche Forschungsgemeinschaft, Sonderforschungsbereich 314,
"Kuenstliche Intelligenz - Wissensbasierte Systeme".

Inhalt

Felix, qui potuit rerum
cognoscere causas

Vergil, Georgica 2, 490

1. Grundlagen

1.1 Vorueberlegungen

Soll ein Expertensystem konstruiert werden, ist es wichtig einen abgegrenzten Problembereich zu haben, sodass moeglichst wenig Wechselwirkungen nach aussen bestehen. Beispielsweise gilt dies in der medizinischen Diagnostik fuer den Bereich der Ohrenerkrankungen (vgl. [Wetter 1983]): Erkrankungen des Ohrenbereichs haben kaum Einfluss auf den Koerper, umgekehrt haben Erkrankungen des Koerpers kaum Auswirkungen auf den Ohrenbereich. Dieses Beispiel laesst sich auf die Rechnerperipherie uebertragen, speziell auf Drucker. Auch Drucker bilden einen eigenen, abgegrenzten Bereich mit relativ wenig Wechselwirkungen mit dem Rechner.
Dennoch handelt es sich bei der Kombination Rechner-Drucker-Mensch um ein recht komplexes System, sodass bei der Durchfuehrung eines Druckbefehls Fehler verschiedenster Art auftreten koennen. Der Benutzer, dem meist der Ueberblick ueber das gesamte Betriebssystem fehlt, hat dann oft Schwierigkeiten, die Ursache des aufgetretenen Fehlverhaltens zu diagnostizieren, denn die aufgetretenen Stoerungen koennen im Hardware- oder Softwarebereich liegen oder auch von einem Bedienungsfehler des Benutzers verursacht worden sein.
Um moeglichst jeden Fehler in diesem Mensch-Maschine-System zu diagnostizieren, bedarf es eines Experten (es stellte sich Gebhard Przyrembel zur Verfuegung), der ein zusammenhaengendes Modell der gesamten Rechenanlage hat. Der Experte benutzt eine bestimmte Problemloesungsstrategie, um die Ursachen eines aufgetretenen Fehlverhaltens zu diagnostizieren. Dabei kann er oft anhand einfacher Regeln bestimmten Symptomen (Fehlverhalten) eine Ursache (Diagnose) zuordnen. Bei komplexen Fehlerursachen, wenn es zum Beispiel viele Diagnosealternativen gibt, besteht die Moeglichkeit einer solch einfachen Zuordnung nicht. Der Experte muss dann versuchen, anhand bestimmter Aktionen genuegend Information zu akquirieren, um den Bereich der potentiellen Diagnosen sukzessive soweit wie moeglich einzuschraenken, bis

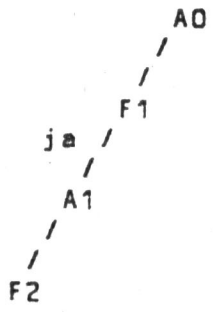
eine Diagnose gestellt werden kann. Diese Art der Vorgehensweise des Experten laeuft zum Teil schematisch ab. Um ein regelbasiertes System zu erhalten, muessen diese Diagnose-Schemata in Regelform repraesentiert werden. Im folgenden wird eine Moeglichkeit vorgestellt diese Repraesentation auf dem Weg ueber Entscheidungsbaeume durchzufuehren.

Die Transformation von Entscheidungsbaeumen in Regeln laesst sich dann formal wie folgt durchfuehren:
Sei B = (V1,V2,E1,E2) ein binaerer Baum mit Bipartition (V1,V2).
B besitzt zwei Arten von Knoten V1 und V2, wobei V1 eine Menge von Entscheidungsknoten und V2 eine Menge von Aktionsknoten ist.
Nach einem Entscheidungsknoten wird in Abhaengigkeit davon, ob die Bedingung im Entscheidungsknoten zu 'wahr' oder 'falsch' ausgewertet wurde, zu einer Kante aus der Menge E1 oder aus der Menge E2 verzweigt.
In einem Aktionsknoten wird eine bestimmte Aktion ausgefuehrt, wobei auch die 'leere Aktion' zu der Menge der Aktionen gehoert.
Darueberhinaus besitzt B zwei Arten von Kanten E1 und E2, wobei eine Kante aus E1 durchlaufen wird, wenn die Bedingung im inzidenten Vaterknoten der Kante zu 'wahr' und eine Kante aus E2 durchlaufen wird, wenn die Bedingung zu 'falsch' ausgewertet wurde.
Die Blaetter des Baumes sind Entscheidungsknoten, bei denen keine Alternative mehr zur Wahl steht, in unserem Fall also eine Diagnose.
Betrachtet man nun einen Pfad von der Wurzel des Baumes bis zu einem Blatt, so sind alle inneren Knoten Praemissen (hier: Aussagen ueber Symptome) und das entsprechende Blatt die Konklusion einer zu erzeugenden Regel. Die Mengenzugehoerigkeit einer entscheidungsknoteninzidenten 'Sohnkante' gibt dabei an, ob die Bedingung im Entscheidungsknoten 'wahr' oder 'falsch' sein musste, um auf dem betrachteten Pfad zu einer Konklusion zu gelangen.
Mit Hilfe der aufgefuehrten Baumstruktur laesst sich ein Problem sehr gut strukturieren, womit bei der Wissensakquisition ein Hoechstmass an Vollstaendigkeit erreicht werden kann.

Beispiele

Im Beispiel auf der naechsten Seite werden Aktionsknoten durch den Buchstaben A und Entscheidungsknoten durch den Buchstaben F gekennzeichnet.
Kanten aus E1 werden durch das Wortsymbol 'ja', Kanten aus E2 durch das Wortsymbol 'nein' gekennzeichnet.

Dabei bedeutet



Fuehre A0 aus und pruefe F1,
wenn F1 gilt, dann fuehre A1 aus und pruefe F2

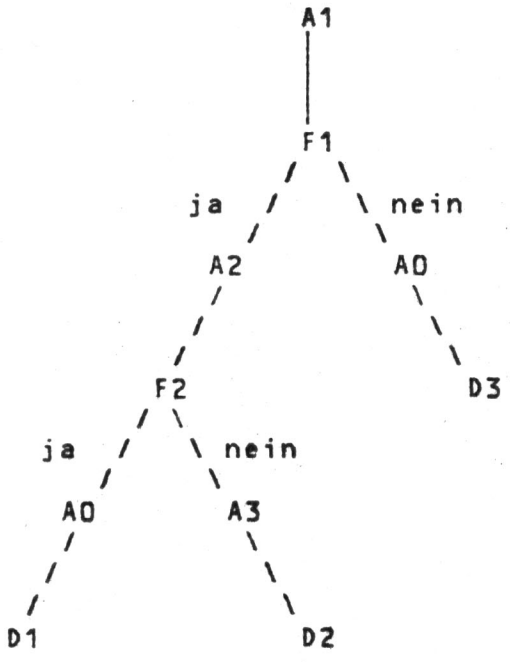
Im Beispiel treten folgende Knoten auf:

Symptome : F1 , F2

Diagnosen : D1 , D2 , D3

Aktionen : A0 , A1 , A2 , A3

B:



B transformiert in Regelform ergibt :

```
IF F1 AND F2 THEN D1

IF F1 AND NOT(F2) THEN D2

IF NOT(F1) THEN D3
```

In einem weiteren Beispiel soll auf den Problembereich "Druckerstoerungen" eingegangen werden.
Dazu wird ein kleiner Teilbereich (simplifiziert) eines Entscheidungsbaumes von micro-UNIXPERT (UNIX-PERiphery-Tester) ausgewaehlt. Im Beispiel werden folgende Bezeichnungen verwendet:

Symptome:

F1	Es gab eine Warnung des Betriebssystems
F2	Der Druckauftrag befindet sich an der ersten Position der Warteschlange
F3	Der Druckauftrag hat die Warteschlange wieder verlassen

Diagnosen:

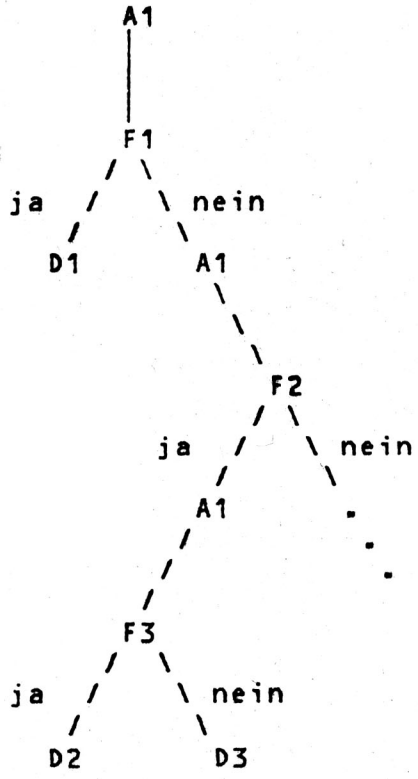
D1	Es liegt am SPOOLER DAEMON
D2	Der Drucker ist nicht eingeschaltet oder es bestehen nicht alle Verbindungen
D3	Der Drucker steht nicht auf ONLINE

Aktionen:

A1	Einsehen der Druckerwarteschlange

Auf der folgenden Seite sind der Entscheidungsbaum und die Regeln, die durch die Entscheidungsbaumtransformation aus diesem entstanden sind, dargestellt:

Entscheidungsbaum:



Transformation in Regelform ergibt:

IF F1 THEN D1

IF NOT(F1) AND F2 AND F3 THEN D2

IF NOT(F1) AND F2 AND NOT(F3) THEN D3

1.2 Fehlerursachen

1.2.1 Der UNIX-Druckbefehl lpr

Der UNIX-Druckbefehl lpr hat folgende Syntax:

```
lpr [- Option] Dateiname
```

Beispiel

```
lpr -Plpb14 Druckdatei
```

Um Dateien auszudrucken, benutzt lpr einen SPOOLER DAEMON. Muss ein Druckjob warten, weil gerade ein anderer laeuft, wird er in eine Warteschlange eingereiht. Die Warteschlange ist mit dem UNIX-Befehl lpq einzusehen. Es ist moeglich, im lpr-Befehl Optionen anzugeben, mit Hilfe derer man die Ausgabe auf den Drucker beeinflussen kann. Jede einzelne Option wird dabei durch das Zeichen - eingeleitet. Die vollstaendige Liste der moeglichen Optionen ist im UNIX-Programmer`s Manual aufgefuehrt. Hier sollen nur einige wesentliche Faelle beleuchtet werden.
Mit Hilfe der -P Option ist es moeglich, den Druck auf einen bestimmten Drucker umzuleiten. Die im Universitaetsbereich Kaiserslautern von der UNIX-VAX 11/750 ansteuerbaren Drucker in Bau 48 bzw. Bau 14 koennen durch die Optionen -Plp bzw. -Plpb14 angesprochen werden.
Es ist weiterhin moeglich, in der Datei .login einen bestimmten Drucker voreinzustellen. Wird dann im Druckbefehl keine Option angegeben, so wird dieser Drucker angesteuert. Wird keine Option angegeben und ist in der Datei .login kein Drucker voreingestellt, so wird die Datei auf dem Default-Printer (zur Zeit Bau 48) ausgedruckt. Eine explizite -P Option im Befehl hat Vorrang vor der Voreinstellung und diese wiederum vor der Default-Einstellung.
Es kann nun der Bedienungsfehler vorkommen, dass statt der -P Option irrtuemlich die -p Option angegeben wird. Dies fuehrt dazu, dass die auszudruckende Datei nach bestimmten Kriterien formatiert und nicht zu dem Drucker umgeleitet wird, den der Benutzer eigentlich ansprechen wollte. Die Datei wird dann doch auf dem Default-Printer oder dem in der Voreinstellung spezifizierten Drucker ausgedruckt. Die restlichen Optionen sind hier nicht relevant, zumal viele Optionen zwar im UNIX-Programmer`s Manual aufgefuehrt sind, aber in der benutzten Betriebssystemversion (UNIX 4.2 BSD) keine Wirkung haben. Der Fall, dass keine -P Option angegeben wurde, und der Druck daraufhin zu dem in der Voreinstellung spezifizierten oder dem Default-Printer geleitet wird, ist eine der haeufigsten Fehlerursachen fuer "nicht vorhandene" Ausdrucke.
Wird im lpr-Befehl kein Dateiname angegeben, erwartet das Betriebssystem eine Eingabe vom Terminal. Indiz dafuer ist, dass der Systemprompt nach dem Absenden des Befehls nicht erscheint.
Ein Programm, das eine Fehlerursache finden soll, muss natuerlich alle Fehlerfaelle, die mit dem Druckbefehl zusammenhaengen,

abfangen koennen. micro-UNIXPERT benutzt dazu u.a. eine Syntaxanalyse, die Informationen aus dem eingegebenen Druckbefehl extrahiert.

1.2.2 Der Drucker

Drucker sind aus vielerlei Gruenden sehr fehleranfaellig. Mit "Fehler" seien dabei nicht nur interne Hardwarefehler, wie defekte Teile gemeint, sondern auch solche wie unterbrochene Verbindungen (Multiplexer, Kabel). Ein Teil der auftretenden Fehler haengt aber mit einem falschen Verhalten der Benutzer zusammen. Es ist vorgekommen, dass ein Drucker angesteuert wurde, obwohl er ausgeschaltet war oder auf OFFLINE stand. Ausser diesen, vom Benutzer sehr einfach zu behebenden Fehlern, gibt es viele Fehlerursachen, die der unerfahrene Benutzer nicht selbst beheben kann, sodass er sich dann mit dem zustaendigen Betreuer in Verbindung setzen muss. Um eine moeglichst grosse Unabhaengigkeit vom Druckertyp zu erreichen, stellt micro-UNIXPERT seine Diagnose bezueglich Drucker-Hardwarefehlern moeglichst stark abstrahiert von Druckerdetails. Das bedeutet insbesondere, dass die Drucker im allgemeinen ausgewechselt werden koennen, ohne dass dadurch eine Programmaenderung erforderlich wird.

1.2.3 Der Spooler

Wie oben schon erwaehnt, benutzt das UNIX-Betriebssystem einen SPOOLER DAEMON um eine Datei auszudrucken. Treten in diesem Bereich Softwarefehler auf, ist ein Ausdrucken unmoeglich. Der Benutzer kann den Fehler in diesem Fall nicht beheben und muss meist bis zum naechsten Hochfahren des Systems warten, um wieder etwas ausdrucken zu koennen.
Die Palette der moeglichen Fehlerursachen soll an dieser Stelle nicht in extenso erlaeutert werden. Auf jeden Fall sind im Programm die wichtigsten, dem Experten bekannten Fehlerfaelle beruecksichtigt.

2. Das Programm

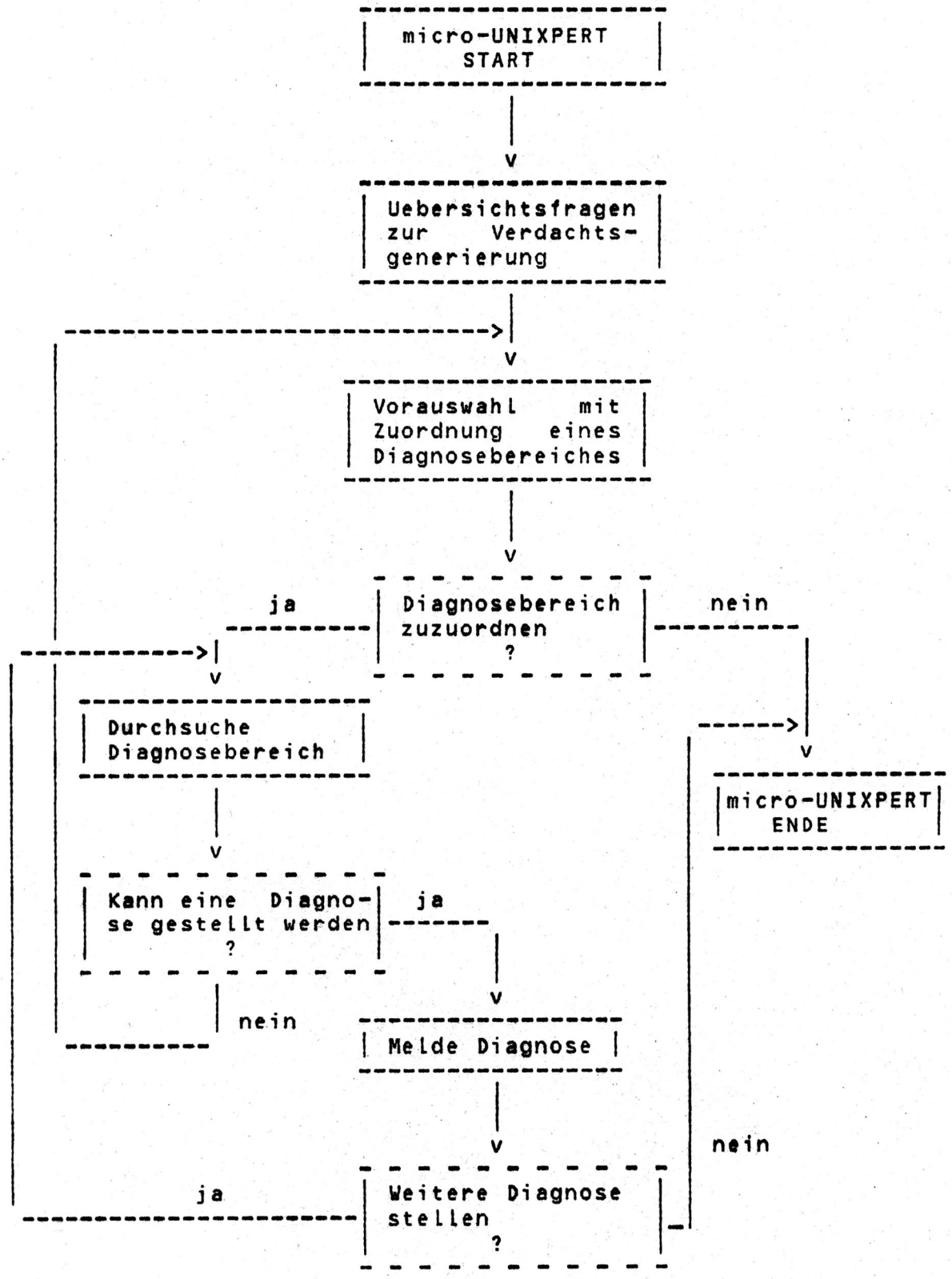
2.1 Warum LISPLOG ?

Die Sprache LISPLOG [Boley & Kammermeier et al. 1985] ist eine LISP/PROLOG - Vereinheitlichung. Die Maechtigkeit des PROLOG-Teils von LISPLOG entspricht in etwa der Maechtigkeit des bekannten Edinburgh - PROLOG [Clocksin & Mellish 1981/84]. In LISPLOG ist aber zusaetzlich zu der logischen Komponente noch die Moeglichkeit gegeben, auf LISP durchzugreifen. Diese Moeglichkeit wird in micro-UNIXPERT sehr oft ausgenutzt, z.B zur Informationsakquisition (siehe Kapitel 2.4.6 : Durchgriffe auf das Betriebssystem) oder fuer die Ein-/Ausgabe (siehe Kapitel 2.4.3).
Ein gewisser Nachteil ergab sich dadurch, dass in LISPLOG kein allgemeiner Cut zur Verfuegung steht, und dieser dann durch n-solutions simuliert werden musste (zur Problematik des allgemeinen Cuts vergleiche [Boley & Kammermeier et al. 1985]).
Ein Vorteil von LISPLOG ist, wie in PROLOG, die Moeglichkeit der inkrementellen Erstellung eines Systems. Speziell auf die Wissensdomaene von micro-UNIXPERT bezogen bedeutete dies, dass das Wissen des Systems durch sukzessives Hinzufuegen von Regeln, schrittweise vergroessert werden konnte. Es hat sich gezeigt, dass LISPLOG als Implementationssprache fuer wissensbasierte (regelbasierte) Systeme mit Backward-Reasoning geeignet ist.
micro-UNIXPERT fehlen einige Charakteristika moderner Expertensysteme, wie z.B eine dynamische Erweiterung der (Regel-) Wissensbasis (Regeln koennen, anders als Fakten, gegenwaertig noch nicht dynamisch in eine LISPLOG-Wissensbasis eingefuegt werden) oder eine Erklaerungskomponente (Wie, Warum - Fragen). Jedoch soll das Programm in einer Ausbaustufe entsprechend erweitert werden (vergleiche Kapitel 3 : Ausblick).
Im Laufe der Arbeit tauchte der Einwand auf, dass das Programm auch in einer prozeduralen Programmiersprache, wie PASCAL oder MODULA II, haette implementiert werden koennen [Gordon 1985]. Selbstverstaendlich waere dies moeglich gewesen, denn beide Sprachen sind turingmaschinenaequivalent, das heisst alles was ueberhaupt programmiert werden kann, kann auch in einer dieser Sprachen programmiert werden. Aber die Benutzung von PASCAL oder MODULA II fuer ein regelbasiertes System, hat gegenueber LISPLOG Nachteile bezueglich des Aufwands in der Programmierung.
micro-UNIXPERT hat als typische Eigenschaften die Informationsakquisition des Systems durch dynamische Erweiterung der Wissensbasis (das Programm veraendert sich selbst!), die Selbsterklaerungsfaehigkeit, logisches Schliessen, das Durchsuchen eines Suchraumes mit Hilfe von Backtracking (vgl. S.20: 'feindiagnose') und, wie oben bereits erwaehnt, die inkrementelle Erweiterbarkeit. Unter anderem muessten diese Punkte durch eine prozedurale Programmiersprache simuliert werden, was einen erheblichen Zusatzaufwand zur Folge haette, der dem LISPLOG-Programmierer erspart bleibt, da LISPLOG alle aufgezaehlten Punkte unterstuetzt.

2.2 Arbeitsweise

Zur Verdeutlichung der Arbeitsweise des Programms wird auf Seite 14 ein grobes Ablaufschema angegeben.
Das Programm startet mit einer Reihe von Uebersichtsfragen zur Verdachtsgenerierung. Die aus den Uebersichtsfragen direkt gewonnenen und mit einer Analyse der Syntax des eingegebenen Druckbefehls abgeleiteten Erkenntnisse dienen der Vorauswahl eines Diagnosebereiches. Mit Hilfe dieser Erkenntnisse (Symptome) ist es nicht moeglich, einen Verdacht direkt auf eine bestimmte Diagnose zu lenken, sondern nur auf eine Menge von Diagnosen. Die Einteilung der Diagnosen in Diagnosebereiche erfolgte aus pragmatischen Gruenden. Beim erstmaligen Durchlauf durch die Vorauswahl kann immer eine Zuordnung eines Diagnosebereiches erfolgen. Dieser Diagnosebereich wird daraufhin nach einer moeglichen Diagnose durchsucht. Das Durchsuchen erfolgt in einer festen Reihenfolge, wobei die Reihenfolge die Prioritaet, der innerhalb eines Diagnosebereiches enthaltenen Regeln bestimmt. Bei einigen Regeln, die aus der Entscheidungsbaumtransformation entstanden sind, wurde die Reihenfolge der Anordnung ausgenutzt, um ein natuerliches Frageverhalten zu erzielen. An dieser Stelle ist Metawissen in die Regelbasis eingearbeitet. Dies sollte zwar im allgemeinen vermieden werden, ergibt sich hier aber implizit aus der sequentiellen Abarbeitung der Regeln durch den Regelinterpreter.
Allgemein gibt es zwei Moeglichkeiten eine Regelmenge abzuarbeiten, naemlich das Forward- und das Backward-Reasoning. Waehrend ein reines Forwardsystem an der Frage des Speicheraufwands scheitert, fehlt einem reinen Backwardsystem die noetige Zielvorgabe, wodurch sich ein Zeitaufwandproblem ergibt. Deshalb ist es wichtig, eine Zielvorgabe zu erhalten. Eine Moeglichkeit ist die Kombination der beiden Mechanismen, etwa die Erstellung einer Verdachtshypothese mit Hilfe von Forward-Reasoning (Vollstaendigkeit, es wird keine Diagnose uebersehen) und anschliessende Ueberpruefung der Hypothese mit Backward-Reasoning (vgl. [Puppe 1983]). Fuer micro-UNIXPERT wurde aber eine andere Moeglichkeit der Vorauswahl gewaehlt, die fuer diesen Problembereich praktischer erschien. Alle Regeln werden durch Backward-Reasoning (goal driven) ausgewertet. Sie werden dabei so weit wie moeglich abgeleitet. Ist es unmoeglich ein Ziel weiter abzuleiten, wird es vom Benutzer erfragt. Nicht ableitbare Ziele sind im Programm immer Symptome, deren Vorhandensein oder Nicht-Vorhandensein hilft, eine Diagnose zu erstellen. Bei allen Fragen, die vom Programm gestellt werden, handelt es sich um qualitative Fragen.
Wie erwaehnt gibt es fuer jedes hier behandelte Symptom nur zwei Alternativen : Entweder es ist vorhanden oder es ist nicht vorhanden. Intermediaere Eigenschaften werden nicht behandelt und sind hier auch wohl nicht sinnvoll. Steht in der Datenbasis ueber ein Symptom keine Information zur Verfuegung, so wird beim Benutzer nachgefragt (vgl. Kapitel 2.4: test,ntest). Das Programm merkt sich den Sachverhalt und kann daraufhin die aktuelle Praemisse der Regel zu 'wahr' oder 'falsch' auswerten.

Das Programm "lernt" also waehrend der Befragung konsultationsspezifische Tatsachen, die es natuerlich vor Beginn der naechsten Konsultation wieder vergessen haben muss. Durch dieses Konzept ist es moeglich, Doppelfragen zu vermeiden. Das bedeutet, dass micro-UNIXPERT keine Frage ueber einen bestimmten Sachverhalt ein weiteres Mal stellen muss, wie es in vielen wissensbasierten Programmen der Fall ist.
Es ist wesentlich, dass sowohl Vorhandensein als auch Nicht-Vorhandensein eines Symptoms gespeichert wird. Wuerde naemlich nur das Vorhandensein eines Symptoms gespeichert, waere die Absenz der Information ueber ein Symptom in der Datenbasis kein Indiz dafuer, dass das Symptom nicht doch aufgetreten ist. Es koennte auch aufgetreten sein, ohne dass die Information darueber im gegenwaertigen Zustand der Konsultation bereits bekannt ist. Da das Programm erst nach und nach Information gewinnt, weiss es zu einem Zeitpunkt nicht alles, was zu wissen ist. Das bedeutet insbesondere, dass "negation by failure", das ja auf der "closed world assumption" basiert, auf die konsultationsspezifischen Faktizitaeten (siehe Kapitel 2.4.1 : Faktizitaeten) nicht anwendbar ist (zur Problematik des "negation by failure" und der "closed world assumption" vergleiche [Reiter 1977], [Kowalski 1983] und [Flannagan 1986]). Das bedeutet, dass die Absenz einer Faktizitaet in der Datenbasis nicht die Negation der Faktizitaet implizieren kann. In solch einem Falle ist eine explizite Negation erforderlich (test,ntest-Konzept). Das heisst, es werden nicht nur positive sondern auch negative Faktizitaeten gespeichert. Aus dem Vergleich der Aussage einer Praemisse mit der entsprechenden Faktizitaet in der Datenbasis kann nun entschieden werden, ob die Aussage der Praemisse wahr oder falsch ist.
Eine explizite Negation waere nicht erforderlich, wenn zunaechst die gesamte moegliche Information akquiriert und dann erst mit der Regelauswertung begonnen wuerde. Das System wuesste dann alles, was zum Stellen einer Diagnose zu wissen ist, und somit wuerde eine "closed world assumption" gelten. Diese Vorgehensweise ist aber nicht praktikabel, da zu viele unnoetige Fragen gestellt werden muessten.
Kann das Programm dann eine Diagnose stellen, wird dies dem Benutzer mitgeteilt und das Programm gibt, wenn moeglich, Abhilfevorschlaege. Der Benutzer hat nun die Moeglichkeit, das Programm zu veranlassen, eine neue Diagnose zu stellen, wenn er mit der Diagnose nicht zufrieden ist. Das kann der Fall sein, wenn eine Fehlerursache nicht eindeutig zu identifizieren ist, wenn also aus den bekannten Praemissen (Symptomen) noch eine weitere Diagnose gestellt werden kann. Das Programm sucht dann nach einer zusaetzlichen Loesung. Wird keine weitere Loesung gefunden, druckt das Programm eine adaequate Meldung und beendet die Konsultation. Ein weiterer Grund das Programm zu veranlassen eine neue Diagnose zu stellen waere der, dass insgesamt mehrere Fehler aufgetreten sind, die dann auf diesem Wege diagnostiziert werden koennen. Im allgemeinen ist dies zwar nicht der Fall, das Programm hat aber grundsaetzlich die Moeglichkeit, dieses Problem zu bewaeltigen, falls es doch einmal auftreten sollte.


micro-UNIXPERT START
Uebersichtsfragen zur Verdachts-generierung
Vorauswahl mit Zuordnung eines Diagnosebereiches
Diagnosebereich zuzuordnen ?
ja
nein
Durchsuche Diagnosebereich
micro-UNIXPERT ENDE
Kann eine Diagnose gestellt werden ?
ja
nein
Melde Diagnose
Weitere Diagnose stellen ?
ja
nein

2.3 Effizienz und Akzeptanz

Bei vielen wissensbasierten Systemen ist das Durchsuchen eines grossen Suchraums erforderlich, was sehr ineffizient sein kann. Die Komplexitaet kann aber verringert werden, wenn der Suchraum a priori eingeschraenkt werden kann und so nur noch ein Teil der Regelwissensbasis durchsucht werden muss. In micro-UNIXPERT wurde dies durch eine Einteilung der Regelwissensbasis in Diagnosebereiche erreicht, die mit Hilfe einer Vorauswahl ausgewaehlt werden koennen.
Fuer ein interaktives System wie micro-UNIXPERT ergeben sich zwei wichtige Aspekte fuer die Akzeptanz des Systems durch den Benutzer: Die (maximale) Antwortzeit des Programmes und die Anzahl der Fragen, die das Progamm dem Benutzer stellt. Die Antwortzeit des Programmes ist hierbei die Zeit zwischen zwei Interaktionen mit dem Benutzer, also z.B. die Zeit zwischen zwei Fragen oder zwischen einer Frage und dem Stellen der Diagnose. Fuer den Benutzer ist es nun wichtig, dass diese Antwortzeiten moeglichst kurz sind und die Anzahl der Fragen moeglichst gering ist. Soll die Anzahl der Fragen verringert werden, muss das Progamm Aufgaben zur Informationsakquisition uebernehmen, die vorher der Benutzer durchgefuehrt hat. Im Falle micro-UNIXPERT muss das Programm auf UNIX-Betriebssystemebene "absteigen", um bestimmte Kommandos auszufuehren, etwa um Warteschlangen oder bestimmte Dateien einzusehen, deren Inhalt zu analysieren und daraus Information abzuleiten. Es liegt zwar in der Hand des Programmierers eine Analyse (z.B. des Inhalts der Druckerwarteschlange) effizient zu programmieren, aber es wird auch viel Zeit benoetigt, um ueberhaupt an die Information heranzukommen. Aus diesen Gruenden steigt zwangslaeufig die Zeit, in der der Benutzer auf eine Aktion des Programmes warten muss und somit auch die (maximale) Antwortzeit. Es ist also wichtig, einen guten Kompromiss fuer dieses Problem zu finden.

2.4 Implementierung

2.4.1 Faktizitaeten

In Kapitel 2.2 wurde schon darauf hingewiesen, dass fuer die Wissensbasis von micro-UNIXPERT "negation by failure" nicht anwendbar ist und deshalb eine explizite Negation erforderlich wird. Dementsprechend muss auch die Basisdatenstruktur auf der operiert wird, organisiert sein.
Die Basisdatenstruktur in micro-UNIXPERT ist eine sog. "Faktizitaet", die besagt, ob eine "Aussage" gilt oder nicht gilt. Faktizitaeten werden in der Form 'gilt Aussage' (positive Faktizitaet) und 'giltnicht Aussage' (negative Faktizitaet) abgespeichert. "Aussagen" sind in den meisten Faellen Informationen ueber das Vorhandensein oder Nicht-Vorhandensein

von Symptomen.

Beispiel

> Die Faktizitaet 'gilt Druckzuschwach' besagt:
> "Es ist bekannt, dass der Druck zu schwach ist"
> (d.h das Symptom "Druckzuschwach" liegt vor)
>
> Die Faktizitaet 'giltnicht fehlenTeilederDatei' besagt:
> "Es ist bekannt, dass keine Teile der Datei fehlen"
> (d.h. das Symptom "fehlenTeilederDatei" liegt nicht vor)

Es ist nun moeglich, explizit einen failure zu verursachen, wenn bekannt ist, dass eine Faktizitaet nicht mit der Praemisse einer Regel uebereinstimmt.
Mit Hilfe der Regeln
(ass (gilt _x) (not(giltnicht _x))) und
(ass (giltnicht _x) (not(gilt _x)))
ist es moeglich die Beziehung zwischen den Praedikaten 'gilt' und 'giltnicht' herzustellen. Diese Regeln werden von micro-UNIXPERT nicht benutzt. Sie stellen nur eine prinzipielle Moeglichkeit fuer spaetere Konsistenztests dar. Es muss allerdings darauf geachtet werden, dass durch das Hinzufuegen der Regeln kein Zykel entsteht.
Das Programm hat in seinem Initialzustand bestimmte Informationen zur Verfuegung (z.B. 'gilt defaultprinterB48'), die ebenso wie die konsultationsspezifischen Faktizitaeten mit den Praedikaten 'gilt' und 'giltnicht' abgespeichert werden. Diese Anfangsinformation wird zur Ausfuehrung bestimmter Hilfsfunktionen benutzt und bleibt, im Gegensatz zu den konsultationsspezifischen Faktizitaeten, ueber die Lebensdauer des Programmlaufs erhalten.

2.4.2 Programmablauf

In diesem Abschnitt werden Regeln und Praedikate beschrieben, die fuer den Ablauf des Programms wesentlich sind. Allgemeine Praedikate sowie weitere Details werden in den nachfolgenden Kapiteln behandelt.

(unixpert0)

Top-level-Praedikat, wird durch das Programm Unixpert, das auf UNIX-Betriebssystemebene gestartet wird, aufgerufen und veranlasst den Start von micro-UNIXPERT.

(unixpert)

Top-level-Praedikat, durch dessen Aufruf das Programm auf LISPLOG-Ebene gestartet wird.

(unixpert listing)

Top-level-Praedikat durch dessen Aufruf das Programm auf LISPLOG-Ebene gestartet wird. Zudem wird zu Beginn und am Ende der Konsultation eine Liste der Faktizitaeten ausgegeben. Die Weiterverarbeitung erfolgt durch das Praedikat 'Beginn listing'.

(Beginn)

Startet die Einleitung des Programms, fuehrt die Feindiagnose und schliesslich eine Endebehandlung durch.

(Beginn listing)

Die Funktion ist im Praedikat 'unixpert listing' beschrieben.

(starte einleitung)

Fragt die beiden Hauptsymptome,
(0) Druck ist nicht gekommen
(1) Druck ist nicht zufriedenstellend , ab.
Die Weiterbehandlung erfolgt durch das Praedikat 'problem'.

Im Fall, dass der Druck nicht gekommen ist, werden folgende Informationen erfragt:
- Wohin wollte der Benutzer die Datei ausdrucken ?
  Zur Wahl stehen zwei Drucker. Die Weiterverarbeitung erfolgt durch das Praedikat 'ziel'.
- Wie ist der Wortlaut, des vom Benutzer durchgefuehrten Druckbefehls ?
  Die Weiterverarbeitung erfolgt durch das Praedikat 'analysiere'.
- Ist nach Ausfuehrung des Befehls das System-Prompt-Zeichen erschienen ?
  Die Kenntnis dieses Sachverhalts ist wesentlich fuer die Beantwortung der Frage, ob das UNIX-Betriebssystem eine Eingabe vom Terminal oder von einer Datei erwartet.
  Die Weiterverarbeitung erfolgt durch das Praedikat 'prompt'.

Fuer den Fall, dass der Druck nicht zufriedenstellend war, wird genauer auf den Grund der Unzufriedenheit eingegangen. Die Weiterverarbeitung erfolgt durch das Praedikat 'Alternative'.

(ziel _x)

Traegt die Information in die Wissensbasis ein, ob die Datei auf den Drucker in Bau 48 oder den Drucker in Bau 14 ausgedruckt werden sollte.

(prompt _x)

Traegt die Information ein, ob das Prompt-Zeichen erschienen ist oder nicht.

(Alternative _x)

Traegt die Information ein, welchen Grund der Unzufriedenheit der Benutzer angegeben hat. Ausserdem wird festgehalten, dass die anderen Moeglichkeiten nicht gegeben sind.

(analysiere _befehl)

Fuehrt eine Syntaxanalyse des eingegebenen Druckbefehls durch.
An dieser Stelle wird (n-solutions ... 1) verwendet, um zu gewaehrleisten, dass im Falle eines failures nicht ein zweites Mal eine Syntaxanalyse durchgefuehrt wird. In diesem speziellen Fall wird so ein allgemeiner (intermediaerer) Cut simuliert.

(optioncheck)

Steuert weitere Untersuchungen, indem festgestellt wird, ob eine falsche Option angegeben wurde.
Wenn im Befehl keine Option (aus einer Menge moeglicher Optionen) auftritt, keine falsche Option angegeben wurde und der Befehl auch sonst korrekt eingegeben wurde, wird die Voreinstellung des Druckers in der Datei .login getestet.

(syntaxanalyse _x)

Analysiert den eingegebenen Befehl.
Zuerst wird getestet, ob der Befehl durch lpr eingeleitet wird. Wenn dies nicht der Fall ist, handelt es sich um eine falsche Befehlseingabe. Mit Hilfe des Praedikats 'append' wird der Befehl in zwei Teile zerlegt, wobei der zweite Teil nach einer Option durchsucht wird, falls der erste Teil korrekt ist. Ohne die explizite Quotierung der Variablen _y kann es zu einem Fehler kommen. lp ist ein Druckbefehl, der vom LISP-System evaluiert wird. Ist _y an lp gebunden, wird dies evaluiert, und das Programm verursacht die Ausfuehrung eines Druckbefehls, anstatt eine Syntaxanalyse durchzufuehren. Die Weiterverarbeitung des

zweiten Teils erfolgt durch das Praedikat 'option'.

(option _z)

Verarbeitet den auf lpr folgenden Teil des Druckbefehls, indem dieser Teil auf das Vorhandensein einer -P-Option hin untersucht wird. Wird dabei eine -P-Option gefunden merkt sich das Programm alle relevanten Fakten dazu. Ausserdem wird auf besondere Fehlerfaelle eingegangen. Das Durchsuchen erfolgt durch das Praedikat 'member', indem die Liste _z1 (Rest des Befehls) nach einem korrekten Atom durchsucht wird, das -P als Prefix hat. Es ist erlaubt die -P-Option auseinanderzuziehen (z.B. -P lpb14). Wenn die Suche mit 'member' zu keinem Erfolg gefuehrt hat, wird mit Hilfe des Praedikats 'suche' nach dem Atom -P gesucht und das darauf folgende Atom in 'suboption' getestet. Zur Wahl stehen hier nur die beiden Moeglichkeiten lp und lpb14. Tritt keine der beiden Moeglichkeiten auf, handelt es sich um eine falsche Option. Mit Hilfe des Praedikats 'durchsuche' wird nun noch der Fehlerfall abgefangen, dass eine -P-Option zwar zusammenhaengend, aber fehlerhaft eingegeben wurde (z.B. -Plpb11), wobei bestimmte Fehlerfaelle ja schon vorher abgefangen wurden. Etwas Muehe macht der Umstand, dass es sich bei -P... um ein Atom handelt, das zum Durchsuchen zuerst in eine Liste umgewandelt werden muss. Ist keine fehlerhafte Option aufgetreten, so tritt die letzte 'option'-Regel inkraft, die als catchall fungiert. Mit Hilfe dieser Regel vermerkt das Progamm, dass weder eine Option aufgetreten ist, noch eine falsche Befehlseingabe vorgelegen hat.

(durchsuche _z1)

Durchsucht eine Liste sukzessive nach einem Atom, das durch -P eingeleitet wird.
Wird ein solches Atom gefunden, muss es sich um eine fehlerhafte Option handeln, denn alle korrekten Optionen wurden schon vorher abgefangen. Das Vorkommen solch eines fehlerhaften Atoms wird in 'teste' vermerkt, und die Durchsuchung des Befehls wird abgebrochen, indem ein failure geliefert wird.
In 'option' erfolgt der Aufruf (not(durchsuche _z1)). Im obigen Falle eines Fehlschlags wird deshalb als Wert 'true' geliefert, und das Ziel ist bewiesen. Wird die Liste aber bis zum Ende durchsucht (nil), ohne dass ein durch -P eingeleitetes Symbol gefunden wurde, ist 'durchsuche' zwar bewiesen, aufgrund der Negation wird aber ein failure geliefert. Daraus folgt, dass die catchall-Regel aufgerufen wird, die dann die oben in 'option' beschriebenen Aktionen ausfuehrt.
Die Verwendung der Faktizitaeten 'gilt gefunden' und 'giltnicht gefunden' ist etwas trickreich, dient aber dazu den allgemeinen Cut zu simulieren, der erforderlich ist, im failure-Fall das anschliessende Backtracking (verbunden mit Rekursionsaufloesung) zu kontrollieren und weitere Aufrufe von 'teste' zu vermeiden. Ausserdem wird verhindert, dass der Befehl weiter durchsucht

wird, wenn -P gefunden wurde.

(teste _y)

Die Funktion wurde in 'durchsuche' beschrieben. Zu erwaehnen ist, dass durch die LISP-Funktion 'explode' ein Atom in eine Liste umgewandelt wird. Diese Liste kann dann mit den Praedikaten 'kopf' und 'rest' weiterbehandelt werden.

(voreinstellung testen)

Fragt nach der Voreinstellung des Druckers in der Datei .login, allerdings nur, wenn die im Praedikat 'optioncheck' aufgefuehrten Bedingungen erfuellt sind. Ist eine Voreinstellung vorhanden, so wird dies vermerkt. Andernfalls wird vermerkt, dass der Druck zum Default-Printer umgeleitet wird.

(feindiagnose _diagnose)

Hauptsteuermechanismus fuer die Wissensdomaene, mit der in Kapitel 2.2 beschriebenen Funktionsweise.
Das Durchsuchen der Wissensdomaene erfolgt unter Ausnutzung des Backtracking-Mechanismus von LISPLOG . Zur Verdeutlichung dieses Sachverhalts wird in Anhang A u.a. ein "getraceter" Beispieldialog vorgefuehrt (siehe Seite 35).
Das Praedikat 'feindiagnose' startet die Vorauswahl fuer die Diagnosen, testet den in der Vorauswahl ausgewaehlten Diagnosebereich, und fuehrt eine Endebehandlung fuer eine Diagnose durch.

(vorauswahl _x)

Die Wissensdomaene ist in fuenf Unterbereiche, sogenannte Diagnosebereiche, eingeteilt, wobei im fuenften Diagnosebereich alle nicht diagnostizierbaren Faelle zusammengefasst werden. Wird der fuenfte Diagnosebereich aufgerufen wird vermerkt, dass nun keine weitere Vorauswahl mehr getroffen werden kann. Im folgenden werden die Kriterien fuer die Auswahl eines Diagnosebereiches beschrieben.
Ein Diagnosebereich wird mit Hilfe der Information, die in 'starte einleitung' akquiriert wurde, ausgewaehlt. Fuer die Auswahl des Diagnosebereiches 4 wird zudem Information benoetigt, die erst in Diagnosebereich 1 gesammelt wurde. Das bedeutet insbesondere, dass die Reihenfolge der Regeln der Vorauswahl nicht willkuerlich ist.
Diagnosebereich 0 wird durchsucht, falls eine falsche Befehlseingabe erfolgt ist. Wenn dies nicht der Fall ist und die Datei zudem nicht auf dem Drucker ausgedruckt wurde, den der Benutzer eigentlich ansprechen wollte, wird Diagnosebereich 1

durchsucht. Diagnosebereich 2 behandelt den speziellen Fall, dass kein System-Prompt erschienen ist und Diagnosebereich 3 deckt die Faelle ab, in denen der Druck nicht zufriedenstellend war. Diagnosebereich 4 behandelt alle uebrigen Faelle, fuer die gilt, dass die Datei nicht auf dem Drucker ausgedruckt wurde, den der Benutzer eigentlich als Zieldrucker angenommen hat. Zu dieser Bedingung kommen allerdings noch Praemissen hinzu, die in Diagnosebereich 1 akquiriert wurden. Diagnosebereich 5 hat die oben beschriebene Funktion.

(testediagnose diagnosebereich... _diagnose)

Diese Regeln stellen die eigentliche Wissensdomaene dar.
Ein Diagnosebereich besteht aus einer Menge solcher Regeln und wird zum Stellen einer Diagnose sukzessive durchsucht. Liefert eine Praemisse einer Regel einen failure, so wird die naechste Regel getestet. Durch die Verwendung von test und ntest sind die Regeln sehr einfach zu lesen. Beispielhaft soll hier eine Regel erlaeutert werden.

```
(ass (testediagnose diagnosebereich4 HWF1)
     (test dng)
     (ntest warning1)
     (ntest jobinWS1)
     (ntest warning2)
     (test jobinWS2)
     (test kommtraus2))
```

wird gelesen

```
   Wenn (der Druck nicht gekommen ist)
    und (es keine Warnung gab)
    und (der Auftrag nicht in der Warteschlange ist)
    und (es beim  zweiten Einsehen der Warteschlange wieder
         keine Warnung gab)
    und (der Auftrag jetzt in der Warteschlange ist)
    und (der Auftrag die Warteschlange wieder verlaesst)
   dann (kann die Diagnose aufgestellt werden, dass es sich
         um HWF1 handelt)
```

Im Programm werden also Kurzbegriffe verwendet, die in der Benutzerschnittstelle in natuerlichsprachliche Ausgabe transformiert werden. Der Experte wendet bestimmte Strategien an, die zeitlich aufeinanderfolgende Aktionen ausloesen. Es ist wichtig diese zeitliche Abfolge von Ereignissen in den Griff zu bekommen. Treten zwei Praemissen in einer Regel auf, die Ereignisse in diesem Sinn darstellen, so bestimmt die zeitliche Aufeinanderfolge der Ereignisse, in welcher Reihenfolge die Praemissen innerhalb der Regel vorkommen. Die Implementation

dieses Tatbestandes ist in LISPLOG sehr einfach moeglich, da die Praemissen einer Regel vom Regelinterpreter in einer festen Reihenfolge, naemlich von links nach rechts, abgearbeitet werden. Gleiche Aktionen, die sich nur durch die zeitliche Aufeinanderfolge unterscheiden, werden durch Numerierung gekennzeichnet (vgl. warning1, warning2).
Einen Sonderfall stellen die Regeln

```
(ass (testediagnose diagnosebereich4 warten)
     (test dng)
          .
          .
          .
      (ntest anersterStelle1))

(ass (testediagnose diagnosebereich4 warten)
     (test dng)
          .
          .
          .
     (ntest anersterStelle2))
```

dar.

Wenn festgestellt wird, dass der Druckauftrag zwar in die Warteschlange eingetragen wurde, aber nicht an erster Stelle der Warteschlange steht, kann zunaechst keine Diagnose gestellt werden. Dazu muss die Warteschlange laengere Zeit beobachtet werden. Da man diese Wartezeit keinem Benutzer zumuten kann, soll er an dieser Stelle micro-UNIXPERT verlassen und das System spaeter wieder aufrufen, wenn er weiss, wie sich die Warteschlange in der Zwischenzeit verhalten hat. Es ist wichtig, dass die erste der beiden oben aufgefuehrten Regeln vor den Regeln steht, die 'test kommtraus1' als Praemisse haben und die zweite Regel vor denjenigen, die 'test kommtraus2' als Praemisse haben. Diese Festlegung bedeutet zwar, dass zusaetzliches Metawissen in die Wissensbasis gesteckt wird, schraenkt aber auch die Anzahl zusaetzlich erforderlicher Regeln ein.

```
(starte ok _diagnose)
```

Fuehrt eine Endebehandlung durch und sorgt dafuer, dass der Benutzer das System veranlassen kann, eine neue Diagnose zu stellen. Fuer die Faelle, dass aus den Angaben des Benutzers keine Diagnose gestellt werden kann (weil diese etwa widerspruechlich sind) oder, dass das Programm eine Diagnose gestellt hat und vom Benutzer veranlasst wird eine neue Diagnose zu stellen, obwohl keine mehr gestellt werden kann, werden adaequate Meldungen ausgegeben.

(Mi) i aus 1..7

Praedikat, das bei einer fehlerhaften Eingabe benutzt wird, eine Frage an den Benutzer zu wiederholen und/oder die Antwort wiederholt einzulesen.

2.4.3 Built-in-Praedikate und LISP-Funktionen

In diesem Kapitel werden, die im Programm benutzten Built-in-Praedikate und LISP-Funktionen beschrieben.

(n-solutions (klauselkopf) x)

Durch die Verwendung von 'n-solutions' kann verhindert werden, dass die Bindungen der Anfragevariablen gedruckt werden, wobei ausserdem eine Frage nach weiteren Loesungen unterbleibt. Es werden so viele Loesungen gesucht, wie durch die Zahl x angegeben wird. Im Programm wurde nur der Fall x=1 benoetigt. 'n-solutions' liefert dann beim Aufruf der Klausel '(klauselkopf)' genau eine Loesung oder nil d.h., dass auch durch einen neuen, durch Backtracking verursachten Aufruf, keine weitere Loesung mehr geliefert werden kann.

(progn arg1 ... argn t)

Evaluiert arg1 bis argn und liefert den Wert 'true' zurueck.
Die Verwendung von 'progn' ist vor allem bei der Ausgabe vorteilhaft, wenn oft (not(terpri)) aufgerufen werden muesste. Bei Verwendung von 'progn' genuegt (terpri), wodurch eine Geschwindigkeitssteigerung erzielt werden kann.

(terpri)

Zeilenvorschub, liefert als Wert 'nil' und verursacht somit einen failure. Da dies im allgemeinen nich erwuenscht ist, muss (not(terpri)) aufgerufen werden, oder (terpri) muss, wie oben beschrieben, in die progn-Konstruktion eingebettet werden.

(princ "text")

Druckt den spezifizierten String.

(read)

Liest ein Atom ein.

(lineread)

Liest eine ganze Zeile in Form einer Liste ein.

(is _y <S-expression>)

Evaluiert <S-expression> und bindet den Wert an _y.

(ass Konklusion [Praemisse1 ...])

Fasst die hinter 'ass' stehende Zeile als Klausel auf und fuegt sie in die Datenbasis ein. Das Praedikat 'ass' ist auch zum dynamischen Einfuegen von Faktizitaeten in die Wissensbasis zu verwenden (also waehrend der Konsultation).

(rex Konklusion [Praemisse1 ...])

Entfernt die spezifizierte Klausel aus der Datenbasis (auch dynamisch).

(abolish Praedikat1 [Praedikat2 ...])

Entfernt alle Klauseln mit dem (den) angegebenen Praedikatnamen aus der Datenbasis (auch dynamisch).

(explode)

Transformiert ein Atom in eine Liste seiner Zeichen.

2.4.4 Allgemeine Praedikate

Im folgenden werden einige Praedikate vorgestellt, die im Programm zu Hilfszwecken eingefuehrt wurden.

(kopf _liste _res)

Liefert den Kopf der Liste _liste in der Variablen _res, wenn die Liste ungleich nil ist, und sonst nil.

(rest _liste _res)

Liefert den Rest der spezifizierten Liste ohne den Kopf.

(suche _atom _liste _restliste)

Sucht _atom in _liste und liefert den Rest der Liste in _restliste, wenn _atom gefunden wird, ansonsten ist der Beweis des Ziels fehlgeschlagen.

(append _liste1 _liste2 _res)

Konkateniert zwei Listen und liefert die so entstandene Liste als Ergebnis. Das Praedikat 'append' kann aber auch dazu verwendet werden eine Liste in zwei Teile zu zerlegen, wie es in der Syntaxanalyse erforderlich ist.

(member _atom _liste)

Stellt fest, ob _atom Element der Liste _liste ist. Wenn ja, ist das Ziel bewiesen, wenn nein schlaegt der Beweis fehl.

(lpqTest)

Gibt die Warteschlange des Druckers aus, den der Benutzer ansprechen wollte. Das Praedikat 'lpqTest' kann nur aufgerufen werden, wenn der Zieldrucker des Benutzers und der tatsaechlich angesprochene Drucker uebereinstimmen. Hat der Benutzer im Druckbefehl eine -P Option angegeben, muss die Warteschlange auch mit dieser Option aufgerufen werden, da er den Drucker explizit angesteuert hat. Hat der Benutzer keine Option angegeben, so wird die Warteschlange des in der Voreinstellung spezifizierten Druckers ausgegeben. War kein Drucker voreingestellt, so wird die Warteschlange des Default-Printers ausgegeben.

(test _x)

Das Ziel ist bewiesen, falls in der Wissensbasis vermerkt ist, dass das Symptom _x aufgetreten ist, d.h. falls in der Wissensbasis die Faktizitaet 'gilt _x' gefunden wird. Der Beweis schlaegt fehl, falls in der Wissensbasis vermerkt ist, dass das Symptom _x nicht aufgetreten ist, d.h falls dort die Faktizitaet 'giltnicht _x' gefunden wird. Liegt keine Information ueber _x vor, wird also weder 'gilt _x' noch 'giltnicht _x' gefunden, so wird beim Benutzer nachgefragt. Die aus der Antwort des Benutzers extrahierte Information wird gespeichert und 'test' verhaelt sich gemaess eines der beiden oben aufgezeigten Faelle.

(ntest _x)

Verhaelt sich umgekehrt wie 'test'.
Das Ziel ist bewiesen, falls 'giltnicht _x' gefunden wird, der

Beweis schlaegt fehl falls, 'gilt _x' gefunden wird, ansonsten wird auch hier beim Benutzer nachgefragt.

2.4.5 Ein-/Ausgabe

Das Programm arbeitet intern mit Namen, die zwar aussagekraeftig, aber fuer eine Programmausgabe nicht geeignet sind. Um eine akzeptable Benutzeroberflaeche zu erhalten, wurde die Ausgabe ueber eine E/A-Schnittstelle durchgefuehrt. Fuer die Ausgabe stehen dabei drei Tabellen zur Verfuegung, die durch die Praedikate 'tabellediag' , 'tabelle' und 'Fehlermeldung' dargestellt werden. In den genannten Tabellen werden den internen Namen, natuerlichsprachliche Ausgaben zugeordnet. Waehrend das Praedikat 'tabelle' fuer die Ausgabe in 'test' und 'ntest' benoetigt wird, fuehrt das Praedikat 'tabellediag' die Diagnosemeldungen des Programms durch. Das Praedikat 'Fehlermeldung' ist fuer die Ausgabe von Fehlermeldungen bei, vom System nicht interpretierbaren Benutzereingaben zustaendig. Die Funktion von 'tabelle' ist aber nicht nur auf die reine Ausgabe beschraenkt, sondern hier werden auch Aktionen durchgefuehrt, die dem Benutzer helfen an die benoetigte Information heranzukommen. Ein Benutzer kann beliebige natuerlichsprachliche Eingaben machen. Das Programm agiert bei Eingaben, die es interpretieren kann und fragt bei Eingaben, die es nicht interpretieren kann, beim Benutzer nach.

2.4.6 Durchgriffe auf das Betriebssystem

Aus in Kapitel 1.2 aufgefuehrten Gruenden ist es noetig, von der Programmebene auf UNIX-Betriebssystemebene abzusteigen, um an die fuer das Stellen einer Diagnose wichtige Information heranzukommen. Stuende dafuer kein besonderer Mechanismus zur Verfuegung, muesste die Diagnose an einem zweiten Terminal durchgefuehrt werden. Im Programm werden zwei Mechanismen benutzt:

(exec UNIX-Kommando)

Wird als Praemisse einer Regel aufgerufen und fuehrt die gewuenschte Betriebssystemfunktion aus.

(shell)

Wird als Praemisse einer Regel aufgerufen, und es kommt zu einer Verzweigung ins Betriebssystem. Der Benutzer kann nun selbst beliebige Betriebssystemkommandos ausfuehren und durch druecken der Tastenkombination CTRL d ins Programm zurueckkehren.

3. Ausblick

Fuer micro-UNIXPERT sind noch zahlreiche Erweiterungen geplant, deren Implementierung den Rahmen der vorliegenden Projektarbeit gesprengt haette.
Das Programm benutzt zur Zeit den Editor "vi" als Hilfsmittel zur Wissensakquisition. Es ist leicht moeglich, fuer diesen Zweck einen anderen Editor zu verwenden (vgl. Kapitel 2.4.6). Ausserdem sind Aenderungen der augenblicklichen Druckerkonfiguration (Bau 14, Bau 48) sowie des Default-Printers sehr einfach durchzufuehren.
Als erste Erweiterung ist eine Erklaerungskomponente geplant, um die Selbsterklaerungsfaehigkeit von logikorientierten Programmen auszunutzen. Zwar ist es schon moeglich, Programme mit LISPLOG zu tracen und somit den Programmablauf deutlich werden zu lassen, wobei das Programm seine Vorgehensweise implizit "erklaert" (vgl. fuenftes Problem in Anhang A), aber zusaetzlich sollen sowohl fachspezifische Begriffe erklaert werden, als auch "WIE"- und "WARUM"-Fragen des Benutzers beantwortet werden koennen. Eine "WIE"-Frage erlaubt, dass, wenn das Programm eine Diagnose gestellt hat, der Benutzer das Programm fragen kann, wie es auf diese Diagnose gekommen ist. Eine "WARUM"-Frage ermoeglicht, dass, wenn das Programm nach dem Auftreten eines Symptoms fragt, der Benutzer nachhaken kann, warum es diese Frage gestellt hat.
Ein weiteres Ziel ist die Reduktion der Anzahl der Fragen, indem das Programm Funktionen uebernimmt, die sonst der Benutzer haette ausfuehren muessen. Eine naehere Eroerterung dieses Themas ist bereits in Kapitel 2.3 erfolgt.
Ferner kann es vorkommen, dass waehrend der Konsultation vom Benutzer Fakten eingegeben werden, die sich widersprechen und zu einer Inkonsistenz der Datenbasis fuehren wuerden. Es wird angestrebt, einen Konsistenztest fuer die Datenbasis einzufuehren, damit der Benutzer auf die Widerspruechlichkeit seiner Angaben hingewiesen, und so eine Inkonsistenz der Datenbasis weitestgehend vermieden wird.
Schliesslich ist eine Erweiterung der Wissensdomaene des Systems denkbar. Das neue System (UNIXPERT) wuerde sich dann nicht nur auf die Behandlung von Druckerstoerungen beschraenken, sondern zusaetzliche Problembereiche an der UNIX-Peripherie erfassen (z.B. Terminal-Fehlerdiagnose).

Literatur

[Boley & Kammermeier et al. 1985]
H. Boley, F. Kammermeier u. die LISPLOG-Gruppe: LISPLOG: Momentaufnahmen einer LISP/PROLOG-Vereinheitlichung.
Universitaet Kaiserslautern, MEMO SEKI-85-03

[Clocksin & Mellish 1981/84]
W.Clocksin, C.Mellish : Programming in PROLOG.
Springer-Verlag, Berlin Heidelberg New York, 1981.
Second Edition 1984

[Flannagan 1986]
T.Flannagan : The Consistency of Negation as Failure.
The Journal of Logic Programming,
Volume 3, Number 2, July 1986

[Gordon 1985]
Thomas F. Gordon : On the suitability of Modula II for artificial intelligence programming.
Angewandte Informatik 7/85

[Kowalski 1983]
Robert Kowalski : Logic Programming.
In Proc. IFIP, pp. 133-145, Amsterdam, North Holland

[Puppe 1983]
F. Puppe : Med1 - Ein heuristisches Diagnosesystem mit effizienter Kontrollstruktur.
Universitaet Kaiserslautern, MEMO SEKI-KL-83-04

[Reiter 1977]
Raymond Reiter : On closed world data bases.
In Journee d'Etudes: Logique et bases de donnee, Departement d'Etudes et de Recherche en Informatique, Preprints,
Toulouse, 16. - 18. November 1977

[Wetter 1983]
Thomas Wetter : Ein modallogisch beschriebenes Expertensystem, ausgefuehrt am Beispiel von Ohrenerkrankungen.
Universitaet Aachen, Dissertation 1983

Anhang A : Beispieldialog

```
(* Start des Lisp-Systems,              *)
(* ausgehend vom UNIX - Betriebssystem *)

%Lisp lisplog.1.o


Franz Lisp, Opus 38.89 mit Cmu-, Ktu- und
Flavors-Erweiterungen  08.02.1985

Patch Nr. 38.3 geladen
Patch Nr. 38.4 geladen


News sind mit der Funktion (news) zu erhalten
Stand : 00-00-0000
1.lisplog

(* Einlesen micro-UNIXPERT *)

*consult unixpert0
[load unixpert0]


(*          Erstes Problem           *)
(*  Benutzung des Editors 'VI' zur   *)
(*  Informationsakquisition          *)


(unixpert0)
*(unixpert)
       *********************************
          micro-UNIXPERT gestartet
       *********************************

Welches Problem haben Sie? (0/1)
(0) Der Druck ist nicht gekommen.
(1) Sie sind mit dem Druck nicht zufrieden.
> 1
Bitte naeher spezifizieren (0/1/2/3/4/5)
(0) Druck zu schwach.
(1) Einzelne Zeichen ausgesetzt.
(2) Datei abgeschnitten.
(3) Es fehlen zwischendurch ganze Teile der Datei.
(4) Schriftbild verzerrt.
```



```
(5) Absolut unsinnige Ausgabe.
> 2

Fehlt nur die letzte Zeile ihrer Datei?
(ja/nein)
> ja
Bitte geben Sie explizit den Dateinamen
der zu druckenden Datei an.
> testdatei

Nun wird der Editor 'vi' aufgerufen.
Bitte ueberpruefen Sie, ob Sie die letzte Zeile
Ihrer Datei mit einem carriage return abge-
schlossen haben? (ja/nein)

"testdatei" 4 lines, 118 characters
Dies ist eine Testdatei.
Fuer diesen Text gilt, dass die letzte Zeile nicht
mit einem carriage-return abgeschlossen wurde.
~
~
~
~
~
"testdatei" 4 lines, 118 characters
> nein

Es lag daran, dass das carriage return
in der letzten Zeile Ihrer Datei
gefehlt hat.


Soll eine neue Diagnose versucht werden?(ja/nein)
> ja

Mit Ihren Angaben kann keine weitere
Diagnose gestellt werden!
      micro-UNIXPERT normal beendet
nil


(*        Zweites Problem          *)
(* Benutzerfehler bezueglich der *)
(* -P Option                     *)


*(unixpert)
       *********************************
          micro-UNIXPERT gestartet
       *********************************

Welches Problem haben Sie? (0/1)
```

```
(0) Der Druck ist nicht gekommen.
(1) Sie sind mit dem Druck nicht zufrieden.
> 0

Wo sollte der Druck hin? (B48/B14)
> B48
Geben Sie bitte den Wortlaut des Befehls ein.
> lpr -Plpb48 testdatei
Ist das Prompt-Zeichen erschienen nachdem Sie die
Return-Taste gedrueckt haben ? (ja/nein)
> ja

Die Druckeroption fuer Bau 48 lautet:  -Plp .


Soll eine neue Diagnose versucht werden?(ja/nein)
> nein

      micro-UNIXPERT normal beendet
nil



(* Drittes Problem mit Auflistung der      *)
(* Faktizitaeten                           *)
(* Fehlerfall : Ausdrucken auf Drucker,    *)
(*              den der Benutzer nicht     *)
(*              ansprechen wollte          *)


*(unixpert listing)
(ass (gilt t))
(ass (gilt defaultprinterB48))
(ass (giltnicht nil))
(ass (giltnicht Ende))
(ass (giltnicht defaultprinterB14))
(ass (giltnicht gefunden))
(ass (giltnicht Regelgetestet))
(ass (giltnicht Diagnosegestellt))

       ********************************
          micro-UNIXPERT gestartet
       ********************************

Welches Problem haben Sie? (0/1)
(0) Der Druck ist nicht gekommen.
(1) Sie sind mit dem Druck nicht zufrieden.
> 0

Wo sollte der Druck hin? (B48/B14)
> B48
Geben Sie bitte den Wortlaut des Befehls ein.
> lpr testdatei
```



```
Ist das Prompt-Zeichen erschienen nachdem Sie die
Return-Taste gedrueckt haben ?(ja/nein)
> jein
Was heisst jein ?
Bitte geben Sie eine der aufgefuehrten
Alternativen an.
Ist das Prompt-Zeichen erschienen nachdem Sie die
Return-Taste gedrueckt haben ? (ja/nein)
> ja

Bitte testen Sie die Voreinstellung der Datei .login
und machen Sie eine der folgenden Eingaben:
   B48  , wenn PRINTER auf lp gesetzt ist
   B14  , wenn PRINTER auf lpb14 gesetzt ist
   keine, wenn keine der beiden Voreinstellungen
          aufgetreten ist.
Zurueck mit CTRL d
% vi .login

".login" 14 lines, 332 characters
stty dec
set noglob; eval `tset -IQ`
unset noglob
set path=(/usr/users/lisplog
/usr/users/lisplog/tools /etc /usr/ucb /bin /usr/bi
n /usr/ktu /usr/new /usr/users/zoellner/kl-tuniks .)
set history=100
uptime
alias j jobs
alias ls ls -F
alias rm 'rm -i *'
setenv PRINTER lpb14
setenv EXINIT 'set ai lisp'
set ignoreeof
prmail
~
~
~
~
~
~
~
".login" 14 lines, 332 characters
% ^D
> B14

Sie haben die Voreinstellung auf den Drucker in Bau 14,
erwarten den Ausdruck aber in Bau 48.
Daher muessen Sie die Option -Plp bei Ihrem Druckkommando
eingeben!


Soll eine neue Diagnose versucht werden?(ja/nein)
```

```
> nein
(ass (gilt t))
(ass (gilt defaultprinterB48))
(ass (gilt dng))
(ass (gilt b48))
(ass (gilt prompt))
(ass (gilt lautvoreinstellungB14))
(ass (gilt Regelgetestet))
(ass (gilt befehlok))
(ass (gilt Diagnosegestellt))
(ass (giltnicht nil))
(ass (giltnicht Ende))
(ass (giltnicht defaultprinterB14))
(ass (giltnicht gefunden))
(ass (giltnicht dnz))
(ass (giltnicht b14))
(ass (giltnicht option))
(ass (giltnicht falscheBefehlseingabe))
(ass (giltnicht lautvoreinstellungB48))
(ass (giltnicht keinevoreinstellung))
(ass (giltnicht voreinstellungok))


      micro-UNIXPERT normal beendet
nil


(*          Viertes Problem          *)
(* Erkennen eines Hardwarefehlers  *)


*(unixpert)
       ********************************
          micro-UNIXPERT gestartet
       ********************************

Welches Problem haben Sie? (0/1)
(0) Der Druck ist nicht gekommen.
(1) Sie sind mit dem Druck nicht zufrieden.
> 0

Wo sollte der Druck hin? (B48/B14)
> B48
Geben Sie bitte den Wortlaut des Befehls ein.
> lpr -Plp testdatei
Ist das Prompt-Zeichen erschienen nachdem Sie die
Return-Taste gedrueckt haben ? (ja/nein)
> ja

---------------------------------------
no entries
```



```
---------------------------------------
Bitte sehen Sie in der Warteschlange nach,
ob es ein Warning gab.  (ja/nein)

> nein

Ist Ihr Auftrag in der Warteschlange?
(ja/nein)
> nein

Bitte geben Sie einen der folgenden Befehle ein:
  (1)    lpr -Plp;lpq -Plp Druck sollte nach Bau 48
  (2)    lpr -Plpb14;lpq -Plpb14 Druck sollte nach Bau 14
Druecken Sie anschliessend die return-Taste,
machen Sie eine Eingabe von etwa einer halben
Zeile Laenge und druecken Sie dann CTRL d.
Um in das Programm zurueckzukehren, bitte wieder
CTRL d druecken.
% lpr -Plp;lpq -Plp
Dies ist eine Testeingabe                (* Eingabe *)

lp is ready and printing
Rank   Owner      Job  Files                    Total Size
active lessel     716  (standart input)         26  bytes

% ^D
Gab es jetzt ein Warning ? (ja/nein)
> nein

Ist Ihr Auftrag in der Warteschlange? (ja/nein)
> ja

Befindet sich Ihr Druckauftrag an der
obersten Stelle der Warteschlange ? (ja/nein)
> ja

---------------------------------------
no entries

---------------------------------------
Ist Ihr Job mittlerweile aus der Warteschlange
verschwunden? (ja/nein)

> ja

Bitte ueberpruefen Sie ob der Drucker einge-
schaltet ist, und ob alle Verbindungen bestehen.


Soll eine neue Diagnose versucht werden?(ja/nein)
> ja
```

```
Wenn der Drucker eingeschaltet ist und alle
Verbindungen bestehen, liegt es an der Hard-
ware des Druckers.
Rufen Sie bitte den zustaendigen Betreuer an !


Soll eine neue Diagnose versucht werden?(ja/nein)
> ja


Ist der Ausdruck Ihrer Probedatei gekommen? (ja/nein)
> nein

Wenn der Drucker eingeschaltet ist und alle Verbindungen
bestehen, liegt es an der Hardware des Druckers.
Rufen Sie bitte den zustaendigen Betreuer an !


Soll eine neue Diagnose versucht werden? (ja/nein)
> ja

Mit Ihren Angaben kann keine weitere Diagnose gestellt werden!
      micro-UNIXPERT normal beendet
nil



(* Das fuenfte Problem (leichte  Abwandlung     *)
(* des vierten Problems) ist eine getracete     *)
(* Konsultation zum Aufzeigen des               *)
(* Backtracking-Mechanismus.                    *)
(* Es werden nur die Praedikate ausgegeben,     *)
(* die fuer die Abarbeitung der Wissensdomaene *)
(* wichtig sind.                                *)


*spy starte feindiagnose vorauswahl testediagnose gilt giltnicht
     test ntest
(starte feindiagnose vorauswahl testediagnose gilt giltnicht
     test ntest)
*(unixpert)

 CALL : (starte einleitung)
       ********************************
          micro-UNIXPERT gestartet
       ********************************

Welches Problem haben Sie? (0/1)
(0) Der Druck ist nicht gekommen.
(1) Sie sind mit dem Druck nicht zufrieden.
```



```
> 0

Wo sollte der Druck hin? (B48/B14)
> B14
Geben Sie bitte den Wortlaut des Befehls ein.
> lpr -Plpb14 druckdatei
Ist das Prompt-Zeichen erschienen nachdem Sie die
Return-Taste gedrueckt haben ? (ja/nein)
> ja

 EXIT : (starte einleitung)
 CALL : (feindiagnose _diagnose-1)
  CALL : (vorauswahl _x-12)
   CALL : (gilt falscheBefehlseingabe)
   FAIL : (gilt falscheBefehlseingabe)
  REDO : (vorauswahl _x-12)
   CALL : (giltnicht falscheBefehlseingabe)
   EXIT : (giltnicht falscheBefehlseingabe)
   CALL : (gilt dng)
   EXIT : (gilt dng)
  EXIT : (vorauswahl diagnosebereich1)
  CALL : (testediagnose diagnosebereich1 _diagnose-1)
   CALL : (gilt b14)
   EXIT : (gilt b14)
   CALL : (gilt lautoptionB48)
   FAIL : (gilt lautoptionB48)
   REDO : (gilt b14)
   FAIL : (gilt b14)
  REDO : (testediagnose diagnosebereich1 _diagnose-1)
   CALL : (gilt b48)
   FAIL : (gilt b48)
  REDO : (testediagnose diagnosebereich1 _diagnose-1)
   CALL : (giltnicht option)
   FAIL : (giltnicht option)
  REDO : (testediagnose diagnosebereich1 _diagnose-1)
   CALL : (giltnicht option)
   FAIL : (giltnicht option)
  REDO : (testediagnose diagnosebereich1 _diagnose-1)
   CALL : (gilt b14)
   EXIT : (gilt b14)
   CALL : (giltnicht option)
   FAIL : (giltnicht option)
   REDO : (gilt b14)
   FAIL : (gilt b14)
  REDO : (testediagnose diagnosebereich1 _diagnose-1)
   CALL : (gilt b48)
   FAIL : (gilt b48)
  REDO : (testediagnose diagnosebereich1 _diagnose-1)
   CALL : (giltnicht Regelgetestet)
   EXIT : (giltnicht Regelgetestet)
  REDO : (giltnicht Regelgetestet)
  FAIL : (giltnicht Regelgetestet)
   REDO : (testediagnose diagnosebereich1 _diagnose-1)
```

```
   FAIL : (testediagnose diagnosebereich1 _diagnose-1)
  REDO : (gilt dng)
  FAIL : (gilt dng)
   REDO : (vorauswahl _x-12)
    CALL : (giltnicht falscheBefehlseingabe)
    EXIT : (giltnicht falscheBefehlseingabe)
    CALL : (giltnicht prompt).
    FAIL : (giltnicht prompt)
    FAIL : (giltnicht falscheBefehlseingabe)
   REDO : (vorauswahl _x-12)
    CALL : (gilt dnz)
    FAIL : (gilt dnz)
   REDO : (vorauswahl _x-12)
    CALL : (giltnicht falscheBefehlseingabe)
    EXIT : (giltnicht falscheBefehlseingabe)
    CALL : (gilt prompt)
    EXIT : (gilt prompt)
    CALL : (gilt dng)
    EXIT : (gilt dng)
    CALL : (gilt befehlok)
    EXIT : (gilt befehlok)
    CALL : (gilt voreinstellungok)
    EXIT : (gilt voreinstellungok)
   EXIT : (vorauswahl diagnosebereich4)
   CALL : (testediagnose diagnosebereich4 _diagnose-1)
    CALL : (test dng)
     CALL : (gilt dng)
     EXIT : (gilt dng)
    EXIT : (test dng)
    CALL : (test warning1)
     CALL : (gilt warning1)
     FAIL : (gilt warning1)
    REDO : (test warning1)
 CALL : (gilt warning1)
 FAIL : (gilt warning1)
 CALL : (giltnicht warning1)
 FAIL : (giltnicht warning1)
-------------------------------------


(* Im Praedikat 'lpqTest' aufgerufene Faktizitaeten *)

 CALL : (gilt lautoptionB48)
 FAIL : (gilt lautoptionB48)
 CALL : (gilt lautvoreinstellungB48)
 FAIL : (gilt lautvoreinstellungB48)
 CALL : (gilt lautoptionB14)

no entries                    (* Warteschlange *)

 EXIT : (gilt lautoptionB14)
-------------------------------------
Bitte sehen Sie in der Warteschlange nach,
```



```
ob es ein Warning gab.  (ja/nein)

> nein

    REDO : (gilt dng)
    FAIL : (gilt dng)
    FAIL : (test warning1)
     REDO : (test dng)
 CALL : (gilt dng)
 EXIT : (gilt dng)
     REDO : (testediagnose diagnosebereich4 _diagnose-1)
      CALL : (test dng)
       CALL : (gilt dng)
       EXIT : (gilt dng)
      EXIT : (test dng)
      CALL : (ntest warning1)
       CALL : (giltnicht warning1)
       EXIT : (giltnicht warning1)
      EXIT : (ntest warning1)
      CALL : (test jobinWS1)
       CALL : (gilt jobinWS1)
       FAIL : (gilt jobinWS1)
      REDO : (test jobinWS1)
 CALL : (gilt jobinWS1)
 FAIL : (gilt jobinWS1)
 CALL : (giltnicht jobinWS1)
 FAIL : (giltnicht jobinWS1)
Ist Ihr Auftrag in der Warteschlange? (ja/nein)
> nein

      REDO : (ntest warning1)
 CALL : (giltnicht warning1)
 EXIT : (giltnicht warning1)
      REDO : (gilt dng)
      FAIL : (gilt dng)
      FAIL : (ntest warning1)
      FAIL : (test jobinWS1)
       FAIL : (giltnicht warning1)
      REDO : (test dng)
 CALL : (gilt dng)
 EXIT : (gilt dng)
      REDO : (testediagnose diagnosebereich4 _diagnose-1)
       CALL : (test dng)
        CALL : (gilt dng)
        EXIT : (gilt dng)
       EXIT : (test dng)
       CALL : (ntest warning1)
        CALL : (giltnicht warning1)
        EXIT : (giltnicht warning1)
       EXIT : (ntest warning1)
       CALL : (test jobinWS1)
        CALL : (gilt jobinWS1)
        FAIL : (gilt jobinWS1)
```

```
      REDO : (test jobinWS1)
CALL : (gilt jobinWS1)
FAIL : (gilt jobinWS1)
CALL : (giltnicht jobinWS1)
EXIT : (giltnicht jobinWS1)
      REDO : (giltnicht warning1)
      FAIL : (giltnicht warning1)
      FAIL : (test jobinWS1)
       REDO : (ntest warning1)
CALL : (giltnicht warning1)
EXIT : (giltnicht warning1)
       REDO : (gilt dng)
       FAIL : (gilt dng)
       FAIL : (ntest warning1)
      REDO : (test dng)
CALL : (gilt dng)
EXIT : (gilt dng)
      REDO : (testediagnose diagnosebereich4 _diagnose-1)
       CALL : (test dng)
        CALL : (gilt dng)
        EXIT : (gilt dng)
       EXIT : (test dng)
       CALL : (ntest warning1)
        CALL : (giltnicht warning1)
        EXIT : (giltnicht warning1)
       EXIT : (ntest warning1)
       CALL : (test jobinWS1)
        CALL : (gilt jobinWS1)
        FAIL : (gilt jobinWS1)
       REDO : (test jobinWS1)
CALL : (gilt jobinWS1)
FAIL : (gilt jobinWS1)
CALL : (giltnicht jobinWS1)
EXIT : (giltnicht jobinWS1)
       REDO : (giltnicht warning1)
       FAIL : (giltnicht warning1)
       FAIL : (test jobinWS1)
        REDO : (ntest warning1)
CALL : (giltnicht warning1)
EXIT : (giltnicht warning1)
        REDO : (gilt dng)
        FAIL : (gilt dng)
        FAIL : (ntest warning1)
       REDO : (test dng)
CALL : (gilt dng)
EXIT : (gilt dng)
      REDO : (testediagnose diagnosebereich4 _diagnose-1)
        CALL : (test dng)
         CALL : (gilt dng)
         EXIT : (gilt dng)
        EXIT : (test dng)
        CALL : (ntest warning1)
         CALL : (giltnicht warning1)
```



```
         EXIT : (giltnicht warning1)
        EXIT : (ntest warning1)
        CALL : (test jobinWS1)
         CALL : (gilt jobinWS1)
         FAIL : (gilt jobinWS1)
        REDO : (test jobinWS1)
CALL : (gilt jobinWS1)
FAIL : (gilt jobinWS1)
CALL : (giltnicht jobinWS1)
EXIT : (giltnicht jobinWS1)
        REDO : (giltnicht warning1)
        FAIL : (giltnicht warning1)
        FAIL : (test jobinWS1)
         REDO : (ntest warning1)
CALL : (giltnicht warning1)
EXIT : (giltnicht warning1)
         REDO : (gilt dng)
         FAIL : (gilt dng)
         FAIL : (ntest warning1)
        REDO : (test dng)
CALL : (gilt dng)
EXIT : (gilt dng)
      REDO : (testediagnose diagnosebereich4 _diagnose-1)
         CALL : (test dng)
          CALL : (gilt dng)
          EXIT : (gilt dng)
         EXIT : (test dng)
         CALL : (ntest warning1)
          CALL : (giltnicht warning1)
          EXIT : (giltnicht warning1)
         EXIT : (ntest warning1)
         CALL : (test jobinWS1)
          CALL : (gilt jobinWS1)
          FAIL : (gilt jobinWS1)
         REDO : (test jobinWS1)
CALL : (gilt jobinWS1)
FAIL : (gilt jobinWS1)
CALL : (giltnicht jobinWS1)
EXIT : (giltnicht jobinWS1)
         REDO : (giltnicht warning1)
         FAIL : (giltnicht warning1)
         FAIL : (test jobinWS1)
          REDO : (ntest warning1)
CALL : (giltnicht warning1)
EXIT : (giltnicht warning1)
          REDO : (gilt dng)
          FAIL : (gilt dng)
          FAIL : (ntest warning1)
         REDO : (test dng)
CALL : (gilt dng)
EXIT : (gilt dng)
       REDO : (testediagnose diagnosebereich4 _diagnose-1)
          CALL : (test dng)
```

```
            CALL : (gilt dng)
             EXIT : (gilt dng)
           EXIT : (test dng)
           CALL : (ntest warning1)
            CALL : (giltnicht warning1)
            EXIT : (giltnicht warning1)
           EXIT : (ntest warning1)
           CALL : (ntest jobinWS1)
            CALL : (giltnicht jobinWS1)
            EXIT : (giltnicht jobinWS1)
           EXIT : (ntest jobinWS1)
           CALL : (test warning2)
            CALL : (gilt warning2)
            FAIL : (gilt warning2)
           REDO : (test warning2)
 CALL : (gilt warning2)
 FAIL : (gilt warning2)
 CALL : (giltnicht warning2)
 FAIL : (giltnicht warning2)
Bitte geben Sie einen der folgenden Befehle ein:
  (1)    lpr -Plp;lpq -Plp        Druck sollte nach Bau 48
  (2)    lpr -Plpb14;lpq -Plpb14  Druck sollte nach Bau 14
Druecken Sie anschliessend die return-Taste,
machen Sie eine Eingabe von etwa einer halben
Zeile Laenge und druecken Sie dann CTRL d.
Um in das Programm zurueckzukehren, bitte wieder
CTRL d druecken.
% lpr -Plpb14;lpq -Plpb14
Dies ist eine Testeingabe
^D
Rank   Owner      Job  Files                      Total Size
1st    lessel     499  (standard input)           26 bytes
% ^D
Gab es jetzt ein Warning ? (ja/nein)
> nein

           REDO : (ntest jobinWS1)
 CALL : (giltnicht jobinWS1)
 EXIT : (giltnicht jobinWS1)
           REDO : (giltnicht warning1)
           FAIL : (giltnicht warning1)
           FAIL : (ntest jobinWS1)
           FAIL : (test warning2)
            FAIL : (giltnicht jobinWS1)
           REDO : (ntest warning1)
 CALL : (giltnicht warning1)
 EXIT : (giltnicht warning1)
           REDO : (gilt dng)
           FAIL : (gilt dng)
           FAIL : (ntest warning1)
            REDO : (test dng)
 CALL : (gilt dng)
 EXIT : (gilt dng)
```



```
      REDO : (testediagnose diagnosebereich4 _diagnose-1)
            CALL : (test dng)
             CALL : (gilt dng)
             EXIT : (gilt dng)
            EXIT : (test dng)
            CALL : (ntest warning1)
             CALL : (giltnicht warning1)
             EXIT : (giltnicht warning1)
            EXIT : (ntest warning1)
            CALL : (ntest jobinWS1)
             CALL : (giltnicht jobinWS1)
             EXIT : (giltnicht jobinWS1)
            EXIT : (ntest jobinWS1)
            CALL : (ntest warning2)
             CALL : (giltnicht warning2)
             EXIT : (giltnicht warning2)
            EXIT : (ntest warning2)
            CALL : (ntest jobinWS2)
             CALL : (giltnicht jobinWS2)
             FAIL : (giltnicht jobinWS2)
            REDO : (ntest jobinWS2)
 CALL : (giltnicht jobinWS2)
 FAIL : (giltnicht jobinWS2)
 CALL : (gilt jobinWS2)
 FAIL : (gilt jobinWS2)
Ist Ihr Auftrag in der Warteschlange? (ja/nein)
> ja

           REDO : (ntest warning2)
 CALL : (giltnicht warning2)
 EXIT : (giltnicht warning2)
           REDO : (giltnicht jobinWS1)
           FAIL : (giltnicht jobinWS1)
           FAIL : (ntest warning2)
           FAIL : (ntest jobinWS2)
            FAIL : (giltnicht warning2)
           REDO : (ntest jobinWS1)
 CALL : (giltnicht jobinWS1)
 EXIT : (giltnicht jobinWS1)
           REDO : (giltnicht warning1)
           FAIL : (giltnicht warning1)
           FAIL : (ntest jobinWS1)
            REDO : (ntest warning1)
 CALL : (giltnicht warning1)
 EXIT : (giltnicht warning1)
            REDO : (gilt dng)
            FAIL : (gilt dng)
            FAIL : (ntest warning1)
           REDO : (test dng)
 CALL : (gilt dng)
 EXIT : (gilt dng)
      REDO : (testediagnose diagnosebereich4 _diagnose-1)
            CALL : (test dng)
```

```
                CALL : (gilt dng)
                EXIT : (gilt dng)
               EXIT : (test dng)
               CALL : (ntest warning1)
                CALL : (giltnicht warning1)
                EXIT : (giltnicht warning1)
               EXIT : (ntest warning1)
               CALL : (ntest jobinWS1)
                CALL : (giltnicht jobinWS1)
                EXIT : (giltnicht jobinWS1)
               EXIT : (ntest jobinWS1)
               CALL : (ntest warning2)
                CALL : (giltnicht warning2)
                EXIT : (giltnicht warning2)
               EXIT : (ntest warning2)
               CALL : (test jobinWS2)
                CALL : (gilt jobinWS2)
                EXIT : (gilt jobinWS2)
               EXIT : (test jobinWS2)
               CALL : (ntest anersterStelle2)
                CALL : (giltnicht anersterStelle2)
                FAIL : (giltnicht anersterStelle2)
               REDO : (ntest anersterStelle2)
 CALL : (giltnicht anersterStelle2)
 FAIL : (giltnicht anersterStelle2)
 CALL : (gilt anersterStelle2)
 FAIL : (gilt anersterStelle2)
Befindet sich Ihr Druckauftrag an der
obersten Stelle der Warteschlange ? (ja/nein)

> ja

               REDO : (test jobinWS2)
 CALL : (gilt jobinWS2)
 EXIT : (gilt jobinWS2)
               REDO : (ntest warning2)
 CALL : (giltnicht warning2)
 EXIT : (giltnicht warning2)
              REDO : (giltnicht jobinWS1)
              FAIL : (giltnicht jobinWS1)
              FAIL : (ntest warning2)
              FAIL : (test jobinWS2)
              FAIL : (ntest anersterStelle2)
               FAIL : (gilt jobinWS2)
              REDO : (ntest jobinWS1)
 CALL : (giltnicht jobinWS1)
 EXIT : (giltnicht jobinWS1)
              REDO : (giltnicht warning1)
              FAIL : (giltnicht warning1)
              FAIL : (ntest jobinWS1)
               FAIL : (giltnicht warning2)
              REDO : (ntest warning1)
 CALL : (giltnicht warning1)
```



```
 EXIT : (giltnicht warning1)
              REDO : (gilt dng)
              FAIL : (gilt dng)
              FAIL : (ntest warning1)
               REDO : (test dng)
 CALL : (gilt dng)
 EXIT : (gilt dng)
       REDO : (testediagnose diagnosebereich4 _diagnose-1)
               CALL : (test dng)
                CALL : (gilt dng)
                EXIT : (gilt dng)
               EXIT : (test dng)
               CALL : (ntest warning1)
                CALL : (giltnicht warning1)
                EXIT : (giltnicht warning1)
               EXIT : (ntest warning1)
               CALL : (ntest jobinWS1)
                CALL : (giltnicht jobinWS1)
                EXIT : (giltnicht jobinWS1)
               EXIT : (ntest jobinWS1)
               CALL : (ntest warning2)
                CALL : (giltnicht warning2)
                EXIT : (giltnicht warning2)
               EXIT : (ntest warning2)
               CALL : (test jobinWS2)
                CALL : (gilt jobinWS2)
                EXIT : (gilt jobinWS2)
               EXIT : (test jobinWS2)
               CALL : (test kommtraus2)
                CALL : (gilt kommtraus2)
                FAIL : (gilt kommtraus2)
               REDO : (test kommtraus2)
 CALL : (gilt kommtraus2)
 FAIL : (gilt kommtraus2)
 CALL : (giltnicht kommtraus2)
 FAIL : (giltnicht kommtraus2)
-------------------------------------

(* Im Praedikat 'lpqTest' aufgerufene Faktizitaeten *)
 CALL : (gilt lautoptionB48)
 FAIL : (gilt lautoptionB48)
 CALL : (gilt lautvoreinstellungB48)
 FAIL : (gilt lautvoreinstellungB48)
 CALL : (gilt lautoptionB14)

 no entries              (* Warteschlange *)

 EXIT : (gilt lautoptionB14)
-------------------------------------
Ist Ihr Job mittlerweile aus der Warteschlange
verschwunden? (ja/nein)

> ja
```

```
                EXIT : (test kommtraus2)
               EXIT : (testediagnose diagnosebereich4 HWF1)
               CALL : (starte ok HWF1)
                CALL : (giltnicht Ende)
                EXIT : (giltnicht Ende)
Bitte ueberpruefen Sie ob der Drucker eingeschaltet ist
und ob alle Verbindungen bestehen.


Soll eine neue Diagnose versucht werden? (ja/nein)
> ja

               REDO : (giltnicht Ende)
               FAIL : (giltnicht Ende)
                REDO : (starte ok HWF1)
                 CALL : (gilt Ende)
                 FAIL : (gilt Ende)
                REDO : (starte ok HWF1)
                 CALL : (gilt Ende)
                 FAIL : (gilt Ende)
                REDO : (starte ok HWF1)
                FAIL : (starte ok HWF1)
               REDO : (gilt jobinWS2)
               FAIL : (gilt jobinWS2)
                FAIL : (test kommtraus2)
                 REDO : (test jobinWS2)
CALL : (gilt jobinWS2)
EXIT : (gilt jobinWS2)
                 REDO : (ntest warning2)
CALL : (giltnicht warning2)
EXIT : (giltnicht warning2)
                 REDO : (giltnicht jobinWS1)
                 FAIL : (giltnicht jobinWS1)
                 FAIL : (ntest warning2)
                 FAIL : (test jobinWS2)
                REDO : (ntest jobinWS1)
CALL : (giltnicht jobinWS1)
EXIT : (giltnicht jobinWS1)
               REDO : (giltnicht warning1)
               FAIL : (giltnicht warning1)
               FAIL : (ntest jobinWS1)
                FAIL : (giltnicht warning2)
               REDO : (ntest warning1)
CALL : (giltnicht warning1)
EXIT : (giltnicht warning1)
               REDO : (gilt dng)
               FAIL : (gilt dng)
               FAIL : (ntest warning1)
                REDO : (test dng)
CALL : (gilt dng)
EXIT : (gilt dng)
      REDO : (testediagnose diagnosebereich4 _diagnose-1)
                CALL : (test dng)
```



```
                 CALL : (gilt dng)
                 EXIT : (gilt dng)
               EXIT : (test dng)
               CALL : (ntest warning1)
                CALL : (giltnicht warning1)
                EXIT : (giltnicht warning1)
               EXIT : (ntest warning1)
               CALL : (ntest jobinWS1)
                CALL : (giltnicht jobinWS1)
                EXIT : (giltnicht jobinWS1)
               EXIT : (ntest jobinWS1)
               CALL : (ntest warning2)
                CALL : (giltnicht warning2)
                EXIT : (giltnicht warning2)
               EXIT : (ntest warning2)
               CALL : (test jobinWS2)
                CALL : (gilt jobinWS2)
                EXIT : (gilt jobinWS2)
               EXIT : (test jobinWS2)
               CALL : (test kommtraus2)
                CALL : (gilt kommtraus2)
                EXIT : (gilt kommtraus2)
               EXIT : (test kommtraus2)
              EXIT : (testediagnose diagnosebereich4 HWF3)
              CALL : (starte ok HWF3)
               CALL : (giltnicht Ende)
               EXIT : (giltnicht Ende)
Wenn der Drucker eingeschaltet ist und alle Verbindungen
bestehen, liegt es an der Hardware des Druckers.
Rufen Sie bitte den zustaendigen Betreuer an !


Soll eine neue Diagnose versucht werden? (ja/nein)
> nein

                EXIT : (starte ok HWF3)
              EXIT : (feindiagnose HWF3)
              CALL : (starte ende)
     micro-UNIXPERT normal beendet

               EXIT : (starte ende)
nil


(* Lisp-System verlassen *)

*(exit)

 Auf Wiedersehen !!!


(*  Wieder auf UNIX - Betriebssystemebene *)
```

Anhang B : Listing

```
;unixpert0 140486


(ass (unixpert0)
     (n-solutions (Beginn) 1) (exit))

(ass (unixpert)
     (n-solutions (Beginn) 1) nil)

(ass (unixpert listing)
     (n-solutions (Beginn listing) 1) nil)


(ass (Beginn)
     (starte einleitung)
     (feindiagnose _diagnose)
     (starte ende))

(ass (Beginn listing)
     (not(listing gilt giltnicht))
     (starte einleitung)
     (feindiagnose _diagnose)
     (not(listing gilt giltnicht))
     (starte ende))


(ass (starte einleitung)
     (progn
     (princ "       ********************************")
     (terpri)
     (princ "          micro-UNIXPERT gestartet")
     (terpri)
     (princ "       ********************************")
     (terpri) (terpri) t)  (M1))

(ass (M1)
     (progn
     (princ "Welches Problem haben Sie? (0/1)")
     (terpri)
     (princ "(0) Der Druck ist nicht gekommen.")
     (terpri)
     (princ "(1) Sie sind mit dem Druck nicht zufrieden.")
     (terpri) (princ "> ")
      t)
     (is _y (read))
     (problem _y))
```



```
(ass (problem 0)
     (ass (gilt dng))
     (ass (giltnicht dnz))
     (not(terpri))
     (M2) (M3 _befehl) (M4) (analysiere _befehl))

(ass (M2)
     (princ "Wo sollte der Druck hin? (B48/B14)")
     (not(terpri)) (princ "> ")
     (is _l (read)) (ziel _l))

(ass (M3 _befehl)
     (princ "Geben Sie bitte den Wortlaut des Befehls ein.")
     (not(terpri)) (princ "> ")
     (is _befehl (lineread t)))

(ass (M4)
     (princ "Ist das Prompt-Zeichen erschienen nachdem Sie die")
     (not(terpri))
     (princ "Return-Taste gedrueckt haben ? (ja/nein)")
     (not(terpri)) (princ "> ")
     (is _m (read)) (prompt _m))


(ass (problem 1)
     (ass (gilt dnz))
     (ass (giltnicht dng))
     (M5))

(ass (M5)
     (progn
     (princ "Bitte naeher spezifizieren (0/1/2/3/4/5)")
     (terpri)
     (princ "(0) Druck zu schwach.")
     (terpri)
     (princ "(1) Einzelne Zeichen ausgesetzt.")
     (terpri)
     (princ "(2) Datei abgeschnitten.")
     (terpri)
     (princ "(3) Es fehlen zwischendurch ganze")
     (princ " Teile der Datei.")
     (terpri)
     (princ "(4) Schriftbild verzerrt.")
     (terpri)
     (princ "(5) Absolut unsinnige Ausgabe.")
     (terpri) (princ "> ")
      t)
     (is _z (read)) (Alternative _z))


(ass (problem _sonst)
     (Fehlermeldung _sonst) (M1))
```

```
(ass (ziel B48)
     (ass (gilt b48))
     (ass (giltnicht b14)))
(ass (ziel B14)
     (ass (gilt b14))
     (ass (giltnicht b48)))
(ass (ziel _sonst)
     (Fehlermeldung _sonst) (M2))

(ass (prompt ja)
     (ass (gilt prompt)))
(ass (prompt nein)
     (ass (giltnicht prompt)))
(ass (prompt _sonst)
     (Fehlermeldung _sonst) (M4))


(ass (Alternative 0)
     (ass (gilt Druckzuschwach))
     (ass (giltnicht Zeichenausgesetzt))
     (ass (giltnicht Dateiabgeschnitten))
     (ass (giltnicht fehlenTeilederDatei))
     (ass (giltnicht Schriftbildverzerrt))
     (ass (giltnicht unsinnigeAusgabe)))
(ass (Alternative 1)
     (ass (gilt Zeichenausgesetzt))
     (ass (giltnicht Druckzuschwach))
     (ass (giltnicht Dateiabgeschnitten))
     (ass (giltnicht fehlenTeilederDatei))
     (ass (giltnicht Schriftbildverzerrt))
     (ass (giltnicht unsinnigeAusgabe)))
(ass (Alternative 2)
     (ass (gilt Dateiabgeschnitten))
     (ass (giltnicht Druckzuschwach))
     (ass (giltnicht Zeichenausgesetzt))
     (ass (giltnicht fehlenTeilederDatei))
     (ass (giltnicht Schriftbildverzerrt))
     (ass (giltnicht unsinnigeAusgabe)))
(ass (Alternative 3)
     (ass (gilt fehlenTeilederDatei))
     (ass (giltnicht Druckzuschwach))
     (ass (giltnicht Zeichenausgesetzt))
     (ass (giltnicht Dateiabgeschnitten))
     (ass (giltnicht Schriftbildverzerrt))
     (ass (giltnicht unsinnigeAusgabe)))
(ass (Alternative 4)
     (ass (gilt Schriftbildverzerrt))
     (ass (giltnicht Druckzuschwach))
     (ass (giltnicht Zeichenausgesetzt))
     (ass (giltnicht Dateiabgeschnitten))
     (ass (giltnicht fehlenTeilederDatei))
     (ass (giltnicht unsinnigeAusgabe)))
(ass (Alternative 5)
```



```
     (ass (gilt unsinnigeAusgabe))
     (ass (giltnicht Druckzuschwach))
     (ass (giltnicht Zeichenausgesetzt))
     (ass (giltnicht Dateiabgeschnitten))
     (ass (giltnicht fehlenTeilederDatei))
     (ass (giltnicht Schriftbildverzerrt)))
(ass (Alternative _sonst)
     (Fehlermeldung _sonst) (M5))


(ass (analysiere _befehl)
     (n-solutions (syntaxanalyse _befehl) 1)
     (optioncheck)
     (giltnicht option)
     (giltnicht falscheBefehlseingabe)
     (voreinstellung testen))
(ass (analysiere _befehl)
     (gilt option))
(ass (analysiere _befehl)
     (gilt falscheBefehlseingabe))

(ass (optioncheck)
     (gilt gefunden)
     (ass (gilt falscheoption))
     (ass (gilt falscheBefehlseingabe)))
(ass (optioncheck))

(ass (syntaxanalyse _x)
     (append _y _z _x)
     (equal (quote _y) (lpr))
     (option _z))
(ass (syntaxanalyse _x)
     (ass (gilt falscheBefehlseingabe))
     (ass (giltnicht Formatierung))
     (ass (giltnicht optionzulang))
     (ass (giltnicht falscheoption))
     (ass (giltnicht option)))

(ass (option _z1)
     (member -Plpb14 _z1)
     (ass (gilt lautoptionB14))
     (ass (giltnicht lautoptionB48))
     (ass (gilt option))
     (ass (giltnicht falscheBefehlseingabe)))
(ass (option _z1)
     (member -Plp _z1)
     (ass (gilt lautoptionB48))
     (ass (giltnicht lautoptionB14))
     (ass (gilt option))
     (ass (giltnicht falscheBefehlseingabe)))
(ass (option _z1)
     (member -Plpb48 _z1)
     (ass (gilt falscheBefehlseingabe))
```

```
     (ass (gilt optionzulang)))
(ass (option _z1)
     (member -plpb14 _z1)
     (ass (giltnicht option))
     (ass (gilt falscheBefehlseingabe))
     (ass (gilt Formatierung)))
(ass (option _z1)
     (member -plp _z1)
     (ass (giltnicht option))
     (ass (gilt falscheBefehlseingabe))
     (ass (gilt Formatierung)))
(ass (option _z1)
     (suche -P _z1 _rest)
     (suboption _rest))
(ass (option _z1)
     (not(durchsuche _z1)))
(ass (option _z1)
     (ass (giltnicht option))
     (ass (giltnicht falscheBefehlseingabe)))

(ass (suboption _x1) (append _x2 _y2 _x1)
                     (equal (quote _x2) (lpb14))
                     (ass (gilt lautoptionB14))
                     (ass (giltnicht lautoptionB48))
                     (ass (gilt option))
                     (ass (giltnicht falscheBefehlseingabe)))
(ass (suboption _x1) (append _x2 _y2 _x1)
                     (equal (quote _x2) (quote (lp)))
                     (ass (gilt lautoptionB48))
                     (ass (giltnicht lautoptionB14))
                     (ass (gilt option))
                     (ass (giltnicht falscheBefehlseingabe)))
(ass (suboption _x1) (ass (gilt falscheBefehlseingabe))
                     (ass (gilt falscheoption)))

(ass (suche _x (_x . _restliste) _restliste))
(ass (suche _x (_anonym . _y) _restliste)
     (suche _x _y _restliste))

(ass (durchsuche nil))
(ass (durchsuche _x)
     (giltnicht gefunden)
     (kopf _x _at)
     (teste _at)
     (rest _x _at5)
     (giltnicht gefunden)
     (durchsuche _at5))

(ass (teste _y)
     (giltnicht gefunden)
     (is _hilf (explode _y))
     (kopf _hilf _at1)
     (equal (quote - ) (quote _at1))
```



```
     (rest _hilf _at2)
     (kopf _at2 _at3)
     (equal P  (quote _at3))
     (rex (giltnicht gefunden))
     (ass (gilt gefunden)))
(ass (teste _y))


(ass (kopf nil nil))
(ass (kopf (_x . _y) _x))

(ass (rest nil nil))
(ass (rest (_x . _y) _y))


(ass (voreinstellung testen)
     (progn
     (princ "Bitte testen Sie die Voreinstellung")
     (princ " der Datei .login")
     (terpri)
     (princ "und machen Sie eine der folgenden Eingaben:")
     (terpri)
     (princ "   B48  , wenn PRINTER auf lp gesetzt ist")
     (terpri)
     (princ "   B14  , wenn PRINTER auf lpb14 gesetzt ist")
     (terpri)
     (princ "   keine, wenn keine der beiden Voreinstellungen")
     (terpri)
     (princ "          aufgetreten ist.")
     (terpri)
     (princ "Zurueck mit CTRL d ")
     (terpri) t)
     (shell)
     (not(terpri))
     (princ "> ") (is _k (read))
     (wennvoreinstellung _k))

(ass (wennvoreinstellung B48)
     (ass (gilt lautvoreinstellungB48))
     (ass (giltnicht lautvoreinstellungB14))
     (ass (giltnicht keinevoreinstellung)))
(ass (wennvoreinstellung B14)
     (ass (gilt lautvoreinstellungB14))
     (ass (giltnicht lautvoreinstellungB48))
     (ass (giltnicht keinevoreinstellung)))
(ass (wennvoreinstellung keine)
     (ass (gilt keinevoreinstellung))
     (ass (giltnicht lautvoreinstellungB48))
     (ass (giltnicht lautvoreinstellungB14)))
(ass (wennvoreinstellung _sonst)
     (Fehlermeldung _sonst)
     (voreinstellung testen))
```

```
(ass (append nil _x _x))
(ass (append (_a . _b) _c (_a . _d)) (append _b _c _d))

(ass (member _x (_x . _anonym)))
(ass (member _x (_anonym . _y)) (member _x _y))


(ass (test _x)
     (gilt _x))
(ass (test _x)
     (not(gilt _x))
     (not(giltnicht _x))
     (n-solutions (tabelle _x) 1) (M6 _x))

(ass (M6 _x)
     (not(terpri)) (princ "> ") (is _y (read))
     (auswertung1 _y _x))

(ass (ntest _x)
     (giltnicht _x))
(ass (ntest _x)
     (not(giltnicht _x))
     (not(gilt _x))
     (n-solutions (tabelle _x) 1) (M7 _x))

(ass (M7 _x)
     (not(terpri)) (princ "> ") (is _y (read))
     (auswertung2 _y _x))

(ass !(auswertung1 nein _x)
      (ass (giltnicht _x)) nil)
(ass !(auswertung1  ja _x)
      (ass (gilt _x)))
(ass !(auswertung1 _sonst _x)
      (Fehlermeldung _sonst) (M6 _x))

(ass !(auswertung2 nein _x)
      (ass (giltnicht _x)))
(ass !(auswertung2  ja _x)
      (ass (gilt _x)) nil)
(ass !(auswertung2 _sonst _x)
      (Fehlermeldung _sonst) (M7 _x))


(ass (feindiagnose _diagnose)
     (vorauswahl _x)
     (testediagnose _x _diagnose)
     (starte ok _diagnose))


(ass (vorauswahl diagnosebereich0)
     (gilt falscheBefehlseingabe))
(ass (vorauswahl diagnosebereich1)
```



```
     (giltnicht falscheBefehlseingabe)
     (gilt dng))
(ass (vorauswahl diagnosebereich2)
     (giltnicht falscheBefehlseingabe)
     (giltnicht prompt)
     (gilt dng))
(ass (vorauswahl diagnosebereich3)
     (gilt dnz))
(ass (vorauswahl diagnosebereich4)
     (giltnicht falscheBefehlseingabe)
     (gilt prompt)
     (gilt dng)
     (gilt befehlok)
     (gilt voreinstellungok))
(ass (vorauswahl diagnosebereich5))


(ass (testediagnose diagnosebereich0 kleinpverwendet)
     (gilt Formatierung))
(ass (testediagnose diagnosebereich0 optionzulang)
     (gilt optionzulang))
(ass (testediagnose diagnosebereich0 falscheoption)
     (gilt falscheoption))
(ass (testediagnose diagnosebereich0 Befehlunkorrekteingegeben)
     (gilt falscheBefehlseingabe)
     (giltnicht Formatierung)
     (giltnicht optionzulang)
     (giltnicht falscheoption))
(ass (testediagnose diagnosebereich1 falscheSpezifikation1)
     (gilt b14)
     (gilt lautoptionB48)
     (rex (giltnicht Regelgetestet))
     (ass (gilt Regelgetestet))
     (ass (giltnicht befehlok))
     (ass (gilt voreinstellungok)))
(ass (testediagnose diagnosebereich1 falscheSpezifikation2)
     (gilt b48) (gilt lautoptionB14)
     (rex (giltnicht Regelgetestet))
     (ass (gilt Regelgetestet))
     (ass (giltnicht befehlok))
     (ass (gilt voreinstellungok)))
(ass (testediagnose diagnosebereich1 falscheVoreinstellung1)
     (giltnicht option)
     (gilt lautvoreinstellungB48)
     (gilt b14)
     (rex (giltnicht Regelgetestet))
     (ass (gilt Regelgetestet))
     (ass (gilt befehlok))
     (ass (giltnicht voreinstellungok)))
(ass (testediagnose diagnosebereich1 falscheVoreinstellung2)
     (giltnicht option)
     (gilt lautvoreinstellungB14)
     (gilt b48)
```

```
     (rex (giltnicht Regelgetestet))
     (ass (gilt Regelgetestet))
     (ass (gilt befehlok))
     (ass (giltnicht voreinstellungok)))
(ass (testediagnose diagnosebereich1 Defaultprinter1)
     (gilt b14)
     (giltnicht option)
     (gilt keinevoreinstellung)
     (gilt defaultprinterB48)
     (ass (gilt befehlok))
     (ass (giltnicht voreinstellungok))
     (rex (giltnicht Regelgetestet))
     (ass (gilt Regelgetestet)))
(ass (testediagnose diagnosebereich1 Defaultprinter2)
     (gilt b48)
     (giltnicht option)
     (gilt keinevoreinstellung)
     (gilt defaultprinterB14)
     (ass (gilt befehlok))
     (ass (giltnicht voreinstellungok))
     (rex (giltnicht Regelgetestet))
     (ass (gilt Regelgetestet)))

(ass (testediagnose diagnosebereich1 _sonst)
     (giltnicht Regelgetestet)
     (ass (gilt befehlok))
     (ass (gilt voreinstellungok))
     nil)
(ass (testediagnose diagnosebereich2 Dateinamevergessen)
     (Eingabe vom Terminal erwartet))
(ass (Eingabe vom Terminal erwartet)
     (test Meldungkorrekt)
     (ntest prompt))
(ass (Eingabe vom Terminal erwartet)
     (ntest Meldungkorrekt) NIL)
(ass (testediagnose diagnosebereich3 Farbband)
     (test Druckzuschwach))
(ass (testediagnose diagnosebereich3 Farbband)
     (test Zeichenausgesetzt))
(ass (testediagnose diagnosebereich3 fehltcrinletzterZeile)
     (test Dateiabgeschnitten)
     (test fehltnurletzteZeile)
     (ntest crinletzterZeile))
(ass (testediagnose diagnosebereich3 Dateizugross)
     (test Dateiabgeschnitten)
     (test fehltnurletzteZeile)
     (test crinletzterZeile))
(ass (testediagnose diagnosebereich3 Dateizugross)
     (test Dateiabgeschnitten)
     (ntest fehltnurletzteZeile))
(ass (testediagnose diagnosebereich3 spooler)
     (test fehlenTeilederDatei))
(ass (testediagnose diagnosebereich3 objectfile)
```



```
     (test unsinnigeAusgabe))
(ass (testediagnose diagnosebereich3 HW )
     (test Schriftbildverzerrt))
(ass (testediagnose diagnosebereich3 Papierstau)
     (test Schriftbildverzerrt))
(ass (testediagnose diagnosebereich4 spooler)
     (test dng)
     (test warning1))
(ass (testediagnose diagnosebereich4 warten)
     (test dng)
     (ntest warning1)
     (test jobinWS1)
     (ntest anersterStelle1))
(ass (testediagnose diagnosebereich4 HWF1)
     (test dng)
     (ntest warning1)
     (test jobinWS1)
     (test kommtraus1))
(ass (testediagnose diagnosebereich4 HWF3)
     (test dng)
     (ntest warning1)
     (test jobinWS1)
     (test kommtraus1))
(ass (testediagnose diagnosebereich4 HWF2)
     (test dng)
     (ntest warning1)
     (test jobinWS1)
     (ntest kommtraus1))
(ass (testediagnose diagnosebereich4 spooler1)
     (test dng)
     (ntest warning1)
     (test jobinWS1)
     (ntest kommtraus1))
(ass (testediagnose diagnosebereich4 spooler)
     (test dng)
     (ntest warning1)
     (ntest jobinWS1)
     (test warning2))
(ass (testediagnose diagnosebereich4 spooler)
     (test dng)
     (ntest warning1)
     (ntest jobinWS1)
     (ntest warning2)
     (ntest jobinWS2))
(ass (testediagnose diagnosebereich4 warten)
     (test dng)
     (ntest warning1)
     (ntest jobinWS1)
     (ntest warning2)
     (test jobinWS2)
     (ntest anersterStelle2))
(ass (testediagnose diagnosebereich4 HWF1)
     (test dng)
```

```
     (ntest warning1)
     (ntest jobinWS1)
     (ntest warning2)
     (test jobinWS2)
     (test kommtraus2))
(ass (testediagnose diagnosebereich4 HWF3)
     (test dng)
     (ntest warning1)
     (ntest jobinWS1)
     (ntest warning2)
     (test jobinWS2)
     (test kommtraus2))
(ass (testediagnose diagnosebereich4 BF)
     (test dng)
     (ntest warning1)
     (ntest jobinWS1)
     (ntest warning2)
     (test jobinWS2)
     (test kommtraus2)
     (test Druckkommtjetzt))
(ass (testediagnose diagnosebereich4 HWF3)
     (test dng)
     (ntest warning1)
     (ntest jobinWS1)
     (ntest warning2)
     (test jobinWS2)
     (test kommtraus2)
     (ntest Druckkommtjetzt))
(ass (testediagnose diagnosebereich5 _y)
     (rex (giltnicht Ende))
     (ass (gilt Ende)))


(ass (starte ok _diagnose)
     (giltnicht Ende)
     (not(terpri))
     (tabellediag _diagnose)
     (progn
     (terpri)
     (terpri)
     (terpri)
      t)
     (rex (giltnicht Diagnosegestellt))
     (ass (gilt Diagnosegestellt))
     (princ "Soll eine neue Diagnose versucht werden?")
     (princ "(ja/nein)")
     (not(terpri)) (princ "> ")
     (is _k (read)) (eq _k nein))
(ass (starte ok _diagnose)
     (gilt Ende)
     (giltnicht Diagnosegestellt)
     (not(terpri))
     (princ "Mit Ihren Angaben kann keine Diagnose")
```



```
     (princ " gestellt werden!"))
(ass (starte ok _diagnose)
     (gilt Ende)
     (gilt Diagnosegestellt)
     (not(terpri))
     (princ "Mit Ihren Angaben kann keine weitere Diagnose")
     (princ " gestellt werden!"))


(ass (starte ende)
     (not(terpri))
     (abolish gilt)
     (abolish giltnicht)
     (ass (gilt t))
     (ass (giltnicht nil))
     (ass (giltnicht Ende))
     (ass (gilt defaultprinterB48))
     (ass (giltnicht defaultprinterB14))
     (ass (giltnicht gefunden))
     (ass (giltnicht Regelgetestet))
     (ass (giltnicht Diagnosegestellt))
     (princ "      micro-UNIXPERT normal beendet")
     (not(terpri)))


(ass (Fehlermeldung _x)
     (princ "Was heisst ")
     (princ _x)
     (princ " ?") (not(terpri))
     (princ "Bitte geben Sie eine der aufgefuehrten")
     (not(terpri))
     (princ "Alternativen an.")
     (not(terpri)))


(ass (tabelle Meldungkorrekt)
     (progn
     (terpri)
     (princ "Bitte unterbrechen Sie die Befehlsausfuehrung")
     (terpri)
     (princ "mit CTRL z und geben Sie anschliessend bg ein.")
     (terpri)
     (princ "Erhalten Sie eine Meldung, die von")
     (terpri)
     (princ " tty eingeleitet wird? (ja/nein) ")
      t))
(ass (tabelle fehltnurletzteZeile)
     (not(terpri))
     (princ "Fehlt nur die letzte Zeile ihrer Datei?")
     (not(terpri))
     (princ "(ja/nein) "))
(ass (tabelle crinletzterZeile)
```

```
     (princ "Bitte geben Sie explizit den Dateinamen ")
     (not(terpri))
     (princ "der zu druckenden Datei an.")
     (not(terpri))
     (princ "> ")
     (is _k (read))
     (princ "Nun wird der Editor 'vi' aufgerufen.")
     (not(terpri))
     (princ "Bitte ueberpruefen Sie, ob Sie die letzte Zeile")
     (not(terpri))
     (princ "Ihrer Datei mit einem carriage return abge-")
     (not(terpri))
     (princ "schlossen haben? (ja/nein) ")
     (not(terpri))
     (n-solutions (time 1 10) 1)
     (exec vi _k)
     (not(terpri)))
(ass (tabelle warning1)
     (progn
     (terpri)
     (princ "-------------------------------------")
     (terpri)
     t)
     (lpqTest)
     (progn
     (terpri)
     (princ "-------------------------------------")
     (terpri)
     (princ "Bitte sehen Sie in der Warteschlange nach, ")
     (terpri)
     (princ "ob es ein Warning gab.  (ja/nein) ")
     (terpri)
     t))
(ass (tabelle jobinWS1)
     (not(terpri))
     (princ "Ist Ihr Auftrag in der Warteschlange?")
     (not(terpri))
     (princ "(ja/nein) "))
(ass (tabelle anersterStelle1)
     (not(terpri))
     (princ "Befindet sich Ihr Druckauftrag an der ")
     (not(terpri))
     (princ "obersten Stelle der Warteschlange ? (ja/nein)")
     (not(terpri)))
(ass (tabelle anersterStelle2)
     (not(terpri))
     (princ "Befindet sich Ihr Druckauftrag an der ")
     (not(terpri))
     (princ "obersten Stelle der Warteschlange ? (ja/nein)")
     (not(terpri)))
(ass (tabelle kommtraus1)
     (progn
     (terpri)
```



```
     (princ "-------------------------------------")
     (terpri)
     t)
     (lpqTest)
     (progn
     (terpri)
     (princ "-------------------------------------")
     (terpri)
     (princ "Ist Ihr Job mittlerweile aus der Warteschlange")
     (terpri)
     (princ "verschwunden? (ja/nein) ")
     (terpri)
     t))
(ass (tabelle warning2)
     (progn
     (terpri)
     (princ "Bitte geben Sie einen der folgenden Befehle ein:")
     (terpri)
     (princ "  (1)    lpr -Plp;lpq -Plp")
     (princ " Druck sollte nach Bau 48")
     (terpri)
     (princ "  (2)    lpr -Plpb14;lpq -Plpb14")
     (princ " Druck sollte nach Bau 14")
     (terpri)
     (princ "Druecken Sie anschliessend die return-Taste,")
     (terpri)
     (princ "machen Sie eine Eingabe von etwa einer halben")
     (terpri)
     (princ "Zeile Laenge und druecken Sie dann CTRL d. ")
     (terpri)
     (princ "Um in das Programm zurueckzukehren, bitte wieder")
     (terpri)
     (princ "CTRL d druecken.")
     (terpri)
      t)
     (shell)
     (not(terpri))
     (princ "Gab es jetzt ein Warning ? (ja/nein) "))
(ass (tabelle jobinWS2)
     (not(terpri))
     (princ "Ist Ihr Auftrag in der Warteschlange?(ja/nein)"))
(ass (tabelle kommtraus2)
     (not(terpri))
     (princ "-------------------------------------")
     (not(terpri))
     (lpqTest)
     (progn
     (terpri)
     (princ "-------------------------------------")
     (terpri)
     (princ "Ist Ihr Job mittlerweile aus der Warteschlange")
     (terpri)
     (princ "verschwunden? (ja/nein) ")
```

```
     (terpri)
      t))
(ass (tabelle Druckkommtjetzt)
     (not(terpri))
     (princ "Ist der Ausdruck Ihrer Probedatei gekommen?")
     (not(terpri))
     (princ "(ja/nein) "))

(ass (lpqTest)
     (gilt lautoptionB48)
     (exec lpq -Plp))
(ass (lpqTest)
     (gilt lautvoreinstellungB48)
     (exec lpq))
(ass (lpqTest)
     (gilt lautoptionB14)
     (exec lpq -Plpb14))
(ass (lpqTest)
     (gilt lautvoreinstellungB14)
     (exec lpq))
(ass (lpqTest)
     (gilt keinevoreinstellung)
     (exec lpq))

(ass (time _von _bis)
     (is _v (add _von 1))
     (neq _v _bis)
     (time _v _bis))
(ass (time _von _bis))


(ass (tabellediag Farbband)
     (princ "Wahrscheinlich ist das Farbband des Druckers")
     (not(terpri))
     (princ "verheddert oder zu Ende.")
     (not(terpri))
     (princ "Bitte rufen Sie den zustaendigen")
     (princ " Betreuer an! "))
(ass (tabellediag fehltcrinletzterZeile)
     (princ "Es lag daran, dass das carriage return")
     (not(terpri))
     (princ "in der letzten Zeile Ihrer Datei")
     (not(terpri))
     (princ "gefehlt hat."))
(ass (tabellediag Dateizugross)
     (princ "Wahrscheinlich ist Ihre Datei")
     (princ " zu gross! "))
(ass (tabellediag spooler)
     (princ "Wahrscheinlich liegt der Fehler")
     (princ " beim Spooler.")
     (not(terpri))
     (princ "Bitte rufen Sie den zustaendigen Betreuer an ")
     (not(terpri))
```



```
     (princ "oder benachrichtigen Sie das Rechenzentrum ! "))
(ass (tabellediag warten)
     (princ "Falls die Warteschlange laengere Zeit beobachtet")
     (not(terpri))
     (princ "wurde und der Druckauftrag nicht weiter nach")
     (not(terpri))
     (princ "oben gerueckt ist, bitte eine neue Diagnose")
     (not(terpri))
     (princ "stellen lassen."))
(ass (tabellediag HWF1)
     (princ "Bitte ueberpruefen Sie ob der Drucker einge-")
     (not(terpri))
     (princ "schaltet ist, und ob alle Verbindungen bestehen."
     (not(terpri)))
(ass (tabellediag HWF2)
     (princ "Ueberpruefen Sie bitte, ob der Drucker")
     (princ " auf ONLINE steht."))
(ass (tabellediag HWF3)
     (princ "Wenn der Drucker eingeschaltet ist und alle")
     (not(terpri))
     (princ "Verbindungen bestehen, liegt es an der Hard-")
     (not(terpri))
     (princ "ware des Druckers.")
     (not(terpri))
     (princ "Rufen Sie bitte den zustaendigen ")
     (princ "Betreuer an ! "))
(ass (tabellediag spooler1)
     (princ "Wenn der Drucker auf ONLINE steht, liegt es")
     (not(terpri))
     (princ "wahrscheinlich am spooler.")
     (not(terpri))
     (princ "Bitte rufen Sie den zustaendigen Betreuer an ")
     (not(terpri))
     (princ "oder benachrichtigen Sie das Rechenzentrum ! "))
(ass (tabellediag HW)
     (princ "Es liegt wahrscheinlich ein")
     (princ " Hardwarefehler vor.")
     (not(terpri))
     (princ "Rufen Sie bitte den zustaendigen")
     (princ " Betreuer an! "))
(ass (tabellediag BF)
     (princ "Wahrscheinlich haben Sie beim ")
     (princ "ersten Mal das Druckkommando ")
     (not(terpri))
     (princ "falsch eingegeben.")
     (not(terpri))
     (princ "Bitte versuchen Sie es noch")
     (princ " einmal! "))
(ass (tabellediag objectfile)
     (princ "Sie haben beim Druckkommando")
     (princ " wahrscheinlich eine Objekt-")
     (not(terpri))
     (princ "file als Dateinamen angegeben.")
```

```
     (not(terpri))
     (princ "Bitte geben sie erneut das")
     (princ " Druckkommando mit Ihrer ")
     (not(terpri))
     (princ "Quelldatei ein! "))
(ass (tabellediag Papierstau)
     (princ "Bitte ueberpruefen Sie, ob ein ")
     (princ "Papierstau am Drucker vorliegt.")
     (not(terpri))
     (princ "Rufen Sie bitte den zustaendigen ")
     (princ "Betreuer an! "))
(ass (tabellediag Dateinamevergessen)
     (princ "Wahrscheinlich haben Sie vergessen, im")
     (not(terpri))
     (princ "Druckkommando den Dateinamen anzugeben.")
     (not(terpri))
     (princ "Versuchen Sie das Druckkommando")
     (princ " bitte erneut. "))
(ass (tabellediag falscheVoreinstellung2)
     (princ "Sie haben die Voreinstellung")
     (princ " auf den Drucker in Bau 14,")
     (not(terpri))
     (princ "erwarten den Ausdruck aber in")
     (princ " Bau 48.")
     (not(terpri))
     (princ "Daher muessen Sie die Option")
     (princ " -Plp bei Ihrem Druckkommando")
     (not(terpri))
     (princ "eingeben! "))
(ass (tabellediag falscheVoreinstellung1)
     (princ "Sie haben die Voreinstellung")
     (princ " auf den Drucker in Bau 48,")
     (not(terpri))
     (princ "erwarten den Ausdruck aber in")
     (princ " Bau 14.")
     (not(terpri))
     (princ "Daher muessen Sie die Option ")
     (princ "-Plpb14 bei Ihrem ")
     (not(terpri))
     (princ "Druckkommando eingeben! "))
(ass (tabellediag falscheSpezifikation2)
     (princ "Sie haben in der Option des")
     (princ " Druckbefehls Bau 14 angegeben,")
     (not(terpri))
     (princ "erwarten den Ausdruck aber in")
     (princ " Bau 48.")
     (not(terpri))
     (princ "Geben Sie bitte die Option -Plp bei ")
     (not(terpri))
     (princ "Ihrem Druckbefehl mit ein!"))
(ass (tabellediag falscheSpezifikation1)
     (princ "Sie haben in der Option des")
     (princ " Druckbefehls Bau 48 angegeben,")
```



```
     (not(terpri))
     (princ "erwarten den Ausdruck aber in")
     (princ " Bau 14.")
     (not(terpri))
     (princ "Geben Sie deshalb bitte die Option -Plpb14")
     (not(terpri))
     (princ "bei Ihrem Druckbefehl mit ein!")
     (not(terpri)))
(ass (tabellediag Defaultprinter1)
     (progn
     (princ "Sie erwarten den Druck in Bau 14, haben aber")
     (terpri)
     (princ "weder im Befehl noch in der Voreinstellung")
     (terpri)
     (princ "eine -P-Option angegeben, sodass der Druck auf")
     (terpri)
     (princ "den Default-Printer in Bau 48 ausgedruckt wird.")
      t))
(ass (tabellediag Defaultprinter2)
     (progn
     (princ "Sie erwarten den Druck in Bau 48, haben aber")
     (terpri)
     (princ "weder im Befehl noch in der Voreinstellung")
     (terpri)
     (princ "eine -P-Option angegeben, sodass der Druck auf")
     (terpri)
     (princ "den Default-Printer in Bau 14 ausgedruckt wird.")
      t))
(ass (tabellediag Befehlunkorrekteingegeben)
     (princ "Sie haben den Befehl unkorrekt eingegeben.")
     (not(terpri))
     (princ "Der Druckbefehl lautet:")
     (not(terpri))
     (princ "lpr <dateiname> und")
     (princ " lpr -Plp <dateiname>,")
     (not(terpri))
     (princ "bzw. lpr -Plpb14 <dateiname> fuer den")
     (not(terpri))
     (princ "Ausdruck in Bau 48,bzw in Bau14."))
(ass (tabellediag falscheoption)
     (princ "Sie haben in Ihrem Druckbefehl")
     (princ " eine falsche Option eingegeben.")
     (not(terpri))
     (princ "Die Druckeroption lautet:")
     (not(terpri))
     (princ " -Plp fuer Bau 48 und -Plpb14 fuer")
     (princ " den Drucker in Bau 14! "))
(ass (tabellediag optionzulang)
     (princ "Die Druckeroption fuer Bau 48 lautet: ")
     (princ " -Plp ."))
(ass (tabellediag kleinpverwendet)
     (princ "Sie haben in der Option Ihres Druckbefehls")
     (not(terpri))
```

```
     (princ "ein kleines p statt einem grossen P verwendet")
     (not(terpri))
     (princ "Bitte versuchen Sie den Druckbefehl")
     (princ " erneut!"))


(ass (gilt t))
(ass (gilt defaultprinterB48))

(ass (giltnicht defaultprinterB14))
(ass (giltnicht nil))
(ass (giltnicht Ende))
(ass (giltnicht gefunden))
(ass (giltnicht Regelgetestet))
(ass (giltnicht Diagnosegestellt))
```