

Teaching Guide

Kleine Stolpersteine im Informatik-Studium und wie sie ausgeräumt werden könnten.

von Xuân Baldauf <teaching-guide--2004.04.22@studium.baldauf.org>

Vorwort

Ziel dieses Dokuments ist, vom Fachschaftsrat Informatik an der Universität Leipzig diskutiert, geändert und schließlich als „Empfehlung“ beschlossen zu werden. Danach soll es Dozenten bekannt gemacht und vom Fachschaftsrat unter Einbringung von interessierten Studenten regelmäßig gepflegt werden.

Dieses Dokument wurde erstellt in einer Prüfungssaison aus zwischenzeitlicher Demotivation nach 3 Prüfungen, für weitere 5 Prüfungen sofort weiter zu lernen und den Beobachtungen, die sich beim Lernen für Prüfungen so ergeben haben.

Dieses Dokument soll eine Empfehlung für Dozenten sein, um das Leben von Studenten zu erleichtern.

1 Vorlesungs-Materialien

1.1 Erstellen Sie ein Skript und stellen Sie dieses den Studenten zur Verfügung.

Viele Studenten verstehen nicht auf Anhieb, was vorgetragen wird. Und selbst wenn dies der Fall ist, können sie es auch zwischenzeitlich vergessen. Wenn Sie kein Skript erstellen wollen oder können, dann

1.1.1 Schreiben Sie skriptreif an die Tafel, wenn Sie kein Skript herausgeben.

In diesem Fall ist die Wahrscheinlichkeit nicht gering, dass sich ein Student erbarmt und selbst ein Skript schreibt.

- Gutes Beispiel: Peter Kunkel in der Vorlesung „Numerische Mathematik“ (2003 Wintersemester). Herr Kunkel fürchtet um seine Zuhörerschaft, wenn er sein Skript veröffentlichen würde. Aber schreibt so an die Tafel, wie es im Skript stehen würde.

Es kann sein, dass Sie auf einem bestehenden Lehrbuch aufbauen. In diesem Fall kann das Skript-Veröffentlichen entfallen:

1.1.2 Erklären Sie das Lehrbuch zum Skript, und zwar ausdrücklich und zu Beginn der Veranstaltungsreihe.

Das Lehrbuch sollte diese Eigenschaften haben:

- Das Lehrbuch sollte nicht übermäßig teuer sein.
- Das Lehrbuch sollte in der Bibliothek in ausreichender Anzahl vorhanden sein.

Wenn Sie statt einem Skript auf ein Lehrbuch verweisen, sollte das Lehrbuch aber auch wirklich ein Skript ersetzen können. Das heißt:

1.1.2.1 Ist das Lehrbuch umfangreicher als die Vorlesung, so grenzen Sie die Themen zu Beginn der Veranstaltungsreihe ein.

Sinn der Sache ist, dass der Student schon vorher entscheiden kann, wieviel er vom Lehrbuch können muss und ob er gegriffen das Lehrbuch kauft.

1.1.2.2 Ist die Vorlesung umfangreicher als das Lehrbuch, so erstellen Sie ein Skript für die fehlenden Themen.

In einigen Fällen ist das Lehrbuch, auf das der Dozent aufbaut, von ihm selbst geschrieben:

1.1.2.3 Ist das Lehrbuch von Ihnen selbst geschrieben, so überlegen Sie, ob es nicht in einer Online-Ausgabe zur Verfügung stellen.

Dies widerspricht normalerweise dem kommerziellen Interesse des Dozenten oder zumindest seines Verlags, da eine entgeltlose Online-Ausgabe des eigenen Buchs in Konkurrenz zur Druckversion steht. Jedoch ist in der Regel diese Konkurrenz marginal, da diese Studenten, die sich eine Online-Version antun, in der Regel aus finanziellen Gründen nicht das Buch kaufen können. Je nach ökonomischer Theorie fördert eine Online-Version sogar den Absatz des eigenen Buchs.

- Gutes Beispiel: Erhard Rahm in der Vorlesung „Mehrrechner-Datenbanksysteme“ (2003 Wintersemester).

1.2 Stellen Sie die in der Veranstaltung verwendeten Folien zur Verfügung.

Der Student hat damit besser die Möglichkeit, sich die Vorlesung noch einmal revue passieren zu lassen.

1.2.1 Stellen Sie die in der Veranstaltung verwendeten Folien vor der Veranstaltung zur Verfügung.

Dies ist wichtig besonders dann, wenn der Student einmal nicht sofort ein Teil-Thema verstanden hat. Dann kann er nämlich während der Vorlesung sich noch einmal die Folien durchlesen und doch noch eine Erleuchtung erfahren, sodass er für den weiteren Verlauf der Vorlesung nicht den Anschluss verloren hat.

1.2.1.1 Stellen Sie die in der Veranstaltung verwendeten Folien wenigstens nach der Veranstaltung zur Verfügung.

Der Student sollte wenigstens die Möglichkeit zum Lernen und zur

Prüfungsvorbereitung mit den Folien haben.

1.2.1.2 Stellen Sie die in der Veranstaltung verwendeten Folien wenigstens 21 Tage vor der Prüfung zur Verfügung.

Es soll vorgekommen sein, dass Dozenten ein Skript versprochen haben, dieses aber nicht rechtzeitig vor der Prüfung verfügbar oder fertig war. Der Wert eines solchen Skripts ist für den Studenten viel geringer, wenn er damit nicht lernen kann.

1.2.1.3 Nummerieren Sie alle verwendeten Folien.

Es ist sehr zweckmäßig, auf die verwendeten Folien ganz klein eine Folien-Nummer zu notieren. Dann lassen sich leichter Folien referenzieren, d.h. das Nachschlagen von Folien und das Diskutieren über ihren Inhalt wird dadurch sehr erleichtert. Sind Folien nummeriert, so lässt sich auch der Vorlesungsvortrag leichter verfolgen, wenn der Student die Folien bereits (ausgedruckt oder digital) vorliegen hat.

1.2.2 Wenn Sie die Folien zum Skript erklären, dann müssen die Folien skriptreif sein.

Dies ist nicht immer einfach, denn oft bestehen einzelne Stichpunkte auf Folien lediglich aus Wortgruppen, nicht aus ganzen Sätzen. Ersteres ist für die Präsentation sinnvoll, letzteres für das volle Verständnis beim Lernen für die Prüfung. Deswegen ist ein Extra-Skript meist die bessere Wahl.

1.3 Stellen Sie Vorlesungsmaterialien online zur Verfügung.

Informatikstudenten haben heutzutage Internet-Zugang. Wenn nicht von zu Hause, dann wenigstens über die Informatik-Pools. Früher (und in anderen Fakultäten noch heute) war es üblich, lediglich Kopiervorlagen herauszugeben. Digitale Medien haben aber viele Vorteile:

- Das Herunterladen ist aber für viele günstiger und schneller, als Kopien der Vorlesungsunterlagen anzufertigen oder zu besorgen.
- Zudem erlauben digitale Medien eine farbige Darstellung, die bei normalen Kopierern verloren gehen würde.
- Außerdem gibt es keinen Qualitätsverlust.
- Mit digitalen Medien lässt sich die Zettelwirtschaft verringern. Dateien lassen sich bequem in Verzeichnisse schachteln.
- In digitalen Medien kann man nach Stichworten suchen lassen.
- Digitale Medien können interaktiv sein.
 - Beispiel: Die Seminare zur Vorlesung „Signalverarbeitung“ von Jens-Peer Kuska und Uwe Graichen bauen auf „Mathematica-Notebooks“ auf, also interaktiven Dokumenten. In diesen kann man die Formeln ändern, und darauf ändert sich auch die abhängigen Berechnungen und Grafiken.

1.3.1 Haben Sie urheberrechtliche Bedenken wegen der Online-Freigabe von Dokumenten, dann beschränken Sie den Zugriff durch Passwort und IP-Adressen.

Häufig werden in Vorlesungsunterlagen Grafiken aus Büchern anderer

Autoren verwendet. (Oft ist es dem Dozenten auch von seinem eigenen Verlag verboten, das Grafiken aus dem eigenen Buch freizugeben.) In diesem Fall gehen Sie am besten so vor:

- Sie geben in der ersten Veranstaltung URL, einen Benutzernamen und ein Passwort bekannt für den Zugriff auf das Dokument. Diese Daten sollten auch im Dokument selbst stehen.
- Sie konfigurieren den Web-Server so, dass der Zugriff dann erlaubt wird,
 - wenn Benutzernamen und Passwort per HTTP-Authentifizierung stimmen oder
 - wenn die Anfrage von einem Rechner aus dem Netz der Universität Leipzig kommt. (Lokale Nutzer haben also Zugriff auch ohne das Passwort zu kennen.) Derzeit erreichen Sie dies, indem sie die IP-Adresse des HTTP-Clients daraufhin überprüfen, ob sie im Netz 139.18.0.0/16 liegt.

1.3.2 Stellen Sie Vorlesungsmaterialien als .pdf-Datei zur Verfügung.

PDF ist derzeit das Dokumentenformat der Wahl für Dokumente, die gedruckt werden sollen. Es ist standardisiert, recht kompakt und man kann darin automatisiert nach Text suchen lassen.

- Beschränken Sie sich also nicht nur auf PostScript-Dateien. Diese sind meist sehr groß, und wenn sie komprimiert sind, dann ist ihr „Handling“ umständlicher.
- Beschränken Sie sich also auch nicht auf HTML. Denn HTML hat noch unzureichende Unterstützung für mathematische Formeln und Vektor-Grafik.
- Zwingen Sie also den Studenten nicht, kommerzielle Software gekauft zu haben, um die Materialien anzeigen und drucken zu können. Das heißt zum Beispiel, sich nicht auf die Herausgabe von MS-Office-Dokumenten zu beschränken, denn es existiert keine MS-Office-Anzeige-Software für Linux von Microsoft. Das soll Sie aber nicht davon abhalten, kommerzielle Software für die Erstellung der Materialien zu nutzen:

1.3.3 Stellen Sie die Quellen für Ihre Vorlesungsmaterialien zur Verfügung.

Die Konversation in PDF ist oft nicht verlustfrei. Und selbst wenn: Stehen die Quellen zur Verfügung, dann lassen sich alternative Ansichten auf das Dokument leichter erzeugen, wie Drucken mit kleinerer Schrift oder weniger gefärbten Flächen (spart Zeit und Tinte).

Es soll Informatik-Studenten geben, die über InstantMessaging lernen. Diese können beispielsweise Original-Dokumente mit Einrückungen besser zitieren als die PDF-Versionen, weil bei letzteren die Einrückungen (nach „Kopieren“, „Einfügen“) verloren gehen.

- Ein gutes Beispiel sind die Materialien zur Vorlesung „Digitale Informationsverarbeitung“ (2003 Wintersemester) von Volker Gruhn. Es werden nicht nur PDF-Dateien, sondern auch die dazugehörigen PowerPoint-Dateien mit freigegeben.

1.3.4 Benutzen Sie eine hochqualitative Konvertierung in das PDF-Format.

Das heißt:

- Verwenden Sie keine Bitmap-Schriftsätze. Diese sehen (zumindest bei älteren Adobe Reader Versionen) scheußlich aus, da sie in der Regel nicht interpoliert werden. Manch Konvertierungs-Verfahren mit älterer Software (wie etwa Drucken in einen virtuellen PostScript-Drucker mit anschließender Konvertierung durch GhostScript nach PDF) rastert während der Konversation die Vektor-Schriftsätze.
- Stellen Sie sicher, dass Text zu Text konvertiert wird. Manchmal (ebenfalls bei älteren GhostScript-Konvertierungs-Verfahren) wird jeder Buchstabe zu einem Grafik-Objekt konvertiert. Das bedeutet, dass man in der PDF-Datei nicht mehr nach Zeichenketten suchen kann, was sehr ärgerlich ist.
- Testen Sie ihre PDF-Datei mit dem „Acrobat Reader 4.0“. Dieser ist immer noch sehr beliebt, da seine Ladezeit viel kürzer ist als die der Folgeversionen.

1.3.5 Geben Sie eine Version der Vorlesungsfolien heraus, wo jede Folie auf genau einer PDF-Seite steht.

Sind die Folien so auf PDF-Seiten abgebildet, dann kann der Nutzer selbst am flexibelsten entscheiden, wie er drucken möchte (eine Folie pro Seite, 2 Folien pro Seite, 4 Folien pro Seite, ...).

Zudem ist die Betrachtung dann im Vollbildmodus des Anzeige-Programms wesentlich effizienter, es muss nicht gescrollt und gezoomt werden.

Oft werden PDF-Dateien mit 2 Folien pro Seite herausgegeben. Dies ist gut, denn viele Studenten wissen nicht, wie sie mehrere logische Seiten auf eine physische Seite drucken. Dies soll aber die 1-Folie-pro-Seite-Version nicht ersetzen.

- Ein gutes Beispiel sind die Vorlesungsfolien von Udo Keschull. Die meisten werden in Versionen mit 1,2,4 und 9 Folien pro Seite zum Download angeboten.

1.3.6 Benutzen Sie lediglich Weiß als Hintergrundfarbe für zu druckende Dokumente.

Wenn Sie andere Farben benutzen, dann müssen diese zusätzlich gedruckt werden, was Zeit und Geld kostet.

1.3.7 Scheuen Sie sich nicht, farbige Folien zu erstellen.

Hat Farbe allerdings didaktischen Wert, dann sollte sie auch eingesetzt werden. Viele Studenten haben Farb-Tintendrucker (und einen Farb-Monitor sowieso). Wenn möglich sollten aber relativ dunkle Töne benutzt werden für die wichtigen Objekte, denn manche Studenten drucken nur schwarzweiß.

1.3.8 Stellen Sie Prozesse nicht durch Lücken, sondern durch Animationen dar.

Manchmal ist es didaktisch sinnvoll, bei bestimmten Berechnungsverfahren auf der Folie nur die Aufgabe zu stellen und während der Vorlesung die Lösung handschriftlich einzutragen. Jemand, der die Folien zum Lernen

herunterlädt, kann dann aber nur raten, was in den Lücken stehen soll, in denen die Lösung stehen soll. Dies ist didaktisch nicht sinnvoll. Besser ist es, den Prozess der Berechnung als Animation in die Folien einzubauen, also den Folienaufbau in zwei Schritte zu unterteilen: Nach einem Schritt steht nur die Aufgabe ohne Lösung da, nach dem zweiten Schritt ebenfalls die Lösung.

1.3.8.1 Stellen Sie Animationen auf mehreren Folien dar, eine Folie pro Animations-Frame.

Haben Sie eine Animation, bei der auf jedem Animations-Frame etwas unterschiedliches gezeigt wird, dann sollte jedes Frame auf einer Extra-Folie (und damit auf einer Extra-PDF-Seite) dargestellt sein. Dann lässt sich bei der Online-Ansicht bequem zwischen den einzelnen Frames umherschalten.

- Ein gutes Beispiel hierfür sind die Folien zur Vorlesung „Rechnerarchitektur“ von Udo Keschull.

Aber:

1.3.8.2 Stellen Sie nichtersetzende Animationen auf einer Folie dar.

Eine nichtersetzende Animation ist eine Animation, wobei in jedem Frame nur etwas hinzukommt (etwa wie oben die Lösung zur Aufgabenstellung) und nichts (außer der weiße Hintergrund) überschrieben wird. Dann macht es keinen Sinn, eine solche Animation auf mehrere Folien zu verteilen.

1.4 Wenn Sie Vorlesungsmaterialien nach der Herausgabe ändern: Kennzeichnen Sie die Vorlesungsmaterialien mit einer Versionsnummer oder einem Versionsdatum

Für den Studenten ist es einigermaßen wichtig zu wissen, ob er die Materialien noch einmal herunterladen muss oder nicht.

- Ein gutes Beispiel ist die Web-Seite zur Vorlesung „Algorithmen und Datenstrukturen“ (2002 Wintersemester) von Uwe Quasthoff. Neben jedem Link auf Materialien steht das Datum der letzten Änderung dieser Dateien.
- Ein weiteres gutes Beispiel sind die Folien-Dateien von Udo Keschull. Sie enthalten im Dateinamen eine Versionsnummer.

1.5 Geben Sie eine konsolidierte Version ihrer Folien heraus.

Das heißt: In einer einzigen Datei sollen alle Folien enthalten sein. Dann lässt sich nämlich sehr schön in dieser einzigen Datei suchen. Suchen über mehrere Dateien ist gerade bei PDF-Dokumenten nicht möglich und auch sonst sehr umständlich. Heutzutage ist sogar Modem-Nutzern zumutbar, ein PDF-Dokument von 4MB Größe herunterzuladen.

- Ein gutes Beispiel sind die Folien-Dateien von Udo Keschull.

Manche Dozenten ziehen es trotzdem vor, für jede Veranstaltung jeweils eine Datei herauszugeben. In diese Fall gilt:

1.5.1 Numerieren Sie die Dateien einer Dateiserie durch, sodass die Sortierung nach Namen eine Sortierung nach Zeit ergibt.

Das heißt: Die Nummerierung im Dateinamen sollte am Anfang des Namens sein, nicht am Ende. Wenn Sie ein Datum mit in den Dateinamen nehmen, dann sollte es vom Format „JJJJ.MM.TT“, nicht „TT.MM.JJJJ“ sein.

- Gutes Beispiel:
 - „kap1.pdf“,
 - „kap2.pdf“,
 - „kap3.pdf“,
 - ...
- Noch besseres Beispiel:
 - „[01] Einführung.pdf“,
 - „[02] Spezielles.pdf“,
 - „[03] Zusammenfassung.pdf“,
 - ...
- Schlechtes Beispiel:
 - „twbs_einfuehrung_pdf_op.pdf“
 - „twbs_logik_inferenz_op.pdf“
 - „twbs_logik_inferenz2_pdf_op.pdf“
 - „twbs_asp_pdf_op.pdf“

1.6 Tragen Sie ihr Online-Angebot im Lern-Server ein.

Unter <http://www.informatik.uni-leipzig.de/lehre/lernserver.html> befindet sich die jeweils aktuelle Fassung von Verweisen auf Vorlesungsmaterialien zu jeweils aktuellen Veranstaltungsreihen. Diese Web-Seite ist eine zentrale Anlauf-Stelle zum Finden von Materialien.

2 Lehrweise

2.1 Definitionen

2.1.1 Geben Sie den erwarteten Kenntnisstand für die Vorlesung an.

Das heißt, dass Sie die Vorlesungen nennen, die bereits gehört worden sein sollen. Dies soll bereits in der Vorlesungankündigung (im Vorlesungsverzeichnis) stehen, und außerdem noch einmal am Anfang des Skripts erwähnt werden.

2.1.2 Markieren Sie Definitionen als solche.

Das heißt, dass Sie zu definierende Begriffe nicht im Vorbeigehen erwähnen, sondern einen gewissen Formalismus einhalten. Damit wird erkenntlich, dass ein Begriff definiert wird, auf den später verwiesen wird. Gleichzeitig lässt sich im Online-Skript nach einem solchen Begriff suchen, sodass man sofort seine Definition findet.

- Beispiel:

Definition: gleich (Mengen)

Für alle Mengen M_0 und für alle Mengen M_1 gilt: Die Mengen M_1 und

M_2 heißen genau dann **gleich** (Schreibweise: $M_0 = M_1$), wenn

- M_0 Teilmenge von M_1 und
- M_1 Teilmenge von M_0 ist.

$$\forall (M_0) : \forall (M_1) : ((M_0 = M_1) \Leftrightarrow ((M_0 \subseteq M_1) \wedge (M_1 \subseteq M_0)))$$

2.1.3 **Beziehen Sie sich nur auf vorher definierte Begriffe.**

„Um Rekursion zu verstehen, muss man vorher Rekursion verstehen.“

Das heißt, dass beim Versuch, Ihr Skript zu verstehen, man nur zurück gehen muss (gegebenenfalls in vorherige Vorlesungen), nicht vor. Sind sie nicht sicher, ob Begriffe bereits in vorherigen Vorlesungen definiert wurden, dann definieren Sie diese noch einmal in Kurzform.

2.1.4 **Wenn Sie rekursive Definitionen verwenden, dann beschränken Sie sich auf primitiv-rekursive Definitionen.**

Das heißt also, dass der Versuch, Ihre Definition schon auf syntaktischer Ebene zu verstehen (sie am konkreten Beispiel nachzuvollziehen), in endlicher Zeit schaffbar sein muss.

2.2 **Schreiben Sie kanonisch.**

Das heißt: Gibt es eine eindeutige, bevorzugte Form, etwas zu schreiben, dann schreiben sie so, und nur so.

2.2.1 **Wenn Sie von der kanonischen Form abweichen wollen, dann deklarieren Sie dies explizit.**

Wenn es schon unbedingt sein muss, etwas anders zu schreiben als üblich, dann sollte dies ausdrücklich gesagt werden.

2.2.2 **Benutzen Sie standardisierte Begriffe und Zeichen.**

Das heißt zum Beispiel:

- Die imaginäre Zahl $\sqrt{-1}$ soll i heißen, und nicht j .
- Nehmen Sie als Funktionssymbole die Zeichen f, g, h .
- Nehmen Sie als Laufvariable für ganze Zahlen lieber die Variablen k, l, m und besser gar nicht i .
- Nehmen Sie als Dimensionsvariable für natürliche Zahlen n .
- Indizieren Sie eine Variable, die eine Schrittweite symbolisiert, mit einem großen Delta. (Beispiel: t_Δ)

2.2.3 **Benutzen Sie eindeutige Begriffe und Zeichen.**

- Verwenden Sie nicht den griechischen Buchstaben „Chi“ X (groß) oder χ (klein). Verwenden Sie auch kein großes X , sondern lediglich x .
- Nehmen Sie nicht den griechischen Buchstaben „Kappa“ K (groß) oder κ (klein). Verwenden Sie auch kein großes K , sondern lediglich k .
- Unterscheiden Sie nicht „geschwungene“ Buchstaben (wie 'A', 'D') von „nicht geschwungenen“ Buchstaben (wie 'A', 'D').
- Benutzen Sie im Zweifel lieber Symbole, die aus mehr als einem Zeichen bestehen (zum Beispiel: „sin“)

- Gehen Ihnen die Bezeichner aus und wollen Sie keine Symbole aus mehreren gleichberechtigten Zeichen verwenden, dann indizieren Sie ein einziges Symbol (Beispiel: k_0, k_1, k_2, k_3 statt k, l, m, n).
- Schreiben Sie bei jeder Multiplikation einen Multiplikationspunkt. („ $w \cdot r$ “ statt „ wir “) Dann haben Sie auch die Freiheit, Namen aus mehreren Zeichen zusammenzubauen. Dann ist auch eindeutig, was $a(b+c)$ ist, nämlich keine Multiplikation, sondern die Anwendung der Funktion a auf den Parameter $b+c$,

2.2.4 Benutzen Sie keine Abkürzungen.

„Da der Ex-VP so ein VIP ist, sollten wir die PK wegen der PR auf dem WC und nicht im TV machen. Und dann sagt die MP zum KGB „LMAA“.“

Abkürzungen machen im besten Fall das Lesen umständlich und im schlechtesten Fall das Verständnis unmöglich. Menschen sind keine sed-Interpreter (und Sie wissen doch, was die Abkürzung „sed“ bedeutet?).

- Schlechtes Beispiel: „SD- bzw- SN-artige Kopplung von SE-Systemen“
- Schlechtes Beispiel: „PIs“ (statt „ProcessingInstructions“)
- Schlechtes Beispiel: „LE“ (statt „Lern-Einheit“ oder „Leipzig“)
- Schlechtes Beispiel: „GP“ (statt „Geschäftsprozess“ oder „Grand Prix“)

2.2.4.1 Benutzen Sie aussagekräftige Bezeichner.

Was in jeder Veranstaltung über gutes Software-Schreiben gepredigt wird, gilt sowohl für guten Beispiel-Quellcode als auch für auf Verständnis ausgerichtete Skripte und Folien.

2.2.4.2 Schreiben Sie lieber ausführlich und korrekt statt nach „Jeder weiß, was wir meinen.“

Wenn es jeder wissen würde, dann bräuchten Sie gar nichts aufzuschreiben. Ein Beispiel für Ausführlichkeit ist unter Empfehlung 2.3.1.7.2 zu finden.

2.3 Benutzen Sie die Prädikatenlogik.

Die Prädikatenlogik ist für viele Definitionen, Sätze und Beweise ausreichend mächtig. Sie erlaubt es, prädikatenlogische Ausdrücke ineinander einzusetzen, damit der Student auf syntaktischer Ebene schließen und somit Beispiele durchrechnen kann, um Verständnis zu erlangen, wo er es noch nicht hat. (Im Gegensatz dazu setzt die semantische Schlussfolgerung das Verständnis voraus, ist also nicht so erhellend.) Die Prädikatenlogik sollte zumindest in Vorlesungen ab dem 3. Semester benutzt werden. (Im 2. Semester wird die Prädikatenlogik vorgestellt).

(Sie können auch eine abgewandelte Schreibweise der Prädikatenlogik verwenden, wenn diese Eindeutig auf die klassische Prädikatenlogik abbildbar ist und Sie diese Abbildung angeben.)

2.3.1 Schreiben Sie kontextfrei.

Das heißt: Schreiben Sie so, dass es auch ein Computer verstehen kann. Dann kann auch der Student leichter verstehen und muss mangels

Eindeutigkeit nicht rätseln.

- Schlechtes Beispiel: [...fehlt noch...]

Das heißt auch:

2.3.1.1 **Benutzen Sie die eckigen Klammern ']', '[' für offene Intervalle, nicht runde Klammern.**

Was bezeichnet $a=(0,1)$? Ist a ein Paar von 0 und 1? Oder ist a das Intervall zwischen 0 und 1, was weder 0 noch 1 enthält?

Um solches Rätselraten zu vermeiden, vereinbaren Sie folgendes:

- $a=]0,1[$ ist ein Intervall.
- $a=(0,1)$ ist ein Paar.

2.3.1.2 **Benutzen Sie Klammern für Tupel**

$()$ ist ein 0-Tupel.

(0) ist ein 1-Tupel.

$(0,1)$ ist ein Paar.

$(0,1,2)$ ist ein Tripel.

Leider ist das 1-Tupel nicht eindeutig, es könnte auch stattdessen ein einfaches Objekt sein (in diesem Fall die 0). Abhilfe schafft da nur die Benutzung anderer Klammern für Tupel.

- Runde Klammern $()$ eignen sich nicht wegen allgemeiner Klammerung zur Vorrang-Anzeige,
- eckige Klammern $[]$ eignen sich nicht, weil sie für Intervalle verwendet werden,
- geschweifte Klammern $\{\}$ eignen sich nicht, weil sie für Mengen verwendet werden.

Aus diesem Grund benutzen einige Leute spitze Klammern $\langle \rangle$. Unglücklicherweise werden diese bereits für das Skalarprodukt verwendet. Sie können diese spitzen Klammern wegen der Eindeutigkeit für Tupel verwenden, aber dann müssen Sie dies am Anfang so erklären.

2.3.1.3 **Benutzen Sie die geschweiften Klammern '{','}' ausschließlich für Mengen.**

Und nicht für etwas anderes. Das Objekt $c=\{-0.9,0.1\}$ ist eine Menge, keine komplexe Zahl $c=-0.9+i\cdot 0.1$ (wie zum Beispiel in einer Signalverarbeitungs-Klausur zu finden ist).

2.3.1.4 **Das „Komma“ ',' ist kein „und“ '^'.**

Das Komma trennt einzelne Objekte in Aufzählungen.

2.3.1.5 **Quantifizieren Sie alle Variablen.**

Das heißt: Geben Sie für jede Variable in ihrem prädikatenlogischen Ausdruck an, ob sie existenz- oder all-quantifiziert ist. Jede vorkommende Variable muss ausdrücklich gebunden sein.

Das heißt auch: Geben Sie überhaupt einen prädikatenlogischen Ausdruck an.

- schlechtes Beispiel: $\sin(k\pi)=0, k \in \mathbb{Z}$
- gutes Beispiel: $\forall (k \in \mathbb{Z}) : (\sin(k\pi)=0)$
- korrektes Beispiel: $\forall k : ((k \in \mathbb{Z}) \Rightarrow (\sin(k\pi)=0))$

Das mittlere Beispiel entspricht zwar nicht ganz der „reinen Prädikatenlogik“, ist jedoch verständlicher und kürzer. Es ist eindeutig auf das die „reine Prädikatenlogik“ zurückführbar, wie man am unteren Beispiel sieht.

Die Quantifizierung jeder Variable stehe immer links von der Benutzung dieser Variablen.

2.3.1.6 Qualifizieren Sie alle Variablen.

Das heißt: Geben Sie für jede Variable an, aus welcher Menge sie stammen. Die Qualifizierung jeder Variable stehe immer links von der Benutzung dieser Variablen. (Wie in obigem Beispiel mit $k \in \mathbb{Z}$.) Sagen Sie nicht „wir betrachten im Folgenden nur ganze Zahlen“, um die Qualifizierung weglassen zu können.

Dies ist besonders wichtig für Funktionen, die induktiv über Elemente von Mengen definiert sind, wie zum Beispiel die Summe.

- Ein schlechtes Beispiel (aus dem Skript der Vorlesung „Theoretische Grundlagen Wissensbasierter Systeme“ 2003 Wintersemester) ist:

$$P(R_3|S_3) = \frac{\psi(C_3)}{\sum_{R_3} \psi(C_3)}$$

Hier ist nicht klar, worüber eigentlich die Summe gebildet wird. Ist jeder Summand abhängig von einem Element von R_3 ? Was ist die Laufvariable?

2.3.1.7 Benutzen Sie Ausdrücke nicht polymorph.

Es sollte möglich sein, jedem Objekt einen eindeutigen Datentyp zuzuordnen. Das heißt:

- Wenn
 - 'a' ein Zeichen ist und
 - eine Zeichenkette ein Tupel von Zeichen ist,
 dann ist 'a' keine Zeichenkette, aber ('a') ist eine Zeichenkette.
- Wenn die Funktion $head(r)$ den Kopf einer Regel eines Logikprogramms zurückgibt, dann ist das entweder ein Literal oder eine Menge von Literalen, aber nicht einmal das eine und einmal das andere.

2.3.1.7.1 Benutzen Sie Variablen nicht polymorph.

- Schlechtes Beispiel (aus der Vorlesung „Algebraische Automatentheorie“ von Manfred Droste):

Definition: Monoid

Sei M eine Menge und $f : M \times M \rightarrow M$ eine Abbildung.
 (M, f) heißt Monoid, wenn [...]

und später:

Definition: Untermonoid

Sei M ein Monoid und $U \subseteq M$. U heißt **Untermonoid** genau dann, wenn $1 \in U$ und $\forall (a \in U) : \forall (b \in U) : ((a \cdot b) \in U)$.
 Man schreibt in diesem Fall $(U, \square \cdot \square) \subseteq (M, \square \cdot \square)$.

- Ein Monoid ist eine Liste bestehend aus zwei Elementen (einer Menge und einer Funktion), wie oben definiert. Beispielsweise ist $M = (K, f)$. Was ist dann $U \subseteq (K, f)$? Bedeutet dies $U \subseteq \{K, f\}$? Gemeint ist wohl $U \subseteq K$. Wenn aber $U \subseteq K$, wie kann dann die Menge U ein (Unter-)Monoid sein, wenn doch Monoide Listen und nicht Mengen sind?
 Was eigentlich ist das für eine Multiplikation bei $(a \cdot b) \in U$? Gemeint ist wohl „ $(U, \square \cdot \square)$ heißt **Untermonoid** ...“. Was aber ist dann $\square \cdot \square$? Seit wann kann wie bei $(U, \square \cdot \square) \subseteq (M, \square \cdot \square)$ eine Liste eine Teilmenge einer anderen Liste sein?
- Besseres Beispiel:

Definition: Untermonoid

Sei (M, f) ein Monoid und $U \subseteq M$. (U, f) heißt **Untermonoid** genau dann, wenn $1 \in U$ und $\forall (a \in U) : \forall (b \in U) : (f(a, b) \in U)$.

Man beachte, dass die Schreibweisenempfehlung vollständig entfernt wurde, da diese Schreibweise verwirrungstiftend ist.

2.3.1.7.2

Benutzen Sie Operatoren nicht polymorph.

- Schlechtes Beispiel (aus der Vorlesung „Algebraische Automatentheorie“ von Manfred Droste):

Definition: Homomorphismus

Seien M, M' Monoide.
 Eine Abbildung $f : M \rightarrow M'$ heißt **Homomorphismus** genau dann, wenn

- $f(1_M) = 1_{M'}$ und
 - $\forall (a \in M) : \forall (b \in M) : (f(a \cdot b) = f(a) \cdot f(b))$.
- Was ist bei „ $f(a \cdot b) = f(a) \cdot f(b)$ “ das eigentlich für eine Multiplikation? Man könnte meinen, es ist die selbe? Ist es aber nicht.
 - Besseres Beispiel:

Definition: Homomorphismus

Seien H_0, H_1 Monoide mit $H_0=(M_0, f_0), H_1=(M_1, f_1)$.

Sei 1_{H_0} das neutrale Element von H_0 .

Sei 1_{H_1} das neutrale Element von H_1 .

Eine Abbildung $g: M_0 \rightarrow M_1$ heißt **Homomorphismus** genau dann, wenn

- $g(1_{H_0})=1_{H_1}$ und
- $\forall (a \in M): \forall (b \in M): (g(f_0(a, b))=f_1(g(a), g(b)))$.

Diese Definition sieht länger und komplizierter aus, ist aber korrekter.

2.3.2

Setzen Sie Klammern.

Die Prädikatenlogik fordert, dass um alle zweistelligen Junktoren Klammern gesetzt werden. Tun Sie dies, selbst wenn Sie wissen, dass Sie über eine Definition von Bindungsstärken einige Klammern weglassen können. Es erleichtert das Verarbeiten auf syntaktischer Ebene.

- schlechtes Beispiel: $a \wedge c \vee d \Rightarrow b \Leftrightarrow f \vee g \wedge h$
- gutes Beispiel: $((a \wedge c) \vee d) \Rightarrow b \Leftrightarrow (f \vee (g \wedge h))$
- schlechtes Beispiel: $\sum_{k_0=0}^{n_0-1} f(k_0) + \sum_{k_1=0}^{n_1-1} g(k_1)$
- gutes Beispiel: $\sum_{k_0=0}^{n_0-1} \left(f(k_0) + \sum_{k_1=0}^{n_1-1} g(k_1) \right)$
- gutes Beispiel: $\sum_{k_0=0}^{n_0-1} (f(k_0)) + \sum_{k_1=0}^{n_1-1} (g(k_1))$

Das Grundprinzip muss sein: Ohne großen Aufwand lassen sich vom Studenten selbst beliebige Beispiele wählen (durch die Qualifizierung ist die Menge möglicher Werte bekannt) und auf syntaktischer Ebene ausrechnen (durch einfaches Einsetzen von Ausdrücken in andere Ausdrücke).

3 Vorlesungsgestaltung

3.1

Setzen Sie die Veranstaltungen im Zeitraum Montag..Donnerstag 9:00..17:00 Uhr und Freitag 9:00..15:00 an.

Auch wenn Sie selbst nur begrenzten Einfluss auf die Veranstaltungszeiten haben: Viele Informatiker sind Spätaufsteher und Spätinsbettgeher. Insgesamt sollte das Veranstaltungsangebot gleichmäßig über die Woche in diesen Zeiten verteilt sein. Besonders mittwochs kommt es zu Anhäufungen von Veranstaltungsangeboten, dafür sieht es freitags meist recht rar aus. Das sollte nicht so sein, denn jeder Student kann nur in einer Veranstaltung gleichzeitig sein.

- Schlechtes Beispiel: Vorlesung „Videokommunikation“ montags um 7:30 Uhr bis 10:30 Uhr (2003 Wintersemester)

Konkret heißt dies:

- Dozenten, die in Leipzig permanent wohnen, sollten bevorzugt

Veranstaltungen im Zeitraum Donnerstag Nachmittag bis Dienstag Vormittag erhalten, damit die restliche Zeit für diejenigen Dozenten übrig bleibt, die nur von Dienstag bis Donnerstag in Leipzig wohnen.

Auch das Argument „7:30 Uhr kommen dann noch diejenigen, die es wirklich interessiert.“ ist bei genauerer Betrachtung wenig schlüssig, da der Körper auf einen gleichmäßigen Tagesstart getaktet werden muss. Es ist allgemein anerkannt, dass kognitive Leistungsfähigkeit bei den meisten Menschen gegen 8 Uhr noch fast auf Schlaf-Niveau ist, während sie gegen 11 Uhr ihr (erstes) Tages-Maximum erreicht, sodass selbst die Forderung nach einer „Umtaktung“ der interessierten Studenten wenig aussichtsreich ist.

3.2 Geben Sie Beispiele.

Manche Dozenten neigen dazu, ein Thema ausschließlich abstrakt zu behandeln. Die meisten Aussagen werden nicht lediglich über konkrete Objekte, sondern über alle Objekte einer Klasse gemacht. Da in der Regel eine Aussage auf einer anderen aufbaut, ist es oft aussichtslos, eine Aussage verstehen zu können, wenn man nicht bereits alle vorherigen Aussagen verstanden hat. In diesen Fällen helfen komplexere Beispiele und Rechnungen, die mehrere Aussagen benutzen. Der Student erlangt so besser ein intuitives Gefühl für den Wesensgehalt der vorgestellten und durch Beispiele nachempfundenen Aussagen.

3.3 Stellen Sie Übungsaufgaben, die mindestens auf Prüfungs-Niveau sind.

Der Student sollte vorlesungsbegleitend schon vor der Prüfung ein Gefühl dafür entwickeln, welche Art von Aufgaben ihn erwarten werden.

- Zielen die Aufgaben auf Verständnis (Aufgabenlösung ist eine Erklärung) ab,
- oder eher auf Rechnen (Aufgabenlösung ist eine exakte Zahl),
- oder eher auf Fakten (Aufgabenlösung ist eine Liste von Stichworten)?

3.3.1 Geben Sie Musterlösungen heraus oder bieten Sie eine Korrektur an.

Damit der Student auch weiß, wie er im Stoff steht, sollte seine Lösung bewertet werden. Diese Bewertung kann er selbst vornehmen, oder aber (und dies ist besser) vornehmen lassen. Im Grundstudium sollte in jedem Fall ein Übungsbetrieb statt finden, wo also Übungsaufgaben vom Studenten abgegeben werden und dieser die Übungsaufgaben korrigiert und bewertet zurück erhält.

3.3.2 Reicht die personelle oder finanzielle Kapazität für einen Übungsbetrieb nicht aus, so überlassen Sie dies einem Automaten.

Sie ist rar, aber es gibt sie: Software, die Übungsaufgaben an Studenten stellt und Lösungsvorschläge korrigiert. Dazu müssen die Studenten zwar Lösungsvorschläge in einem vorgegebenen Format formulieren, dies ist aber nicht sonderlich schwer. Ist ein Lösungsvorschlag syntaktisch wohlgeformt, dann kann er anhand eines plausiblen Verfahrens auf Richtigkeit überprüft werden. Ist dies nicht der Fall, dann kann die Software begründen, warum die Lösung nicht richtig sein kann. Ein Automat kann so viel mehr Studenten bewältigen als studentische Hilfskräfte. Einzig solche Software selbst muss geschrieben werden, und dies ist leider nicht jedem möglich. In jedem Fall ist

es lohnenswert, solche Ideen zu verfolgen.

- Ein positives Beispiel ist Johannes Waldmann mit seiner Software „autotool“, über welches der Übungsbetrieb für die Vorlesung „Berechenbarkeit und Komplexität“ durchgeführt wurde (2003 Sommersemester).

Ein Nebeneffekt von Automaten ist, dass der Student beim Lernen flexibler ist. Denn mit einem Automaten, der sofort eingesandte Lösungsvorschläge korrigiert und bewertet, lässt sich unabhängiger von Ort und Zeit lernen und verstehen.

3.3.3 Stellen Sie den Unterschied zwischen Prüfungsniveau und Übungsaufgabenniveau heraus.

Oft sind Übungsaufgaben um einiges schwieriger als Prüfungsaufgaben, da erstere der Vertiefung dienen und letztere unter sehr eingeschränkter Zeit zu lösen sind. Zeigen sie auf welchem Niveau die Prüfungsaufgaben zu erwarten sind. Zum Beispiel auch so:

3.3.4 Geben Sie eine Beispiel-Klausur oder alte Klausuren heraus.

Daran können sich Studenten authentisch testen, um ihren Leistungsstand besser abschätzen und ihre Unsicherheit minimieren zu können.

- Ein gutes Beispiel sind die Beispiel-Klausuren für „Rechnerarchitektur“ und „Entwurf hochintegrierter Schaltkreise“ (2003 Wintersemester) von Udo Kebschull.
- Ein gutes Beispiel ist die Beispiel-Klausur für „Numerische Mathematik“ (2002 Wintersemester) von Peter Kunkel.

3.3.5 Spezifizieren Sie die Anforderungen an den Prüfling möglichst genau.

Zusätzlich dazu, „durch die Blume“ zu sagen, wie die Anforderungen sind, indem Sie entsprechende Übungsaufgaben stellen, können Sie diese natürlich auch offen und ausdrücklich Formulieren. Dies reicht zwar in der Regel nicht, um den Schwierigkeitsgrad abzuschätzen (deswegen sollten Beispiel-Aufgaben auch vorhanden sein), wohl aber den Aufgabentyp.

- Gutes Beispiel (Vorlesung von Maik Thränert: „Anwendungen mit XML“ (2003 Wintersemester)):

Lernziele

- DTD von Hand erstellen können
[...]

- Einfache XML-Dokumente gegen ein einfaches Schema prüfen

Durch die Angabe von Lernzielen in den Folien wird hier ausgesagt, dass der Prüfling eine DocumentTypeDefinition in der Prüfung selbst erstellen können muss, ein XML-Schema (komplexerer Nachfolger der DocumentTypeDefinition) jedoch nicht. Er muss allerdings ein XML-Dokument gegen ein XML-Schema validieren können.

Der Prüfling wird also dadurch von der Unsicherheit entlastet, ob er XML-Schema-Dokumente selbst unter Zeitdruck schreiben können muss oder nicht. Dies hebt die Motivation.

3.4 Veranstalten Sie Seminare oder interne Praktika zur Vorlesung.

Es gibt Themen, bei denen der Glaube, es verstanden zu haben, nicht ausreicht, um es verstanden zu haben. Unter anderem für diese sollten Seminare eine Vertiefung des Stoffs ermöglichen. Werden in Vorlesungen vorrangig Definitionen und Sätze gebracht, so sollten in Seminaren Beispiele und „corner cases“ (als Fälle, in denen etwas unerwartetes passiert) betrachtet und durchgerechnet werden.

Interne Praktika haben ähnlich vertiefende Effekte, jedoch werden da meist Aufgaben gestellt, die selbstständig und oft in Gruppen zu lösen sind.

- Ein gutes Beispiel ist das Seminar zur Signalverarbeitung (2003 Wintersemester) von Jens-Peer Kuska.
- Ein gutes Beispiel ist das Praktikum zur Vorlesung „Theoretische Grundlagen wissenschaftlicher Systeme“ von Gabriele Kern-Isberner und Gerhard Brewka.

3.4.1 Veranstalten Sie die Seminare nicht selbst, sondern überlassen Sie dies einem Mitarbeiter.

Seminare sind dazu da, eine umfassendere Sicht auf das Thema zu erhalten. Dazu gehört auch, dass jemand anderes die Themen vermittelt. Denn möglicherweise hat er einen ganz anderen Stil als der Lesende, sich dem Thema zu nähern. Durch unterschiedliche Personen und Vermittlungsstile werden die Chancen vergrößert, dass der Student aus allen Kombinationen das Beste nutzen kann.

3.4.2 Veranstalten Sie die Seminare in zeitlichem Abstand zur Vorlesung, nicht am selben Tag.

Der Student braucht Zeit, um bestimmte Eindrücke (und damit auch Vorlesungen) zu verarbeiten. „Man kann zwar über alles reden, aber nicht über eine Stunde.“ Das heißt, dass das Stressen eines einzigen Themas über längere Zeit Motivationsverluste mit sich bringt.

3.5 Informieren Sie sich, welche Studiengänge Ihre Veranstaltung besuchen.

Einige Dozenten wissen nicht, für welche Studiengänge die Vorlesung halten. Einige Dozenten wissen nicht einmal, was es überhaupt für Studiengänge gibt, die Ihre Veranstaltung in irgend einer Weise tangieren.

Da gibt es zum Beispiel:

- Informatik (Diplom)
- Informatik (Magister mit Informatik als Hauptfach)
- Informatik (Magister mit Informatik als Nebenfach)
- Informatik (Bachelor)
- Informatik (Master)
- Mathematik
- ...

Wenn es Studiengänge gibt, die Ihre Veranstaltung als Pflicht- oder Wahlpflicht-Veranstaltung aufführen, dann finden Sie heraus, auf welches Vorwissen Sie für jeden Studiengang aufbauen dürfen.

3.5.1 Erklären Sie am Anfang der Veranstaltungsreihe ausdrücklich, für welche Studiengänge in jeweils welchem Studienstadium Ihre Veranstaltungsreihe konzipiert ist.

Idealerweise nennen Sie die Vorlesungen einzeln, auf die Sie aufbauen.

- Beispiel: Für die Vorlesung „Logik“ ist Vorlesung „Mengentheorie“ Voraussetzung.
- Beispiel: Für die Vorlesung „Numerik“ sind die Vorlesungen „Lineare Algebra“, „Analytische Geometrie“, „Digitale Informationsverarbeitung“ und „Programmierung und Programmiersprachen“ sowie ein Programmier-Praktikum Voraussetzung.

Durch eine solche Erklärung kann jeder Student des jeweiligen Studiengangs für sich selbst überprüfen, ob er „reif“ für die Veranstaltung sein müsste.

- Nennen Sie eine Voraussetzung, die der Student auf Grund seiner Studienordnungen nicht kennen konnte, dann liegt eine Inkonsistenz vor.
- Enthält Ihr Auditorium Studenten von Studiengängen, die Sie nicht nennen, dann haben diese durch eine solche Erklärung die Möglichkeit zu erkennen, dass Ihr Studiengang nicht berücksichtigt wurde (also ebenfalls eine Inkonsistenz vorliegt).

Liegen Inkonsistenzen vor, dann sollten Studenten nicht unnötig benachteiligt werden. Das heißt, dass Studenten nicht erst nach der Prüfung erkennen sollen, dass eine Inkonsistenz vorlag, sondern bereits am Anfang der Veranstaltungsreihe. So wird auch das Feedback zum Prüfungsamt des jeweiligen Studiengangs effektiver, sodass solche „Bugs“ im System schneller ausgemerzt werden. Sonst halten diese sich über Jahre.

3.5.2 Nennen Sie ausdrücklich diejenigen Inkonsistenzen zwischen Studienordnungen und den Voraussetzungen Ihrer Veranstaltungsreihe, die Sie kennen.

Wenn Sie herausgefunden haben, woraus Ihr Auditorium besteht, dann haben Sie vielleicht auch schon selbst herausgefunden, dass es einige Studiengänge gibt, von deren Studenten Sie gar nicht einige der Voraussetzungen Ihrer Veranstaltungsreihe erwarten können. Ist dem so, dann erklären Sie ausdrücklich, dass Ihnen dieser Umstand bewusst ist (und dass Sie daran selbst vorerst nichts ändern werden). Studenten können so leichter solche Inkonsistenzen erkennen und ihnen entgegen wirken.

4 Prüfungen und Testate

4.1 Aufgabenstellung

4.1.1 Stellen Sie die Aufgaben präzise.

Klausuren nach der derzeitigen Prüfungsordnung haben wegen der extrem knapp bemessenen Zeit es leider an sich, in erster Linie nicht Wissen oder Fähigkeit zu erfassen, sondern handschriftliche Schreibgeschwindigkeit. Prüflinge müssen wissen, wie umfangreich Sie auf eine Fragestellung eingehen müssen, um die volle Punktzahl zu erreichen. Verwenden Sie

folgende Aufforderungen:

- „Ist ...?“ | „Gibt es ...?“ | „Geben Sie an, ob ...“
wenn Sie einen booleschen Wert erwarten.
- „Welche der folgenden ... hat Eigenschaft ...?“
wenn Sie eine Liste von Einträgen der von Ihnen angegebenen Liste erwarten, die angegebene Eigenschaft hat.
- „Nennen Sie ...“
wenn Sie eine Liste von Begriffen erwarten.
- „Geben Sie ... an“ | „Bestimmen Sie ...“ | „Berechnen Sie ...“ | „Bilden Sie ...“
| „Suchen Sie ...“
wenn nur das Ergebnis zählt.
- „Skizzieren Sie ...“:
wenn Sie eine qualitative Grafik erwarten (wo also lediglich Nullstellen, Polstellen usw. eingetragen sein müssen).
- „Zeichnen Sie ...“
wenn Sie eine Grafik mit quantitativ nachprüfbar Merkmalen erwarten (etwa die Amplitude einer Sinuskurve)
- „Konstruieren Sie ...“
wenn Sie eine exakte Grafik erwarten.
- „Begründen Sie Ihre Antwort kurz.“
wenn Sie eine kurze verbale Begründung erwarten. Eine Beweisskizze ist dabei hinreichend.
- „Begründen Sie Ihre Antwort.“
wenn Sie eine verbale Begründung erwarten. Ein Beweis ist dabei hinreichend, aber nicht notwendig.
- „Skizzieren Sie den Rechenweg.“
wenn Sie mehr als nur das Ergebnis erwarten, jedoch dem Prüfling nicht aufbürden wollen, dass er Dinge, die er im Kopf oder mittels (zugelassenem) Taschenrechner berechnen kann, extra aufschreiben muss.
- „Geben Sie den Rechenweg an.“
wenn Sie einen Rechenweg erwarten.
- „Dokumentieren Sie jeden Schritt Ihrer Vorgehensweise.“
wenn Sie für jeden Schritt wissen wollen, wie vorgegangen wurde.
- „Beweisen Sie ...“ | „Zeigen Sie ...“ | „Beweisen Sie ihre Antwort.“
wenn Sie einen Beweis erwarten.

4.1.1.1

Fragen Sie nicht in Form von Understatements.

- Ein schlechtes Beispiel ist die Modulklausur zur Vorlesung „Theoretische Grundlagen wissenschaftlicher Systeme“ (2003.WS) von Gabriele Kern-Isberner. Darin heißt es
„[...] Bestimmen Sie einen Entscheidungsbaum zur Klassifikation von Filmen.“
Erwartet wurde jedoch eine Antwort auf
„[...] Bestimmen Sie einen Entscheidungsbaum zur Klassifikation von Filmen. Dokumentieren Sie jeden Schritt Ihrer Vorgehensweise.“,
also nicht nur das Ergebnis, sondern auch jede zur Erstellung des Entscheidungsbaums notwendige Schritt, wie selbst einfaches Zusammenzählen. Für das Zusammenzählen wurde eine Strichliste als

Dokumentation erwartet, ohne dass dies in der Aufgabenstellung stand.

4.1.2 Im Zweifel für den Prüfling

Ergeben sich Zweifel an der Interpretation einer Aufgabe, insbesondere über den erwarteten Detailgrad der Antworten, dann bewerten Sie im Zweifel für den Prüfling. Er hatte im Gegensatz zu Ihnen keinen Einfluss auf die Art der Fragestellung.

Der erwartete Detailgrad muss ohne weiteren Kontext eindeutig erkennbar sein. Das bedeutet unter anderem:

Um einen höheren erwarteten Detailgrad zu rechtfertigen sind Einwände wie „In einer Vorlesungsveranstaltung wurde aber noch diese und jene Nebenrechnung an der Tafel gemacht.“ höchstens dann gültig, wenn die Aufgabenstellung sinngemäß lautete „Bestimmen Sie wie in der Vorlesung ...“

4.1.2.1 Gehen Sie nicht von einem Generalverdacht eines Täuschungswillens aus.

Das heißt damit auch, dass solche Argumente ungültig sind „Wenn Sie nur das Ergebnis hinschreiben, hätte es ja auch von Ihrem Nachbarn kommen können.“, wenn Sie nicht eindeutig nach mehr als nur dem Ergebnis gefragt haben.

4.2 Hilfsmittel

4.2.1 Geben Sie die erlaubten Hilfsmittel frühzeitig bekannt.

Das heißt: Geben Sie die erlaubten Hilfsmittel spätestens dann bekannt, wenn ein Prüfling für die Prüfung zu üben versuchen wird. Dies ist in der Regel bereits gleich nach Anfang der Vorlesungsreihe, nachdem der erste Stoff vermittelt wurde, den es zu vertiefen gilt. Wenn Sie Übungsaufgaben stellen, geben Sie neben der erwarteten Lösungszeit auch die erlaubten Hilfsmittel an, wenn diese Übungsaufgabe eine Klausuraufgabe wäre.

4.2.2 Erwägen Sie, ob Sie bei eng zeitbegrenzten Klausuren beliebige Literatur zulassen.

Insbesondere dann, wenn es Ihnen als Prüfer vor allem auf das Verständnis der Zusammenhänge und Methoden, weniger auf Auswendig-Lernen ankommt, sollten Sie in Erwägung ziehen, beliebige Literatur zuzulassen. Das erlaubt es nämlich dem Prüfling, sich auf das für Sie wesentliche - das Verständnis zu konzentrieren. Mit der Zulassung beliebiger Literatur kann sich der Prüfling nämlich leisten, die eine oder andere Definition oder einen ganz bestimmten Begriff gerade nicht parat zu haben, sofern er fähig ist, mit der eigenen Literatur zu arbeiten. Er kann sich sogar leisten können, eine bestimmte Randbedingung zu einem bestimmten Verfahren nicht genau formulieren zu können, sofern er ihre Existenz kennt. Er wird sich nicht leisten können, die Zusammenhänge und Methoden erst in der Prüfung zu verstehen, denn für ein ad-hoc-Studium der Literatur während einer 45-minütigen bleibt viel zu wenig Zeit. Ist dem Prüfling die genaue Ausgestaltung einer Stufe eines

mehrstufigen Verfahrens während der Prüfung nicht mehr klar, dann würde er ohne Zulassung beliebiger Literatur für die folgenden Stufen nicht zeigen können, dass er diese sehr wohl beherrscht. Zulassung beliebiger Literatur nähert somit die Prüfungsumgebung einer realeren Umgebung wissenschaftlichen Arbeitens an, in der schließlich auch nachgeschlagen werden kann, sofern man im Zeitlimit bleibt.

- Gutes Beispiel: Die Numerik-Klausuren (2002..2004) von Peter Kunkel.
- Gutes Beispiel: Die Logik-Klausur (2003.SS) von Gerhard Brewka.

4.3 Ablauf

4.3.1 **Fordern Sie in der Klausur-Aufgabenstellung nur die Matrikelnummer des Prüflings, nicht den Namen.**

Wenn Sie den Namen des Prüflings fordern, dann verstoßen Sie womöglich gegen bestehende Datenschutzregelungen und können so die Prüfung aus formalen Gründen anfechtbar sein lassen. (Mehr dazu: <http://fsinf.informatik.uni-leipzig.de/pmwiki/pmwiki.php/Main/DatenschutzHinweise>)

Selbstverständlich können Sie aber während der Prüfung anhand eines Studentenausweises die Zugelassenheit und Anwesenheit der Prüflinge überprüfen bzw. überprüfen lassen.

4.3.2 **Korrigieren Sie die Klausuren schnell, zumindest diese, bei denen Sie extra um schnelle Korrektur gebeten werden.**

Abgesehen davon, dass den Studenten die Prüfungsergebnisse natürlich oft auf den Nägeln brennen, sind einige Prüfungsergebnisse ausschlaggebend dafür, ob jemand Anschlussfinanzierung (BaFöG, ...) erhält, weiterstudiert oder nicht. Davon wieder hängen eine Reihe von persönlichen Entscheidungen (wie Umzüge usw.) ab, die möglichst früh zu treffen sind.

- Gutes Beispiel: Die Korrektur der Klausur „Digitale Informationsverarbeitung | Programmierung und Programmiersprachen“ (2003 Sommersemester) durch den Lehrstuhl für angewandte Telematik | e-Business wurde gleich am Tag nach der Klausur angefangen und beendet. Man konnte übrigens den Verlauf der Korrektur und das Wachstum seiner persönliche Punktzahl live per Web-Cam verfolgen.
- Schlechtes Beispiel: Die Korrektur der Klausur „Betriebssysteme | Kommunikationssysteme“ durch den Lehrstuhl Rechnernetze und Verteilte Systeme brauchte sowohl am Ende des Sommersemesters 2003 als auch am Ende des Wintersemesters 2003 mehr als 9 Wochen.

4.3.3 **Wenn Sie mündliche Prüfungen „nach Vereinbarung“ anbieten, dann bieten Sie einen Termin an, der mindestens 4 Wochen nach der Vorlesungszeit liegt.**

Direkt nach der Vorlesungszeit ist etwa 3 Wochen lang Prüfungssaison. Das heißt, dass auf eine Prüfung die andere folgt. Sie tun vielen Studenten einen Gefallen, wenn Sie den Studenten zwei oder drei weit voneinander entfernte Tage anbieten, zu denen sie sich einen Prüfungstermin aussuchen können. Dann passt dies besser zur Prüfungssaison, zu absolvierenden Praktika und der persönlichen Lebensplanung. Liegen die Prüfungen zu dicht, dann sinkt

die Motivation und wahrscheinlich auch die Leistungsfähigkeit des Studenten.

4.4 Bewertung

4.4.1 Erklären Sie noch in der Vorlesung, nach welchen Kriterien die studentischen Leistungen bewertet werden.

Beispielsweise lässt sich eine Rechnung nach einer beliebigen Teilmenge folgender Kriterien bewerten:

- Es gibt Punkte für den Ansatz.
- Es gibt Punkte für den eingeschlagenen Rechenweg bzw. absolvierte Teil-Etappen des Rechenwegs.
- Es gibt Punkte dafür, verwendete Sätze ausdrücklich zu nennen.
- Es gibt Punkte für die Richtigkeit der Rechnungen im Rechenweg.
- Es gibt Punkte für das richtige Ergebnis.
- Es gibt Punkte für die „Sauberkeit“ (z.B. Einheiten durchschleifen)
- Es gibt Punkte für richtiges Weiterrechnen trotz eines Fehlers im Rechenweg („Folgefehlerpunkte“)

Je nachdem, wie Ihre Kriterien aussehen, sind unterschiedliche Strategien erfolgversprechend:

- Gibt es Punkte auf Ansatz und Rechenweg-Etappen, so sollte jede Aufgabe der Klausur „ein wenig bearbeitet“ werden (Breitensuche).
- Gibt es Punkte nur auf das richtige Ergebnis, dann sollten lieber wenige Aufgaben der Klausur bearbeitet werden, dafür aber vollständig.

Studenten, die mangels Informationen über Ihre Bewertungskriterien die falsche Strategie (insbesondere unter dem Zeitdruck einer Klausur) einschlagen, erscheinen dann in ihrer inhaltlichen Leistung möglicherweise schlechter, als sie tatsächlich sind. Dies ist nicht notwendig.

4.4.2 Informieren Sie sich, aus welchem Studiengang ihre Kandidaten stammen und welche Art von Leistungsnachweis sie benötigen.

Beispielsweise sind viele Scheinklausuren des Diplom-Informatik-Grundstudiums Prüfungsklausuren des Bachelor-Informatik-Grundstudiums. Scheinklausuren müssen nur mit „bestanden“ | „nicht bestanden“ bewertet, Prüfungsklausuren dagegen benotet werden.

4.4.3 Bieten Sie jedem Studenten an, seine Leistungen in Klausuren und Testaten zu benoten, ohne dass der Student dies begründen muss.

Auch wenn viele Diplom-Informatik-Studenten nur Übungsscheine brauchen, sollten ihre Leistungen benotet werden. Denn es kommt vor, dass die Noten doch gebraucht werden, zum Beispiel nach einem Studiengangwechsel. Falls es keine Schwierigkeit macht, eine Note zu finden (z.B. weil sie sowieso von der erreichten Punktzahl der Lösungen von Übungsserien abhängt und diese Punktzahl sowieso ermittelt werden), empfiehlt es sich, die Note auch ohne ausdrückliche Bitte zu vermerken.

4.4.4 Veröffentlichen Sie Klausurergebnisse nie mit Namen der Prüflinge.

Es kommt immer wieder vor, dass Ergebnisse von Klausuren mit dem Namen

der Prüflinge veröffentlicht werden. Die Noten sind persönliche Daten, ähnlich sensibel wie Ihr Kontoauszug. Sie dürfen nur dem Prüfling selbst bekannt gegeben werden. Als Kompromiss zwischen Einfachheit der Bekanntgabe und Privatsphäre sollten Klausurergebnisse nur unter Angabe der Matrikelnummer veröffentlicht werden. Dieses Verfahren hat auch Mängel:

- Aus der Matrikelnummer lässt sich das Immatrikulations-Semester eines jeden Kandidaten ableiten. Wenn nur ein einziger Student A eines Immatrikulations-Semesters X die Prüfung schreibt, dann ist klar, dass der Student zur Note dieses Immatrikulations-Semesters X eben dieser Student A ist.
- Durch „Data-Mining“ über mehrere Prüfungen hinweg lässt sich zusammen mit dem Wissen, wer bei welcher Prüfung teilgenommen hat, erstaunlich genau herausfinden, welcher Student welche Note in jeder Prüfung hat.

Wesentlich mangelärmer wäre zum Beispiel

- die Vergabe von einmaligen Prüfungsnummern für jede Kombination aus Prüfling und Prüfung oder
- der Versand der Prüfungsergebnisse per e-Mail.

Beide Verfahren werden in der Regel berechtigterweise wegen dem erhöhten Verwaltungsaufwand als nicht praktikabel angesehen. Wenn Sie dies anders sehen, dann nutzen Sie bitte ein solches mangelarmes Verfahren.

4.4.5 Veröffentlichen Sie die Klausurergebnisse im Internet.

Viele Studenten sind insbesondere in der vorlesungsfreien Zeit nicht regelmäßig in der Nähe des Ortes, wo die Prüfungsergebnisse aushängen. Aus diesem Grund sollten Sie - insbesondere bei Informatikern - die Ergebnisse im Internet „aushängen“. Die Ergebnisse sollten auf der Homepage der Veranstaltung (wo auch Folien und Skripte zu finden sein sollten) stehen oder von dort verlinkt sein. So lange die Ergebnisse noch nicht verfügbar sind, erklären Sie auf der Homepage, ob die Ergebnisse online verfügbar sein werden, und wenn ja, wann ungefähr:

4.4.6 Geben Sie eine Schätzung ab, wann die Klausur bewertet sein wird.

Dadurch wird der Aufwand der Studenten minimiert, ständig nach Klausurergebnissen zu schauen. Gerade bei den Klausuren am Ende des Sommersemesters kann sich die Bewertung über mehrere Monate hinziehen, da Dozenten zum Teil im Urlaub sind. Dies ist schön, aber wenn der Korrekteur weiß, dass er erst nach dem Urlaub fertig sein wird, dann sollte er auch auf die Homepage schreiben, dass dies so ist. Sonst entsteht nur unnötige Ungeduld und Planungsunsicherheit dadurch, dass man falsche Zeitpunkte für das Datum der Veröffentlichung der Klausurergebnisse annimmt.

4.4.7 Benachrichtigen Sie über die Verfügbarkeit von Klausurergebnissen über ein Push-Medium.

Das heißt: Statt hunderte Studenten ständig überprüfen, ob Ergebnisse da sind („polling“), schicken Sie ein einziges Mal eine Nachricht an alle, dass die Bewertung abgeschlossen ist, zum Beispiel über eine Mailing-Liste. Da es

speziell dafür derzeit keine Mailing-Liste gibt, wäre die allgemeine Mailing-Liste für Informatik-Studenten diskus@fsinf.informatik.uni-leipzig.de zu empfehlen.

4.4.8 **Veröffentlichen Sie den Bewertungsmaßstab.**

In den allermeisten Fällen werden Sie eine Klausur nach einzelnen Punkten für einzelne Teil-Aufgaben bewerten. Spätestens bei der Veröffentlichung haben Sie für sich selbst einen einheitlichen Maßstab, das heißt eine Zuordnung von Punkten zu Noten, überlegt. Veröffentlichen Sie diesen, damit der Prüfling klar wissen kann, ob seine Note den vergebenen Punkten nach berechtigt ist.

4.5 **Klausureinsicht**

4.5.1 **Bieten Sie ausdrücklich zeitlich unbegrenzte Einsicht in Klausuren an.**

Dies nicht zu tun könnte Sie unter Umständen unnötig Sympathiepunkte kosten, da Studenten glauben könnten, Sie würden möglicher Kritik aus dem Weg gehen wollen. Die Einsicht selbst übrigens müssen Sie nicht selbst durchführen, es reicht vollkommen aus, wenn dies ein Mitarbeiter macht.

- Gutes Beispiel: Bernd Schulze bietet ausdrücklich an, dass man zur Einsicht der Numerik-Klausur (2002 Wintersemester) ihn einfach nur in seinem Büro besuchen möge.

4.5.2

Nachwort

Sind Personen nicht mit ihrer Nennung in diesem Dokument einverstanden, so mögen Sie dies dem aktuellen „Pfleger“ („maintainer“) dieses Dokuments mitteilen.

aktueller Dokumenten-Pfleger

Xuân Baldauf <teaching-guide--2004.04.22@studium.baldauf.org>

	URL
Dieses Dokument stand ursprünglich unter	http://studium.baldauf.org/informatik/2004.SS/misc/teaching%20guide/teaching%20guide.pdf
und basiert auf dem Original	http://studium.baldauf.org/informatik/2004.SS/misc/teaching%20guide/teaching%20guide.pdf
Die aktuelle Version ist verfügbar unter	http://studium.baldauf.org/informatik/2004.SS/misc/teaching%20guide/teaching%20guide.pdf

Modifikationsgeschichte

Datum	Name	e-Mail-Adresse
22.04.2004	Xuân Baldauf	teaching-guide--2004.04.22@studium.baldauf.org

anstehende Änderungen

Vor einem „Release“ sollten nach Matthias Quasthoff die Aufforderungen umformuliert werden, damit sich der angesprochene Leser nicht zum Teil pikiert fühlt.