Modelling Examination Regulations in OWL

PROJECT THESIS

Alexander Pohl

supervised by
PROF. DR. FRANÇOIS BRY
DR. NORBERT EISINGER

Contents

1 Introduction **6**
1.1 Motivation . . . 6
1.2 Structure of this Thesis . . . 7

2 A Short Introduction to OWL and Description Logics **7**

3 Modelling the Examination Regulations **11**
3.1 Some Notes on the Examination Regulations . . . 11
3.2 Terminology of the Examination Regulations . . . 11
3.3 Model Requirements . . . 11
3.3.1 Questions Concering the Abilities of a Query System . . . 12
3.4 The Basic Model of the Examination Regulations as a Diagram . . . 14
3.4.1 Elements of the Diagram . . . 14
3.4.2 The Entities and their Relationships . . . 15
3.5 Refining the Entities . . . 15
3.5.1 Modelling a Semester . . . 16
3.5.2 Modelling an Examination . . . 16
3.5.3 The Hierarchy of Examinations . . . 17
3.5.4 Modelling an Examination Result . . . 19
3.5.5 Modelling the Credit Accounts . . . 20
3.5.6 Modelling a Student . . . 22
3.5.7 The Hierarchy of Students . . . 22
3.6 The Questions in the Terminology of the Model . . . 24
3.7 Open World Assumption and Closed World Assumption in OWL . . . 25
3.8 Notes on the Software that is Used for this Project Thesis . . . 25

4 Proposals to Improve OWL **26**
4.1 Customised Datatypes in OWL . . . 26
4.2 Arithmetics in OWL . . . 27
4.3 Finite Quantification . . . 28
4.4 Qualified Number Restrictions and Range Restrictions of Roles . . . 31
4.5 Using SWRL . . . 33

5 Summary **36**

6 Appendix A: The Examination Regulations for this Project Thesis **38**
6.1 Relevant Aspects of the Examination Regulations . . . 38
6.1.1 Acquiring Credits . . . 38
6.1.2 The Structure of the Module-Catalogue A . . . 39
6.1.3 The Structure of the Module-Catalogue B . . . 39
6.1.4 The Structure of the Module-Catalogue C . . . 42
6.1.5 The Structure of the Module-Catalogue D . . . 42
6.1.6 The Structure of the Module-Catalogue E . . . 42
6.1.7 Further Remarks on the Examination Regulations . . . 42
6.2 Recommended Generic Study Schedule . . . 43
6.3 Compatibility Constraints . . . 44

7 Appendix B: The Knowledge Base for the Examination Regulations **45**
7.1 The Concept Semester and the Roles Precedes and Dates-Result . . . 45
7.2 The Credit Accounts . . . 46
7.3 The Concept of Exam Results . . . 47
7.4 The Concept of Exams . . . 47
7.5 The Concepts of Students and their Subsumption Hierarchy . . . 49

8 Appendix C: Formalising a Concrete Study Schedule **52**

additional documents:

- The Generic Examination Regulations for Bachelor Study Programs of the LMU München (in German)
- The Examination Regulations for the Bachelor Study Program for Computer Science of the TU München (in German)

1 Introduction

This project thesis identifies an interesting class of Semantic Web applications called "contracts" and argues that examination regulations of study programs belong to that class of applications. In view of this observation it makes sense to investigate to which extent "contract" applications can be modelled with standard Semantic Web tools and in particular with OWL. This thesis takes examination regulations as a representative "contract" application and develops an OWL model for it, on which a query answering system could be based. As it turns out, the modelling is hampered by several shortcomings of OWL. Therefore the thesis proposes improvements of OWL to overcome these shortcomings.

1.1 Motivation

What will be called "contract" applications is best illustrated by a few examples. These applications have in common that they are based on certain "rules of the game".
As a first example consider a new dimension of searching the web where queries can be formulated, like "Give me those credit-institutes in Bavaria that offer the lowest rate of interest", "Search for those universities that enables me to finish computer science within eight semesters", or "I'm looking for book stores that offer books about computer science. They shall accept on-account payment and offer me a right of return for 31 days". That is, search engines could search the web according to criteria for which pure document processing is not sufficient anymore. This information needs to be semantically filtered out. All these queries focus on conditions or arrangements that are defined in contracts the user completes when he makes use of the offers. To filter this information the search engine must be aware of the elements of the contracts. Understanding and identifying the elements of a contract is a crucial task in "contract" applications.
Another example in the field of e-commerce could be automatic and autonomous trading. Imagine a company that produces machines and constantly needs huge amounts of screws, nuts and bolts so that it has to trade with suppliers every day. Instead of trading via meetings with the persons in charge, the companies could use software agents to trade online. These software agents would be able to autonomously negotiate and complete contracts. The companies would just tell their agents what conditions are important (i.e. range of prices, delivery dates or quantity discounts) and the rest up to the delivery of the products would be done via software autonomously. In order to realize such automatic trading the software agents would need to be aware of the issues they are negotiating about.
As another example, consider a "generic" tax program one can file his tax returns with. This program could load the current tax laws from the website of the local tax office and interpret them in order to find out how the user can maximise his tax refunds. These programs would always be up-to-date since they only need to download the tax laws from the tax office's website, whenever they changed.
Another example would be query systems for freshmen at universities. Consider a software tutor that could help freshmen to come along with problems regarding their studies and would be able to answer their questions. These programs would therefore need to be able to interpret the universities' examination regulations. It should be mentioned that even for students as well as for the academic staff of computer science such a formal version of the examination regulations, for example in OWL, is likely to be more useful than the examination regulations themselves...
All these example applications are different and so are the contracts. While some might include abstract and simple conditions, others might be more complex and contain concrete quantities and mathematical or technical interrelations. However, all these contracts have in common that they define conditions under which two or more parties come to arrangements. These arrangements could be about the exchange of goods (e.g. books for money) or about services between these parties. So for the purpose of this thesis, we will consider "contracts" as "rules of the game" and omit legal issues and consequences of contracts. From this point

of view, we can speak of a contract between a student and his university where the university offers the service of teaching only under certain conditions that must be satisified by the student.
If OWL is appropriate to formally represent contracts is to be examined in this thesis. The example of examination regulations is thereby chosen as the modelling topic of this thesis for several reasons. Firstly, examination regulations can be seen as a small contract between the university and the students, as mentioned above. Secondly, examination regulations are usually simple and concise, so that the modelling expense is limited. Thirdly, the reader might work in the field of computer science and hence studied or has been studying computer sciences or a related subject. So, he might just need a short training period to deal with the modelling of the examination regulations. These facts make examination regulations an appropriate representative for this project thesis.

1.2 Structure of this Thesis

At the beginning of this thesis a short introduction to OWL will be given. Then, a diagrammatic model similar to an entity-relationship-model will be developed. This diagrammatic model will be specified in OWL. As a main concern of this thesis we will propose improvals of OWL, according to the examination regulations, in order to make OWL more convenient to use and thus widening the fields of application. Furthermore, we will discuss in what way the modeller might take advantage of the improvals and what these improvals mean to the reasoning. The examination regulations that were built for this thesis, the original examination regulations of the TU München and the LMU München as well as the model specified in OWL can be found as attachements of this thesis.

2 A Short Introduction to OWL and Description Logics

In this section a short and introductory summary of OWL and its underlying description logics is given. For a complete and more detailed introduction refer to [10] and [7].
The *Web Ontology Language*, OWL for short, is an object-oriented data model language and the current W3C standard for the semantic web. In OWL one defines classes, properties and instances. The classes structure the set of instances according to aspects that are significant for the application. To relate instances to instances or instances to data values, (*instance-valued* and *data-valued*) properties are used. Classes as well as properties can be structured in subclass-hierarchies or subproperty-hierarchies, respectively. These hierarchies, among other things, are specified by *axioms*. The definitions of classes, properties and their hierarchies as well as (class or property) instances and axioms build up an ontology. OWL provides an abstract (declarative) syntax as well as a concrete (XML based) syntax (cf. [10], [7]). OWL ontologies can syntactically be translated into description logics (DL), which can be seen as an object-oriented family of logics. By translating OWL to description logics we obtain a semantics for OWL that is based on sets. The terminology thereby changes. We then denote an OWL class as a *concept*, an OWL property as a *role* and instances as *individuals*. Finally, an OWL ontology is called a *knowledge base* in the DL terminology. For this project thesis it will be sufficient to restrict the treatment of OWL to its corresponding DL syntax, which is as follows.

$$
\begin{aligned}
\mathsf{Concept} &::= \top \mid \bot \mid \mathsf{CName} \mid \neg\mathsf{Concept} \mid \mathsf{Concept} \sqcap \mathsf{Concept} \mid \mathsf{Concept} \sqcup \mathsf{Concept} \mid \\
&\quad \{\mathsf{o}\} \mid \exists \mathsf{Role}.\mathsf{Concept} \mid \forall \mathsf{Role}.\mathsf{Concept} \mid \geq \mathsf{m}\,\mathsf{Role} \mid \\
&\quad \leq \mathsf{m}\,\mathsf{Role} \mid \geq \mathsf{m}\,\mathsf{Attribute} \mid \leq \mathsf{m}\,\mathsf{Attribute} \\
\mathsf{Axiom} &::= \mathsf{Concept} \sqsubseteq \mathsf{Concept} \mid \mathsf{Concept} \equiv \mathsf{Concept} \\
\mathsf{Role} &::= \mathsf{RName} \mid \mathsf{Role}^{-} \\
\mathsf{Attribute} &::= \mathsf{AName}
\end{aligned}
$$

A DL interpretation is a tuple $\mathcal{I} = (\cdot^{\mathcal{I}}, \cdot^{\mathcal{D}}, \Delta^{\mathcal{I}}, \Delta_{\mathcal{D}})$, where $\Delta^{\mathcal{I}}$ is a non-empty set and called the domain of the individuals, $\Delta_{\mathcal{D}}$ is a non-empty set of data values, $\cdot^{\mathcal{I}}$ is a function that maps

- a concept name CName to a subset of $\Delta^{\mathcal{I}}$
- a role name RName denoting an individual-valued property to a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
- an attribute name AName denoting a data-valued property to a subset of $\Delta^{\mathcal{I}} \times \Delta_{\mathcal{D}}$
- an individual name a to an element $\in \Delta^{\mathcal{I}}$.

The function $\cdot^{\mathcal{D}}$ thereby interprets datatypes. Given an interpretation $\mathcal{I}$ a model (or set) theoretic semantics of the DL formulae can be given as follows.

- $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$, $\bot^{\mathcal{I}} = \emptyset$, $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and $(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \backslash C^{\mathcal{I}}$
- $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
- $(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$
- $(\{o\})^{\mathcal{I}} = o^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
- $(\exists R.C)^{\mathcal{I}} = \{x \mid \exists\, y.\, (x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
- $(\forall R.C)^{\mathcal{I}} = \{x \mid \forall\, y.\, (x,y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$
- $(\geq\, m\, R)^{\mathcal{I}} = \{x \mid \#\{y \mid (x,y) \in R^{\mathcal{I}}\} \geq m\}$,
- $(\leq\, m\, R)^{\mathcal{I}} = \{x \mid \#\{y \mid (x,y) \in R^{\mathcal{I}}\} \leq m\}$
- $(\geq\, m\, T)^{\mathcal{I}} = \{x \mid \#\{y \mid (x,y) \in T^{\mathcal{I}}\} \geq m\}$,
- $(\leq\, m\, T)^{\mathcal{I}} = \{x \mid \#\{y \mid (x,y) \in T^{\mathcal{I}}\} \leq m\}$
- $(R^{-})^{\mathcal{I}} = \{y \mid (x,y) \in R^{\mathcal{I}}\}$
- $(C \sqsubseteq D)^{\mathcal{I}} = C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
- $(C \equiv D)^{\mathcal{I}} = C^{\mathcal{I}} \subseteq D^{\mathcal{I}} \wedge D^{\mathcal{I}} \subseteq C^{\mathcal{I}}$.

C and D denote concepts, R denotes an individual-valued property and T denotes a data-valued property.
For this project thesis a class is considered to be the same as a concept. The same applies to property and role as well as to instance and individual. For abbreviation purposes (especially for the modelling section) we will also use *role* to denote an individual-valued property. For a data-valued property we will also use the term *attribute*.

The elements of the declarative syntax of OWL and their counterparts in DL syntax is thus given as figure 2.1 shows.

The abstract syntax for the set of OWL axioms and their corresponding DL syntax is given in figure 2.2.

The first two axioms state that a class A is a subset of each of the classes $C_1, \ldots, C_n$. With the OWL construct EnumeratedClass a class can be defined by explicitly listing its instances. We also discuss enumerated classes in section 4.3. As the unique name assumption is not valid per se it must be explicitly stated that individuals and concepts of different names are different, in fact. This is done with the axioms SameIndividuals and EquivalentClasses. Similarly

OWL element	corresp. DL syntax	short notice
Class(A)	A	A concept
Class(owl : Thing)	$\top$	
Class(owl : Nothing)	$\bot$	
intersectionOf($C_1, C_2, \dots$)	$C_1 \sqcap C_2 \sqcap \dots$	$C_1, C_2, \dots$ concepts
unionOf($C_1, C_2, \dots$)	$C_1 \sqcup C_2 \sqcup \dots$	$C_1, C_2, \dots$ concepts
complementOf(C)	$\neg C$	C concept
oneOf($o_1, o_2, \dots$)	$\{o_1\} \sqcup \{o_1\} \sqcup \dots$	$o_1, o_2, \dots$ individuals
restriction(R someValuesFrom(C))	$\exists R.C$	R role, C concept
restriction(R allValuesFrom(C))	$\forall R.C$	R role, C concept
restriction(R hasValue(o))	$\exists R.\{o\}$	R role, o individual
restriction(R minCardinality(m))	$\geq m\, R$	R role
restriction(R maxCardinality(m))	$\leq m\, R$	R role
restriction(A someValuesFrom(T))	$\exists A.T$	A attribute, T datatype
restriction(A allValuesFrom(T))	$\forall A.T$	A attribute, T datatype
restriction(A hasValue(m))	$\exists A.\{o\}$	A attribute, m data value
restriction(A minCardinality(m))	$\geq m\, A$	A attribute
restriction(A maxCardinality(m))	$\leq m\, A$	A attribute
ObjectProperty(R)	R	R individual-valued property
S′ inverseOf(S))	S^-	S individual-valued property
DatatypeProperty(T)	T	T datatype-valued property

Figure 2.1: OWL elements and their corresponding DL elements

OWL axiom	corresp. DL syntax
Class(A partial $C_1 \dots C_n$)	$A \sqsubseteq C_1 \sqcap \dots \sqcap C_n$
Class(A complete $C_1 \dots C_n$)	$A \equiv C_1 \sqcap \dots \sqcap C_n$
EnumeratedClass(A $o_1 \dots o_n$)	$A \equiv \{o_1\} \sqcap \dots \sqcap \{o_n\}$
SubClassOf(C_1, C_2)	$C_1 \sqsubseteq C_2$
EquivalentClasses($C_1, \dots, C_n$)	$C_1 \equiv \dots \equiv C_n$
DisjointClasses($C_1, \dots, C_n$)	$C_i \sqsubseteq \neg C_j, 1 \leq i < j \leq n$
SubPropertyOf(R_1, R_2)	$R_1 \sqsubseteq R_2$
EquivalentProperties($R_1, \dots, R_n$)	$R_1 \equiv \dots \equiv R_n$
ObjectProperty(R	
super(R_1) ... super(R_n)	$R \sqsubseteq R_i$
domain(C_1) ... domain(C_k)	$\geq 1\, R \sqsubseteq C_i$
range(C_1) ... range(C_k)	$\top \sqsubseteq \forall R.C_i$
[Symmetric]	$R \equiv R^-$
[Transitive]	Trans(R)
[Functional]	Func(R)
[InverseFunctional])	Func(R^-)
Individual(o	
type(C_1) ... type(C_n)	$o : C_i, 1 \leq i \leq n$
value(R_1, o_1) ... value(R_n, o_n))	$(o, o_i) : R_i, 1 \leq i \leq n$
SameIndividuals($o_1 \dots o_n$)	$o_1 = \dots = o_n$
DifferentIndividuals($o_1 \dots o_n$)	$o_i \neq o_j, 1 \leq i < j \leq n$

Figure 2.2: OWL axioms translated into DL syntax

to this axiom with DifferentIndividual and DisjointClasses it is stated that the listed individuals differ and concepts are disjoint. With the axiom Individual an instance o belonging to the classes $C_1, \ldots, C_n$ is defined. In this axiom it is also stated to which instances and values the individual are related to. Concept and property hierarchies are defined with SubClassOf and SubPropertyOf. With ObjectProperty a new individual-valued property is defined.

3 Modelling the Examination Regulations

In this section the basic model of the examination regulations for this project thesis is introduced. A diagrammatic model similar to an entity-relationship-model will be provided. According to this model an ontology can be developed.

3.1 Some Notes on the Examination Regulations

The examination regulations for this project thesis are based on the examination regulations for the bachelor's program of computer science at the Technical University of Munich (TUM). These regulations were chosen because they are kept very simple and concise. They are also of regional significance because, unfortunately, there are currently no examination regulations for the future bachelor program of computer sciences at the Ludwig-Maximilians-University of Munich (LMU) available but guidelines for bachelor's programs at the LMU in general. As the examination regulations of the TUM do not completely fit into the guidelines for bachelor's programs of the LMU some modifications are introduced. The modified examination regulations can be found in section 6.

3.2 Terminology of the Examination Regulations

Before coping with the modelling of the examination regulations it is necessary to introduce domain specific terms and to define their meanings because the reader might not know some terms or he might use some terms in a different way than they are used in this project thesis.
Firstly, a student is enrolled not only in a specific study program but also in a specific semester[1]. As a consequence the student gets enrolled in each semester. The courses the student attends are denoted as *modules* (since a module comprises not only one or more lectures but also its exercise or practical courses). These modules are organised in *module-catalogues*, for example modules of the module-catalogues "Computer Science", "Mathematics", "Physics" and "Economics". These module-catalogues in turn can be subdivided into so-called *module-groups* that aggregate modules according to their common topics like "Database and Information Systems" or "Artificial Intelligence and Robotics". The modules can additionally be distinguished according to the fact whether they must be attended by the student or not. Modules that must be attended by the student are referred to as *obligatory modules*. Modules of a certain module-catalogue from which the student can choose but has to attend a certain amount of these modules are referred to as *mandatory modules*. Modules that are free to attend are referred to as *elective modules*. There are also modules that must be attended within the first two semesters. They will be referred to as *basic modules*. A module is considered as attended or passed if the student has passed its corresponding exam, otherwise it is considered as not attended or failed. An exam is passed if its grade is between one (excellent) and four (poor). So, with passing an exam the student proves his attendance of the corresponding module. In passing exams the student acquires *bonus credits*, in failing re-examinations he acquires *malus credits*. The acquired credits are administered in a *bonus credit account* and a *malus credit account*, respectively. The acquired credits play an important role in the bachelor's examination (bonus credits) and in the admittance to second re-examinations (malus credits).

3.3 Model Requirements

Certain requirements must be satisfied by the model. For example it must be appropriate to support the basic reasoning services of OWL, e.g. subsumption and classification of individuals.
The main idea of the modelling is thus to classifiy a student according to classes and their hierarchies that are defined by the examination regulations. Therefore, further classes of entities

[1] Such an interpretation of *enrollment* also includes the meaning of the German term "Studentische Rückmeldung"

and their hierarchies, e.g. examination classes and credit accounts are necessary. Additionally, the model should be extensible in order to support the answering of advanced questions concerning a variety of topics that are listed below. Some exemplary questions are also given. Most of these questions can only be proceesed by reasoners that generate models and check certain conditions automatically. Although the treatment of such questions is out of scope of this project thesis, we want to consider in the following section what these questions might be.

3.3.1 Questions Concering the Abilities of a Query System

In this section we want to determine the types of questions a possible query system should be able to answer. These types in turn can be distinguished between those on the level of the examination regulations and those on the level of a concrete study schedule.

Questions on the Level of the Examination Regulations
The first type comprises questions regarding consistency. Some examples are given as follows.

- Are all options offered by the examination regulations possible?
- Does the maximal duration of study (including all possible re-examinations) lie within given time limits?
- Are there any regulations that contradict each other?

The second type comprises questions regarding time. Here are some example questions given one may want to have answered by a query system.

- Do the different possible study schedules have unequal durations?
- What is the longest/shortest duration of study?
- Is it possible to acquire the average amount of credits (30 cr) in each semester?
- When should the student decide at the latest/earliest, what mandatory modules of the module-catalogue B he wants to attend?
- Which mandatory modules of the module-catalogue B should be chosen to study as fast as possible?

The last type mentioned here comprises questions regarding consequences. Some example questions are the following.

- Do exceptions have unintended side-effects?
- Does the duration of study vary when placing modules from one semester into another?
- Does a schedule of a basic study period exist, in which it is impossible to choose any elective module from the module-catalogue B due to prerequisites (cf. section 6)?
- Does a schedule of a basic study period exist in which the student does not need to attend any practical?
- How many module examinations can the student repeat once/twice until his malus credits amount to 90 cr? From which semester on may the student repeat exams only once then?
- Suppose a student has acquired exactly 54 cr within the first four semesters. Does that mean that he is forced in any case to study nine semesters for finishing his studies?

Questions on the Level of a Concrete Study Schedule
Suppose a student Hans who has been already studying for three semesters. He has passed the following module exams.

1st semester	Foundations of Computer Science I Technical Foundations of Computer Science Discrete Mathematics Linear Algebra *acquired credits: 36*
2nd semester	Foundations of Computer Science II Introduction to Software Engineering Foundations of Algorithms and Data Structures Analysis Computer Science Law *acquired credits: 33, total: 69*

Hans doesn't want to attend any modules other than he has to. He is planning to attend modules of the module-groups "Artificial Intelligence and Robotics" and/or "Databases and Informations Systems". Hans is wondering:

- Is it necessary to acquire credits in modules of only one module-catalogue?
- May passing any exams except for the basic modules be postponed until the 6th semester?
- Which modules have to be passed in the 2nd and 3rd semester to be admitted to the modules of the module-catalogue B already in the 4th semester?
- Which modules of two module-groups allow a student to acquire the necessary credits as fast as possible?
- Hans is suddenly unhappy with his current choice. On the one hand he wonders, which other module-group must be selected to finish his studies as quickly as possible.
- On the other hand he wants to attend as many additional modules from the module-catalogues B and D as possible and so he wants to determine those module-groups with the most modules of the lowest duration.
- Which modules can already be attended in the 4th semester when studying according to the recommended study schedule?

Questions Concerning the Implementation of a Model
When implementing the model of the examination regulations we may ask the following questions regarding logics.

- Is the negation used in the examination regulations monotonic?
- Is the inclusive-or used in the examination regulations consistent (i.e. A or B or A and B together)?
- How can the two sets of modules $\{A, B, C\}$ and $\{C, D, E, G\}$ be expressed as equivalent (for instance, with respect to the minimum of credits to be acquired)?
- Are there any formulae that make sense for the modelling or for querying but cannot be expressed?

3.4 The Basic Model of the Examination Regulations as a Diagram

In this section a basic model similar to an entity-relationship-model of the examination regulations is developed. From a modelling point of view it is more sensible to distinguish between relationships among entities and attributes of entities despite the fact that OWL makes no difference between relationships and attributes. In this section entities and their relationships are modelled. The model will be refined with attributes later.
Besides the regulation of administrative concerns the examination regulations mention the following entities.

- Account for bonus credits
- Account for malus credits
- Examination
- Examination result
- Semester
- Student

3.4.1 Elements of the Diagram

The diagram consists of entities and their relationships. Also hierarchies of entities and relationships are shown. An entity is represented as an ellipse that surrounds the name of the entity. A relationship connects an entity (domain-entity) to another (or the same) entity (range-entity). Such a relationship will be represented as a rectangle that surrounds the name of the relationship. Hierarchies are represented by arrows where their triangular arrowheads point to the super-entities (super-relationships, resp.) and their non-filled circles "point" to the sub-entities (sub-relationships, resp.). A relationship can also relate a domain-entity to more than one range-entity of different types. This will be represented by multiple arrows that point from the relationship to each type of range-entity.

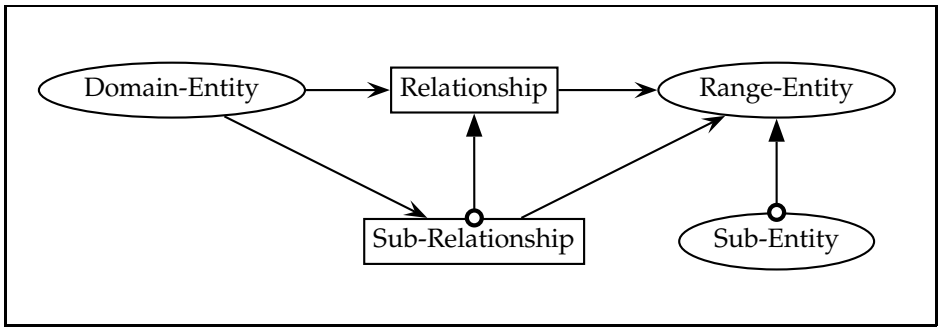


Figure 3.3: The elements of a diagram. The ellipses represent entities, the rectangles represent relationships where the arrows' directions lead from the domain to the relationship and from the relationship to the range. Hierarchies are represented by arrows with a triangular arrowhead pointing to the super-entity (super-relationship, resp.) and a non-filled circle pointing to the sub-entity (sub-relationship, resp.).

The usage of these diagram elements aims at a more concise and simple diagrammatic presentation of the model. The arrows that connect entities with relationships make clear which entities are related to each other. Thus, a labelling of the arrows, for example, is not necessary.

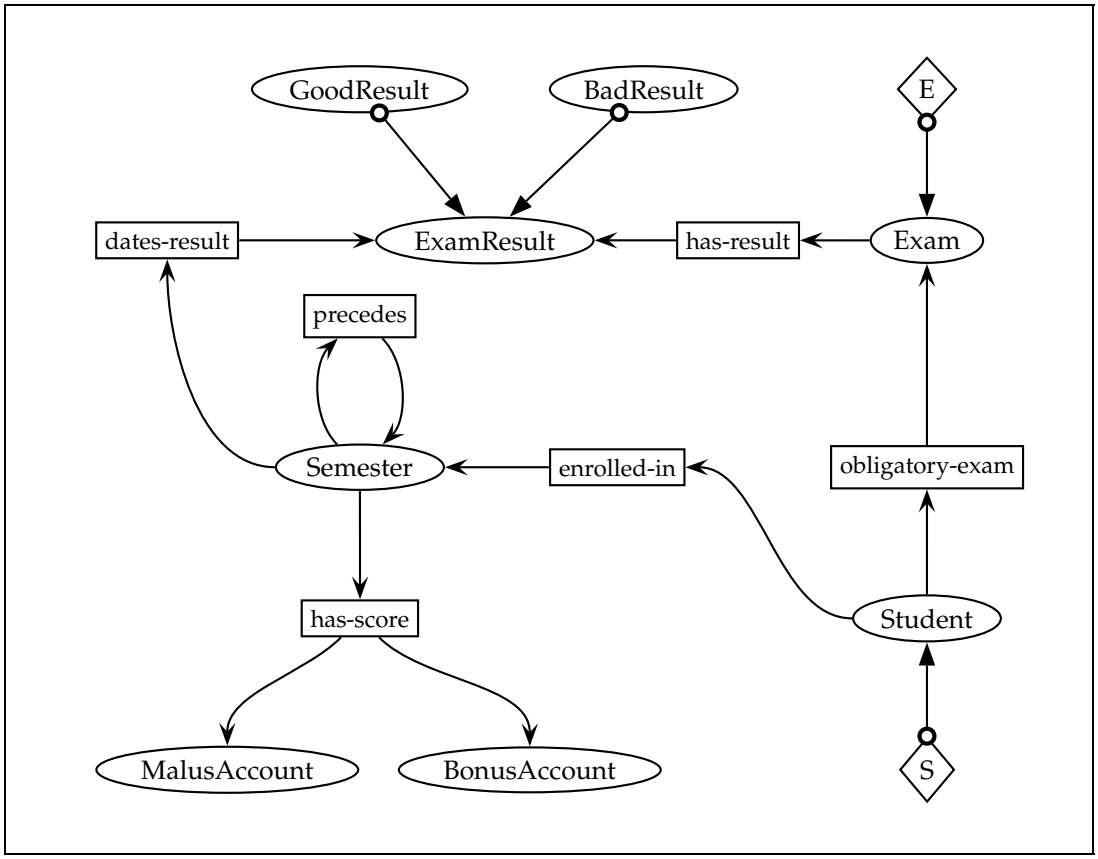


Figure 3.4: The diagram gives an overview of the entities and their relationships. For concisement matter attributes and sub-entities are omitted. The hierarchies of the entities Student and Exam are abbreviated as diamonds. They will be shown in detail later.

3.4.2 The Entities and their Relationships

In figure 3.4 an overview of the entities and the relationships between them is given. There, the hierarchies of the entities Student and Exam are stipulated by the examination regulations. As there are many sub-entities of Student and Exam these hierarchies are ommitted and will be explained in detail in the sections about the refinement of the entities Student and Exam respectively.
The model intends that the acquired credits and examination results are related to their corresponding semesters and not to the student. This makes sense as the credits and examination results are acquired in certain semesters. With the relationship precedes an order of the semesters is defined. During the further modelling it will be proved useful that a relationship between the student and all his exams that have to be passed according to the examination regulations exist. Without such relationship we can only relate a student to those exams that have been taken. But these obligatory exams are all known in advance.

3.5 Refining the Entities

In the previous sections it is shown which entities and relatinoships are necessary to deal with the examination regulations. In this section each entity will be examined according to its hierarchical structure, relationships and refinement with attributes.

3.5.1 Modelling a Semester

The semester is the central entity in the model. With this entity the student is connected to his credit accounts and to his examination results. This makes sense because the examination results and the scores of the credit accounts depend on the semester. The semesters are also needed to provide a notion of time. A finer subdivision of time is not necessary because respites and exceptions are not considered. It is mandatory to order and connect the semesters so that the following semester is reachable from the current one. So, the relationships and attributes of this entity are as follows.

relationships of the entity Semester	
precedes	the following semester (or none)
has-score	connects the semester to its corresponding bonus credit account as well as to its corresponding malus credit account
dates-result	the examination results that are achieved within the semester
attributes of the entity Semester	
has-number	the numeric value of the semester

The entity Semester is refined with attributes as pictured in figure 3.5.

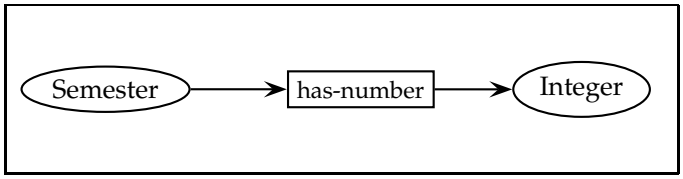


Figure 3.5: The refinement of the entity Semester with attributes

According to the examination regulations the maximum duration of study is at most nine semesters. Hence the possible instances of the entity Semester are already known and can be explicitly listed.

3.5.2 Modelling an Examination

The model of an examination should provide all information that is useful for a user of a query system.

relationships of the entity Exam	
has-result	the results that belong to the exams.
attributes of the entity Exam	
course	the name of the course to which the exam belongs
ultimo	the latest semester in which the exam has to be taken for the first time
credits	the number of credits that can be acquired when passing the exam

So the refinement of Exam is pictured in figure 3.6.

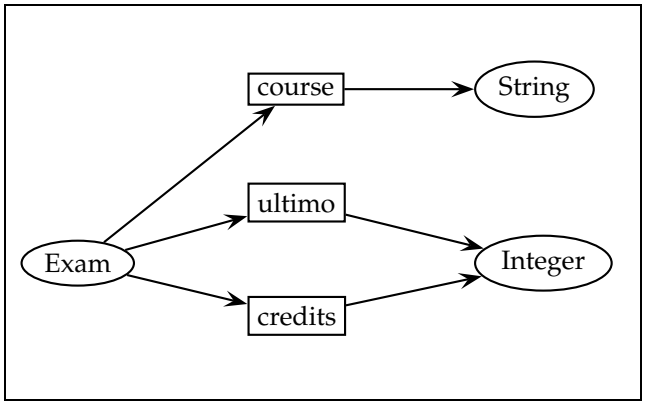


Figure 3.6: Refinement of the entity Exam with attributes

3.5.3 The Hierarchy of Examinations

As mentioned above, the examination regulations define a substructure of examinations. To figure out this structure questions like the following have to be answered.

- To which class of modules do the corresponding modules belong?
- Which results have been achieved in the exams?
- Are the exams still repeatable?
- Are the exams replaceable by exams that belong to modules of the same module class?

So, the hierarchy of examinations is as follows.

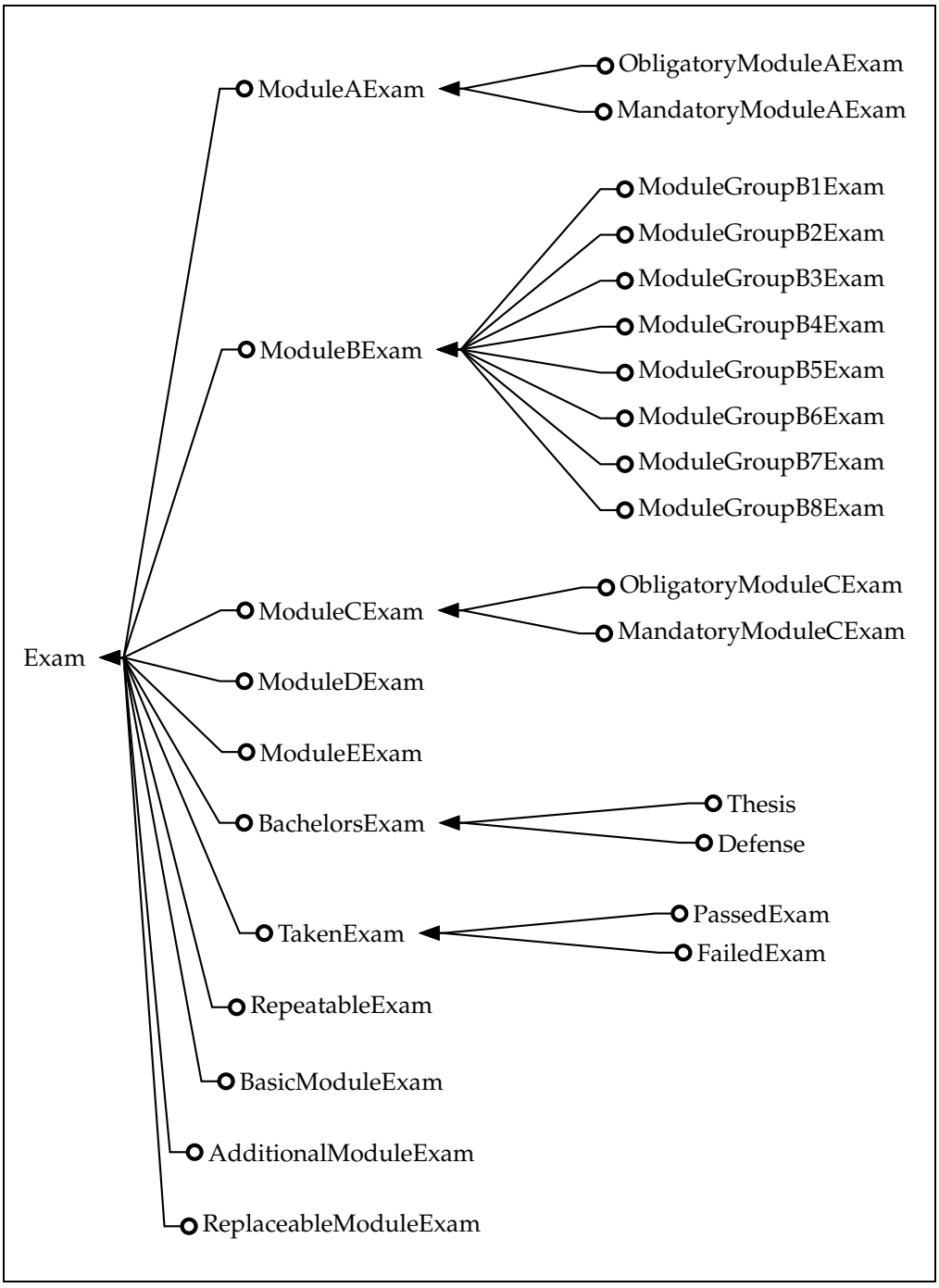


Figure 3.7: Hierarchy of the examination classes.

3.5.4 Modelling an Examination Result

Each exam that is taken by the student will have an examination result assigned. Examinations can have assigned at most three results. The examination results only provide information about the grade that has been achieved.

attributes of the entity ExamResult	
grade	the grade that is achieved in the exam.

This leads to the refinement with attributes and sub-entities as pictured in figure 3.8.

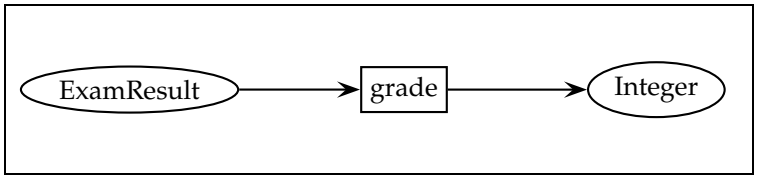


Figure 3.8: Refinement of the entity ExamResult with attributes and sub-entities

3.5.5 Modelling the Credit Accounts

The bonus credit account keeps track of the exams that are passed by the student. The total score of bonus credits is composed of the bonus credits that are acquired in modules of each module-catalogue. So the total score of module exams of each module-catalogue must also be kept. To determine if a student has acquired the required amount of bonus credits in the 4th and the 6th semester these two scores also have to be kept. So the following attributes belong to the entity BonusAccount (cf. figure 3.9).

attributes of the entity BonusAccount	
crObligatoryA	credits that are acquired in obligatory module exams of the module-catalogue A
crMandatoryA	credits that are acquired in mandatory module exams of the module-catalogue A
crElectiveB	credits that are acquired in module exams of the module-catalogue B
crObligatoryC	credits that are acquired in obligatory module exams of the module-catalogue C
crMandatoryC	credits that are acquired in mandatory module exams of the module-catalogue C
crElectiveD	credits that are acquired in elective module exams of the module-catalogue D
crElectiveE	credits that are acquired in elective module exams of the module-catalogue E
crThesis	credits that are acquired in the bachelor's thesis
crDefense	credits that are acquired in the defense of the bachelor's thesis
crBelow6thSem	the total score that is acquired below the 6th semester
crAbove6thSem	the total score that is acquired above or in the 6th semester

These elements can be pictured as follows.

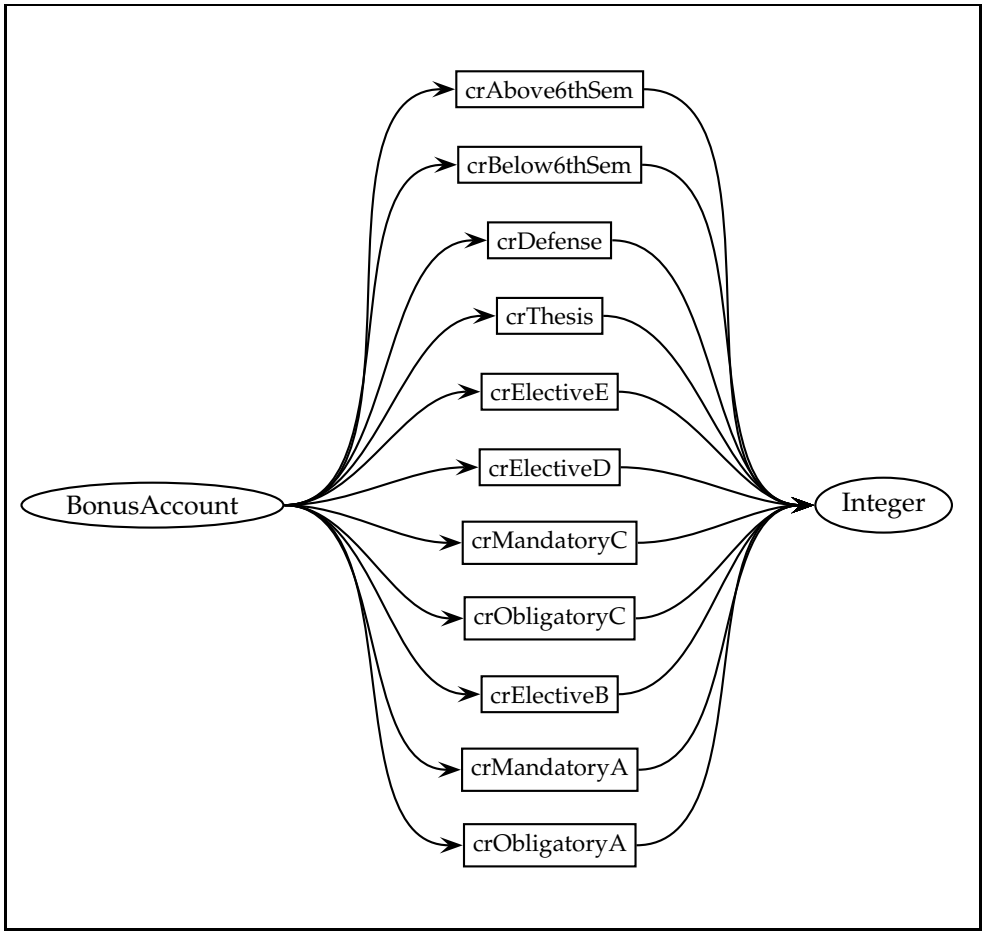


Figure 3.9: The refinement of the entity BonusAccount with attributes

Concerning the malus credit account it is sufficient to save the total amount of malus credits. There the composition of the score is irrelevant.

attributes of the entity MalusAccount	
credits	total amount of malus credits acquired by the student

The refinement of the entity MalusAccount is pictured in figure 3.10.

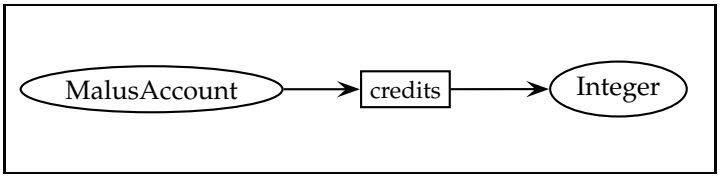


Figure 3.10: The refinement of the entity Semester with attributes

3.5.6 Modelling a Student

All entities mentioned above are used to classify students according to their study characteristics. Before elaborating formal descriptions of its sub-entities the relationships and attributes of Student will be listed below.

relationships of the entity Student	
enrolled-in	the semesters the student has attended
obligatory-exam	the basic module exams that must be passed by the student This relationship allows one to have access to the exams that are not taken yet.

As figured out in section 3.5.7 the entity is divided into sub-entities. According to the semester, the scores of the credit accounts and the exam results the concrete student (representing the user of the query system) is assigned to these classes. Basically the assignment of the concrete student is the main task that must be supported by the model. In the following sections the sub-entities will be formally described in description logics.

3.5.7 The Hierarchy of Students

According to the examination regulations all possible kinds of students build up a hierarchy. These kinds are as follows.

1. Regular student
2. Student in the basic study period
3. Student who has not taken the basic module exams within the first two semesters for the first time
4. Student not participating in the re-examination to the next possible appointment
5. Student who is excluded from the second re-examinations (i.e. more than 90 malus credits acquired)
6. Student in the advanced study period
7. Student who has not taken the remaining examinations within the first six semesters for the first time
8. Student in the 9th semester who has not already passed all elective module exams
9. Student who begins with the bachelor's examination in the 6th semester
10. Student who only lacks the bachelor's examination
11. Student who fails the bachelor's examination for the first time

12. Student who has finished his studies
13. Student who ultimately fails the bachelor's examination
14. Student in the 4th semester with less than 54 bonus credits acquired
15. Student in the 6th semester with less than 108 bonus credits acquired
16. Student who fails the bachelor's examination for the second time

To sketch the hierarchy as concise as possible the student classes are represented in figure 3.11 by their numbers to which they are referred to in the list above.

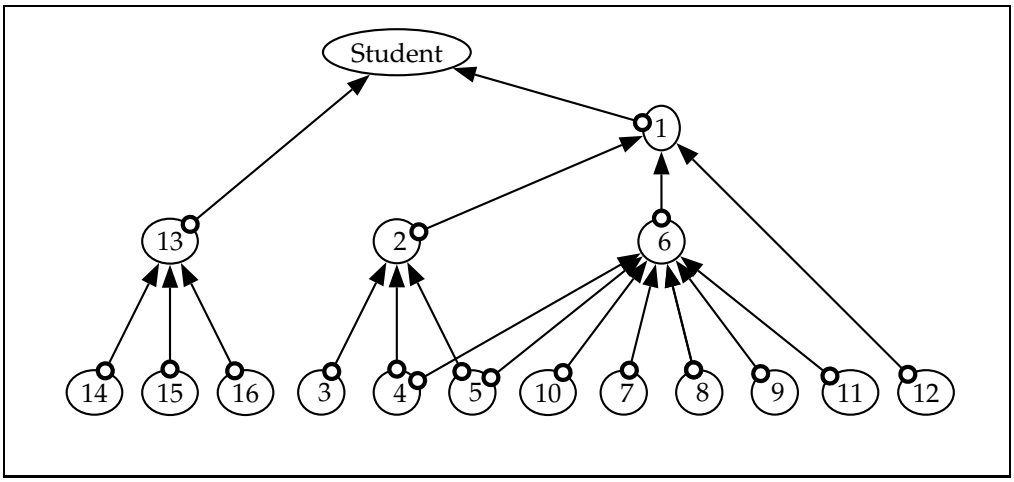


Figure 3.11: Hierarchy of the student classes. The classes are represented by their number to which they are referred to in the list above

The diagram including its refinement and hierarchies could now be directly translated into OWL or its corresponding description logics, respectively. This would be done by translating the entities into concepts and relationships as well as attributes into roles. The knowledge base can be found in section 7. But during the development of the knowledge base it has turnded out that when using the current standard of OWL many things cannot be modelled. For example, we cannot say that a student must have acquired at least 180 bonus credits to finish his studies. Therefore, arithmetics are needed in order to model the sum of bonus credits. The example also shows the need of restricting quantities to lie within certain boundaries (more than 180 credits, less than 54 credits, etc). OWL does not support the modelling with numeric values very well. These aspects as well as a few more have to be discussed before we introduce the knowledge base.

3.6 The Questions in the Terminology of the Model

In the section 3.3.1 we have introduced some questions that we consider to be answered by a query system. In this section we show that in principle, our model is sufficient to answer such questions. To do this we take a few questions and express them using the terminology of our model.

Are there any regulations that contradict each other?
According to our model we would ask whether for any study schedule[2] there are classes that are supposed to be disjoint but have some instance in common.

Do different possible study schedules have unequal durations?
In the terminology of our model we would ask if there are schedules in which a student is finally classified as a student who has finished his studies but where the numbers of the last semesters of the schedules are not equal.

How many module examinations can the student repeat once until his malus credits amount to 90 cr? From which semester may the student repeat exams only once then?
Here we would generate schedules where a student fails each exam two times (provided that the student can fail the exam two times which is not the case for, e.g. basic module exams). At the third time we assume that the exam is passed so that, according to the model, for each exam there are two bad results and one good result. In each semester of each schedule the malus credits increase. According to the model we can then ask which (minimal) amount of bad exam results lead to a malus score of more than or equal to 90 cr and in which semester this score is acquired.

Suppose a student has acquired exactly 54 cr within the first four semesters. Does that mean that he is forced in any case to study nine semester for finishing his studies?
Here, we would generate schedules until the fourth semester in which a student has acquired a total score of 54 cr. Then we would ask if for any continuation of these schedules the number of the semester in which the student is finally classified as a student who has finished his studies is greater than or equal to nine.

May passing any exams except for the basic modules be postponed until the 6th semester?
In the terminology of our model we would ask whether, when starting with schedules until the 6th semester with only the basic module exams passed, every continuation of the schedules leads to the classification of the student as a student that ultimately fails the bachelor's examination.

Which modules of two module groups allow a student to acquire the necessary credits as fast as possible?
In our model we would have to find out which modules of a module-group allow a student to acquire the required credits for the module-catalogue B so that the amount of exams he has to pass is smaller than the amount of exams he has to pass from the other module-group.

[2] By "schedule" we denote an instantiation of our model.

3.7 Open World Assumption and Closed World Assumption in OWL

With closed world assumption we understand the assumption that the knowledge about the domain is complete. By closed world reasoning we denote reasoning under the closed world assumption. With open world assumption, unlike the closed world assumption, we understand the assumption that our knowledge about the domain is incomplete. Reasoning under the open world assumption is also called open world reasoning.
Closed world reasoning is non-monotonic, that is, additional knowledge can falsify our previous knowledge. By contrast, open world reasoning is monotonic, i.e. additional knowledge about the world preserves our previous knowledge.
For the modelling of the examination regulations in OWL we state that both kinds of reasoning make sense and illustrate that on two exemplary reasoning scenarios.

- *Regarding reasoning about exams and their results the closed world assumption makes sense.* Regarding the exam results it makes sense to assume that if we don't explicitly relate an exam to its result (i.e. the exam is taken) then this exam is not taken. In other words, it makes no sense to consider an exam to be taken if we can't deduce that an exam result is related to it.
- *Regarding reasoning about students the open world assumption makes sense.* As we do not know, for example, the number of students in any instantiations of our model, we must deal with incomplete knowledge. Thus, for the reasoning about interrelations between students the open world assumption makes sense. For example, we can't assume that a student Hans finishes his studies faster than any other student only by the fact that Hans is currently the only student in our knowledge base.

So, two types of negations may be needed for this application, a monotonic one and a non-monotonic one. The necessity of adding the capability of closed world reasoning in OWL through epistemic operators is investigated in [3].

3.8 Notes on the Software that is Used for this Project Thesis

During the writing of this project thesis several experiments were conducted in order to get a feeling of how OWL works, and to get to know current OWL ontology engineering tools. As the OWL ontology editor we used Protégé, and we used Pellet as the OWL-DL reasoner. Furthermore we used the Java System Expert Shell (Jess) as the SWRL reasoner (The project thesis deals with SWRL in the section 4.5). Protégé supports Jess as a SWRL reasoner since it provides a control tab for SWRL and the so-called Jess-Bridge which converts SWRL rules to inputs for Jess.
During our experiments with these tools we found out that the SWRL-Bridge can only process a restricted subset of the SWRL rules, namely those which are not defined using anonymous OWL concept constructors as it is proposed in [4]. Another problem arises when mixing OWL with SWRL-rules because Pellet aborts with errors when it encounters SWRL-rules in the ontology document. The SWRL reasoner (or the SWRL-Bridge, resp.) in turn, ignores OWL restrictions on concepts and roles, so that using SWRL rules can lead to inconsistencies between the OWL ontology and the SWRL rules. As a consequence, we could either work with an OWL ontology or with SWRL rules. The interested reader can find a list of description logic reasoners in [8].
The problems which occur in the use of the tools mentioned above and additional issues as well as proposals to solve these problems are discussed in the remaining thesis.

4 Proposals to Improve OWL

During the modelling of the examination regulations some difficulties in modelling according to the syntax and semantics of OWL were explored. In this section proposals to improve OWL are made in order to enlarge the area of application by making OWL more convenient to use. The most of the topics that are addressed in this section already have been explored in several papers, the most of them only concentrate on reasoning aspects. In this section we will make a few proposals to improve OWL and discuss their consequences for the reasoning but also for the modelling.

4.1 Customised Datatypes in OWL

Many student class definitions make the affiliation of students depending on (numeric) attribute values to lie within certain boundaries. For example, consider the class InBasicStudyPeriod to which a student belongs if he hasn't been enrolled in a semester greater than the fourth. That is, there may only be semesters the student has been enrolled in, whose attributes has-number have an integer value lower than or equal to four.

$$\mathsf{InBasicStudyPeriod} \equiv \mathsf{Student} \sqcap \forall \mathsf{enrolled\text{-}in}.(\mathsf{Semester} \sqcap \exists \mathsf{has\text{-}number}.\ \leq_4).$$

The same applies to the class InAdvancedStudyPeriod to which a student belongs if he has been in a semester greater than the fourth.

$$\mathsf{InAdvancedStudyPeriod} \equiv \mathsf{Student} \sqcap \exists \mathsf{enrolled\text{-}in}.(\mathsf{Semester} \sqcap \exists \mathsf{has\text{-}number}.\ >_4).$$

Another example is the definition of the class of students that must not participate in second re-examinations, if they have acquired more than 52 malus credits.

$$\begin{aligned}\mathsf{NoAdmittanceTo2ndReExams} \equiv\ & \mathsf{Student}\ \sqcap \exists \mathsf{enrolled\text{-}in}.(\mathsf{Semester} \sqcap \\ & \exists \mathsf{has\text{-}score}.(\mathsf{MalusAccount} \sqcap \exists \mathsf{credits}.\ >_{52}))\end{aligned}$$

According to the current standard of OWL it is not possible to express that certain attribute values must lie within certain boundaries. That OWL needs to provide the facility to define customised datatypes has been explored in [6]. Many applications need such value restrictions for numeric attributes. In the field of e-commerce, for example, consider the definition of a class of business customers who get quantity discount if they order more than 1000 pieces (which we assume to be modelled as a numeric attribute quantity).

$$\mathsf{Customer\text{-}QuantityDiscount} \equiv \mathsf{Customer} \sqcap \exists \mathsf{order}.(\mathsf{Order} \sqcap \exists \mathsf{quantity}.\ >_{1000})$$

Or imagine a model of a thermo control system that classifies a process as a CriticalProcess if its temperature exceeds 250 °C.

$$\mathsf{CriticalProcess} \equiv \mathsf{Process} \sqcap \exists \mathsf{has\text{-}temperature}.\ >_{250}$$

Customised datatypes can be "simulated" in OWL using the cardinality restrictions of roles (cf. [10]). Consider the example Customer-QuantityDiscount above. To transform the class in a way that we can use cardinality restrictions instead of customised datatypes, we define a new class $\mathsf{Unit} \equiv \{\mathsf{u}\}$ and redefine the attribute quantity by the relation $\mathsf{quantity} \subseteq \mathsf{Order} \times \mathsf{Unit}$. Adding $(\mathsf{o}, \mathsf{u}) : \mathsf{quantity}$ to the instances as much as there are pieces that the order o comprises, we model the quantity of pieces through the number of edges that connect o with u. For this example we assume that the order o contains 1020 pieces. Therefore, we will add the following to the set of role instances.

$$1020 \text{ times } \begin{cases} (\mathsf{o}, \mathsf{u}) : \mathsf{quantity} \\ \ldots \\ (\mathsf{o}, \mathsf{u}) : \mathsf{quantity} \end{cases}$$

We can then redefine Customer-QuantityDiscount using cardinality restrictions for roles:

$$\text{Customer-QuantityDiscount} \equiv \text{Customer} \sqcap \exists \text{order}.(\text{Order} \sqcap > 1000\ \text{quantity}).$$

A problem that arises with this modelling approach is that it is quite unnatural to model numeric boundaries in this way. Another problem is that the set of instances gets blown up with many role-instances. In the example above there are 1020 quantity-instances for only one instance of Order, unmentioned that there might be further instances. As a consequence, this approach is inappropriate in practice.
Extending OWL with customised datatypes (commonly known as *OWL-Eu*) has been explored in [6]. In this paper it is also shown that reasoning in OWL-Eu remains decidable. So, it is desirable that customised datatypes will be added in the next versions of OWL.

4.2 Arithmetics in OWL

When looking at the class TooFewCreditsInSemester4 and its definition

$$\begin{aligned}&\text{Student} \sqcap \exists \text{enrolled-in}.(\text{Semester} \sqcap \exists \text{has-number}.\ =_4 \sqcap \\ &\quad \forall \text{has-score}.(\text{BonusAccount} \sqcap \exists \text{crBelow4thSemester}.\ <_{54})),\end{aligned}$$

a student belongs to that class if he has been enrolled in the fourth semester and has acquired less than 54 credits as a result of the sum of the acquired bonus credits. That the attribute crBelow4thSemester is supposed to be the sum of all acquired bonus credits cannot be explicitly modelled in OWL. Thus, the specification of the model in OWL is not complete. Hence, the user of the model (or the application programmer, resp.) must guarantee that the attribute crBelow4thSemester contains the sum of the acquired bonus credits. So, from a modelling standpoint, it is desirable that OWL provides arithmetic functions like sum, product and division. In the case of crBelow4thSemester one can imagine a role restriction that stipulates that the attribute contains the sum of the acquired bonus credits:

$$\begin{aligned}\text{crBelow4thSemester} \equiv \text{Sum}(&\text{crDefense, crThesis, crElectiveE, crElectiveD,crMandatoryC,} \\ &\text{crObligatoryC, crElectiveB, crMandatoryA, crObligatoryA})\end{aligned}$$

According to [1] adding aggregate functions (like arithmetic functions) makes reasoning in OWL-DL undecidable. This result is already shown by the "weak" set of aggregate functions sum, min and max over non-negative integers. As a consequence, aggregate functions will not be available for reasoning in OWL.
What is proposed in this project thesis, however, is to add aggregate functions in order to provide an interface for an evaluation program that is invoked prior to the reasoning. Such an interface could be syntactically integrated in OWL by adding restrictions to role definitions, which is shown below using the abstract syntax of OWL.

```
1  DatatypeProperty( crBelow4thSemester
2    domain(BonusAccount)
3    range(xsd:NonNegativeInteger)
4    ...
5    restriction(
6     value( sum( crDefense, crThesis, crElectiveE, crElectiveD,
7           crManadatoryC, crObligatoryC, crElectiveB,
8           crMandatoryA, crObligatoryA ))
9     )
10 )
```

In a pre-processing step for each instance of the role crBelow4thSemester the sum of the corresponding credits is computed and added to the instances.

$$
\begin{array}{rcl}
(\mathsf{s}_1, \mathsf{bonus}_1) & : & \mathsf{has\text{-}score} \\
(\mathsf{s}_2, \mathsf{bonus}_2) & : & \mathsf{has\text{-}score} \\
(\mathsf{s}_3, \mathsf{bonus}_3) & : & \mathsf{has\text{-}score} \\
(\mathsf{s}_4, \mathsf{bonus}_4) & : & \mathsf{has\text{-}score} \\
\ldots & & \\
(\mathsf{s}_9, \mathsf{bonus}_9) & : & \mathsf{has\text{-}score} \\
 & & \\
(\mathsf{bonus}_1, 16) & : & \mathsf{crBelow4thSemester} \\
(\mathsf{bonus}_2, 32) & : & \mathsf{crBelow4thSemester} \\
(\mathsf{bonus}_3, 00) & : & \mathsf{crBelow4thSemester} \\
(\mathsf{bonus}_4, 00) & : & \mathsf{crBelow4thSemester} \\
\ldots & & \\
(\mathsf{bonus}_9, 00) & : & \mathsf{crBelow4thSemester}
\end{array}
$$

When pre-processing the document in that way the reasoning remains unaffected. As an advantage the model can be completely specified in OWL. With the possiblilty of specifying such dependencies among the attributes we can avoid inconsistencies in the application.

4.3 Finite Quantification

In many modelling cases classes are defined in a most suitable way by explicitly declaring instance affiliations to them. The colours of wine can only be red, white or rose, a weekday can only be monday, tuesday, wednesday, thursday, friday, saturday or sunday. Besides that, the colours of wine as well as the days of a week are fix. That is, it can only be unintentional from a modelling point of view to allow extensions of the colour of wine and the days of a week. With *Enumerated Classes* OWL provides the facility to define classes by their instances. In DL syntax we would describe the class of wines as $\mathsf{Wine} \equiv \{\mathsf{red}, \mathsf{white}, \mathsf{rose}\}$ and the class of weekdays as $\mathsf{Weekday} \equiv \{\mathsf{mon}, \mathsf{tue}, \mathsf{wed}, \mathsf{thur}, \mathsf{fri}, \mathsf{sat}, \mathsf{sun}\}$. Enumerated classes are also used in the model of the examination regulations to define, e.g. the classes Exam and Semester. That a definition of these two classes is given using enumerated classes meets the examination regulations in a way that the amount of semesters and possible exams are stipulated by the regulations. Extensions of these classes by the user ex post changes the model of examination regulations, thus falsifying the propositions made by the query system.
In some papers the instances of enumerated classes are also referred to as so-called *nominals*. Nominals are instances of singleton classes. An enumerated class can then be seen as a union of nominals, e.g.: $\mathsf{Wine} \equiv \{\mathsf{red}\} \sqcup \{\mathsf{white}\} \sqcup \{\mathsf{rose}\}$. In the following, we will use the terms *instance of an enumerated class* and *nominal* interchangeably.
What is likely to appear in conjunction with enumerated classes are definitions of classes according to certain values of attributes that range over nominals. For example, consider the class of red wine:

$$\mathsf{RedWine} \equiv \mathsf{Wine} \sqcap \exists \mathsf{has\text{-}colour}.\{\mathsf{red}\}.$$

As a further example consider the class of orders that are paid with credit cards:

$$\mathsf{CCPayedOrders} \equiv \mathsf{Order} \sqcap \exists \mathsf{payed\text{-}with}.\{\mathsf{credit\text{-}card}\},$$

where we assume three types of payments:

$$\mathsf{Payment} \equiv \{\mathsf{on\text{-}account}, \mathsf{credit\text{-}card}, \mathsf{cash\text{-}on\text{-}delivery}\}.$$

During modelling it is observed that class definitions including nominals can be very complex. We will demonstrate that with the following example. According to the examination regulations students have to take all basic module exams within the first two semesters for the first time (cf. section 6). To model this a class HasTakenAllBasicModules is defined to which all students belong if they have taken all basic module exams at least once. Hence, for each basic module exam there must be a semester in which the student has acquired at least one basic module examination result. According to the examination regulations the enumerated class BasicModuleExam containing all basic modules is defined as:

$$\mathsf{BasicModuleExam} \equiv \{\mathsf{in0001}, \mathsf{in0003}, \mathsf{in0004}, \mathsf{in0006}, \mathsf{in0007}, \mathsf{in0015}, \mathsf{ma0901}, \mathsf{ma0902}\},$$

where we represent the basic modules by their identifiers that can be found in the sections 6.1.2 to 6.1.5. This leads to the following definition of HasTakenAllBasicModules:

$$\begin{aligned}
\mathsf{HasTakenAllBasicModules} \equiv\ & \mathsf{Student} \sqcap \\
& \exists \mathsf{enrolled\text{-}in}.(\mathsf{Semester} \sqcap \exists \mathsf{dates\text{-}result}.(\mathsf{ExamResult} \sqcap \exists \mathsf{has\text{-}result}^{-}.\{\mathsf{in0001}\})) \sqcap \\
& \exists \mathsf{enrolled\text{-}in}.(\mathsf{Semester} \sqcap \exists \mathsf{dates\text{-}result}.(\mathsf{ExamResult} \sqcap \exists \mathsf{has\text{-}result}^{-}.\{\mathsf{in0003}\})) \sqcap \\
& \exists \mathsf{enrolled\text{-}in}.(\mathsf{Semester} \sqcap \exists \mathsf{dates\text{-}result}.(\mathsf{ExamResult} \sqcap \exists \mathsf{has\text{-}result}^{-}.\{\mathsf{in0004}\})) \sqcap \\
& \exists \mathsf{enrolled\text{-}in}.(\mathsf{Semester} \sqcap \exists \mathsf{dates\text{-}result}.(\mathsf{ExamResult} \sqcap \exists \mathsf{has\text{-}result}^{-}.\{\mathsf{in0006}\})) \sqcap \\
& \exists \mathsf{enrolled\text{-}in}.(\mathsf{Semester} \sqcap \exists \mathsf{dates\text{-}result}.(\mathsf{ExamResult} \sqcap \exists \mathsf{has\text{-}result}^{-}.\{\mathsf{in0007}\})) \sqcap \\
& \exists \mathsf{enrolled\text{-}in}.(\mathsf{Semester} \sqcap \exists \mathsf{dates\text{-}result}.(\mathsf{ExamResult} \sqcap \exists \mathsf{has\text{-}result}^{-}.\{\mathsf{in0015}\})) \sqcap \\
& \exists \mathsf{enrolled\text{-}in}.(\mathsf{Semester} \sqcap \exists \mathsf{dates\text{-}result}.(\mathsf{ExamResult} \sqcap \exists \mathsf{has\text{-}result}^{-}.\{\mathsf{ma0901}\})) \sqcap \\
& \exists \mathsf{enrolled\text{-}in}.(\mathsf{Semester} \sqcap \exists \mathsf{dates\text{-}result}.(\mathsf{ExamResult} \sqcap \exists \mathsf{has\text{-}result}^{-}.\{\mathsf{ma0902}\}))
\end{aligned}$$

Before we continue with the introduction of finite quanitification it should be mentioned that the definition above is different from the following ones[3].

- $\ldots \exists \mathsf{enrolled\text{-}in}.(\mathsf{Semester} \sqcap \exists \mathsf{dates\text{-}result}.(\mathsf{ExamResult} \sqcap \exists \mathsf{has\text{-}result}^{-}.\mathsf{BasicModuleExam}))$ means that the student must have attended at least one semester in which he has acquired at least one basic module exam result. This also includes the unintentional case in which the student has acquired a basic module re-examination result while there may still be basic modules left without an examination result assigned.
- $\ldots \exists \mathsf{enrolled\text{-}in}.(\mathsf{Semester} \sqcap \exists \mathsf{dates\text{-}result}.(\mathsf{ExamResult} \sqcap \forall \mathsf{has\text{-}result}^{-}.\mathsf{BasicModuleExam}))$ means that the student must have attended at least one semester in which he has acquired at least one examination result that is only assigned to basic modules. Here again, not all basic module exams might be included because this sub-formula only makes propositions about existent examination results.
- $\ldots \exists \mathsf{enrolled\text{-}in}.(\mathsf{Semester} \sqcap \forall \mathsf{dates\text{-}result}.(\mathsf{ExamResult} \sqcap \exists \mathsf{has\text{-}result}^{-}.\mathsf{BasicModuleExam}))$ means that the student must have attended at least one semester in which he has only acquired basic module exam results.
- $\ldots \exists \mathsf{enrolled\text{-}in}.(\mathsf{Semester} \sqcap \forall \mathsf{dates\text{-}result}.(\mathsf{ExamResult} \sqcap \forall \mathsf{has\text{-}result}^{-}.\mathsf{BasicModuleExam}))$ means that the student must have attended at least one semester in which he has only acquired exam result that only belong to basic modules.

Using only the OWL-quantifiers, it is impossible to express what is actually meant, which makes the extensive definition of HasTakenAllBasicModules necessary. What might be annoying is the fact, that from the third line on all subsequent lines are copies of the second one with

[3]The following sub-formulae are considered to replace the part from the second line on.

only the nominals changing. This is, what *finite quantification* avoids.
With finite quantification we understand the following concept constructors

$$\bigsqcap_{o:EC} \ldots \exists role.\{o\} \ldots \quad \text{and} \quad \bigsqcup_{o:EC} \ldots \exists role.\{o\} \ldots,$$

where EC is an enumerated class that will be denoted as *index class*, o is a *(finite quantification) variable* ranging over the instances of EC and role denotes an OWL-DL role. With finite quantification HasTakenAllBasicModules can be described as

$$\begin{aligned} &\mathsf{HasTakenAllBasicModules} \equiv \mathsf{Student} \sqcap \\ &\quad \bigsqcap_{e:\mathsf{BasicModuleExam}} \exists \text{enrolled-in}.(\mathsf{Semester} \sqcap \exists \text{dates-result}.(\mathsf{ExamResult} \sqcap \exists \text{has-result}^{-}.\{e\})). \end{aligned}$$

Finite quantification offers more convenience to the modeller, since he is able to define classes like HasTakenAllBasicModules in a shorter form. This increases the conciseness of the knowledge base. Hence, the maintenance of the knowledge base is improved. This permits more efficient changes in the knowledge base. The usage of finite quantification also prevents the modeller from making annoying "copy-and-paste-errors".

From a reasoning point of view finite quantification can be implemented in different ways. A naive implementation provides finite quantification as macros that are expanded in a preprocessing step, i.e. before the reasoner starts its reasoning service. As an example, consider the following "input" for the reasoner:

$$\begin{aligned} \mathsf{StylishWineDrinker} &\equiv \mathsf{WineDrinker} \sqcap \bigsqcup_{w:\mathsf{Wine}} (\exists \text{prefers}.\{w\} \sqcap \exists \text{prefers}.(\mathsf{Glass} \sqcap \exists \text{suitedFor}.\{w\})) \\ \mathsf{Wine} &\equiv \{\text{red}, \text{white}, \text{rose}\} \end{aligned}$$

A macro-preprocessor would expand the finite quantification by generating three copies of the sub-formula behind $\bigsqcup_{w:\mathsf{Wine}}$, replacing the variable w with each nominal of Wine and linking the resulting three subformulae with disjunctions:

$$\begin{aligned} &\mathsf{StylishWineDrinker} \equiv \mathsf{WineDrinker} \sqcap \\ &\quad [(\exists \text{prefers}.\{\text{red}\} \sqcap \exists \text{prefers}.(\mathsf{Glass} \sqcap \exists \text{suitedFor}.\{\text{red}\})) \sqcup \\ &\quad (\exists \text{prefers}.\{\text{white}\} \sqcap \exists \text{prefers}.(\mathsf{Glass} \sqcap \exists \text{suitedFor}.\{\text{white}\})) \sqcup \\ &\quad (\exists \text{prefers}.\{\text{rose}\} \sqcap \exists \text{prefers}.(\mathsf{Glass} \sqcap \exists \text{suitedFor}.\{\text{rose}\}))] \\ &\mathsf{Wine} \equiv \{\text{red}, \text{white}, \text{rose}\} \end{aligned}$$

In this case, the reasoner does not need to know anything about finite quantification. The reasoner only needs to invoke the pre-processor. Another possibility is that the reasoner and the pre-processor work interleaved, i.e. the reasoner does not start the pre-processing of the whole knowledge base at the beginning of its reasoning, but whenever it encounters a class definition that includes finite quantification. As the reasoner might not need to evaluate every class definition of a knowledge base, an interleaved processing of macro-expansion and reasoning could make sense, especially when dealing with large knowledge bases.
Another approach to implement finite quantification is to provide a well defined semantics, so that the reasoner can include finite quantification to its reasoning. To see, that the performance of the reasoning could then be improved, consider the following more abstract example:

$$\begin{aligned} \mathsf{Class\text{-}A} &\equiv \mathsf{C} \sqcap \bigsqcup_{b:\mathsf{Class\text{-}B}} (\bot \sqcap \exists \text{has-value}.\{b\}) \\ \mathsf{Class\text{-}B} &\equiv \{b_1, b_2, \ldots, b_{100}\} \end{aligned}$$

A reasoner that understands finite quantification would detect the unsatisfiability of Class-A in one step, whereas in the expanded version of Class-A the reasoner might need to check all other disjunctive sub-formulae before being sure that this class is unsatisfiable.
Regarding the usage of finite quantification, termination must be guaranteed. Therefore, it is proposed to restrict the index classes to be enumerated classes. Because enumerated classes are defined as unions of finitely many nominals (i.e. there's no computation of classification that might be infinite), the following definitions are impossible:

$$A \equiv \bigsqcup_{b:B} D \sqcap \text{has-value.}\{b\} \quad \text{and} \quad B \equiv \bigsqcup_{a:A} E \sqcap \text{has-value.}\{a\}.$$

What might be disturbing is the occurence of the finite quantification variable, since variables don't occur in OWL-DL explicitly. To hide the finite quantification variable, we could represent the current nominal as a (predefined and special) singleton class, here Δ:

$$\bigsqcap_{EC} \ldots \exists \text{role.}\Delta \ldots \quad \text{and} \quad \bigsqcup_{EC} \ldots \exists \text{role.}\Delta \ldots .$$

In this approach, the finite quantification variable disappears (only the index class remains) while a special singleton class is introduced (here Δ). This class has a predefined and constant name (like $\top$ or $\bot$), but the comprised nominal changes in each expansion step, like a window consisting of one nominal that gets shifted over the index class.
One problem that arises when using the variable-free variant of finite quantification is when finite quantifiers are cascaded. It is then impossible to assign the appearances of Δ to their finite quantifiers in an unambiguous way. To solve this problem we make use of the "depth" of the finite quantifiers. That is, we define the most inner quantifier to be of depth zero, the quantifier that surrounds the most inner one to be of depth one and so forth. The numeration of the depth is thus directed from the most inner quantifier to the most outer one. Accordingly, with this notion of depth we can assign appearances of Δ to the quantifiers to which they belong. This is done by introducing the "depth-selector", ':', that prefixes appearances of Δ. Hence, Δ is assigned to the most inner quantifier, $:\Delta$ is assigned to the quantifier that surrounds the most inner one (defined as of depth one), $::\Delta$ is assigned to the quantifier that surrounds the quantifier of depth one and so on. For example, the formula

$$\bigsqcap_{a:E} \bigsqcap_{b:E} \bigsqcap_{c:E} (\{b\} = \{a\}) \sqcup (\{b\} = \{c\})$$

can then be rewritten as

$$\bigsqcap_{E} \bigsqcap_{E} \bigsqcap_{E} (:\Delta = ::\Delta) \sqcup (:\Delta = \Delta).$$

For convenience, we might also allow abbreviations of the form $:n\Delta$ instead of $\underbrace{:\ldots:}_{n \text{ times}}\Delta$, so that $:3\Delta$ abbreviates $:::\Delta$ and clearly, $:0\Delta$ is the same as Δ.

4.4 Qualified Number Restrictions and Range Restrictions of Roles

With qualified number restrictions we understand the concept constructor $\leq$ n role.C with the semantics

$$(\leq \text{ n role.C})^{\mathcal{I}} = \{x \mid \#\{y \mid (x, y) \in \text{role}^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \leq n\}.$$

To see, in what way this concept constructor is useful for the modelling, imagine that we have to develop a knowledge base for an online store that sells operating systems. Imagine further that the persons in charge have agreed to a model that contains the part which is shown in figure 4.12.

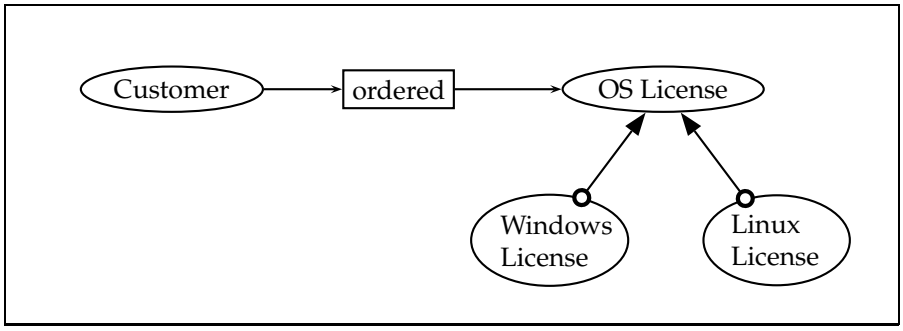


Figure 4.12: Part of a diagrammatic model which shows the usage of qualified number restrictions.

During the refinement of Customer, a subsumed concept Windows-Customer shall be modelled. A customer is thereby classified to Windows-Customer, if he has ordered more than 20 Windows licenses. With qualified number restrictions this concept can be easily described as

$$\text{Windows-Customer} \equiv \text{Customer} \sqcap > 20\,\text{ordered.Windows}$$

As in the current OWL standard qualified number restrictions aren't available, the model needs to be modified as shown in figure 4.13.

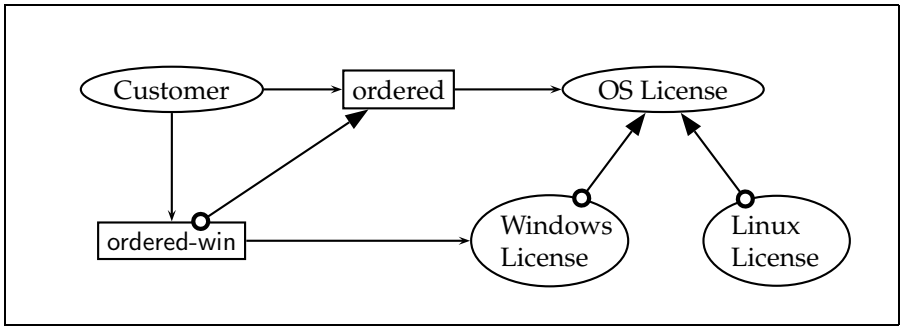


Figure 4.13: Modified part of the model, the modification is necessary in order to describe the concept Windows-Customer.

According to that figure a subproperty ordered-win is introduced that relates customers to their ordered Windows licenses. With this modification we can describe the concept Windows-Customer' as

$$\text{Windows-Customer'} \equiv \text{Customer} \sqcap > 20\,\text{ordered-win.}$$

For the modeller the lack of qualified number restrictions necessitates to make the dependencies between the property ordered and the subsumed concept Windows explicit. These dependencies are already implicitly given by the OS License subsumption hierarchy. With qualified number restrictions it is possible to transfer role hierarchies to such concept hierarchies.
Further use cases of qualified number restrictions and work-arounds to by-pass them can be found in [9]. These work-arounds can be very difficult and cumbersome and may lead to a model that is not correct in general, which can be comprehended in [12]. This article also refers to [11] in order to show that adding qualified number restrictions doesn't change the worst case complexity of OWL-DL reasoning. But the performance of the reasoner actually suffers since the computation of qualified number restrictions can behave very badly. However, according to [12] several reasoners (like the highly optimised reasoner FaCT or Protégé-OWL) already support qualified number restrictions. Since working with qualified number

restrictions seems feasible for many applications and to avoid proprietary OWL derivatives it should be considered to include qualified number restrictions in the future OWL standards.

Qualified number restrictions are based on range restrictions of roles[2]. With these restrictions roles are described as p.C, where p is a role and C is a concept description. Note that p.C describes a new role which arises from the role p restricted to C. For example, the part ordered.Windows in the concept description Windows-Customer describes a new role which arises from the role ordered restricted to the concept Windows. It expresses that regarding the role ordered we only want to take instances of Windows into account. With range restrictions of roles we can define the concept constructors and their semantics as

$$\begin{aligned}(\forall(\mathsf{role.C}).\mathsf{D})^{\mathcal{I}} &= \{x \mid \forall y.\,(x,y) \in \mathsf{role}^{\mathcal{I}} \wedge y \in \mathsf{C}^{\mathcal{I}} \rightarrow y \in \mathsf{D}^{\mathcal{I}}\}\\ (\exists(\mathsf{role.C}).\mathsf{D})^{\mathcal{I}} &= \{x \mid \exists y.\,(x,y) \in \mathsf{role}^{\mathcal{I}} \wedge y \in \mathsf{C}^{\mathcal{I}} \wedge y \in \mathsf{D}^{\mathcal{I}}\}\\ (\leq\ \mathsf{n}\,\mathsf{role.C})^{\mathcal{I}} &= \{x \mid \#\{y \mid (x,y) \in \mathsf{role}^{\mathcal{I}} \wedge y \in \mathsf{C}^{\mathcal{I}}\} \leq \mathsf{n}\},\end{aligned}$$

where the latter concept constructor is apparently what we have introduced as qualified number restrictions at the beginning of this section. The concept constructors without range restrictions of roles can easily be transferred to concept constructors with range restrictions of roles using the equations

$$\begin{aligned}\forall \mathsf{role.C} &\equiv \forall(\mathsf{role}.\top).\mathsf{C}\\ \exists \mathsf{role.C} &\equiv \exists(\mathsf{role}.\top).\mathsf{C}\\ \leq\ \mathsf{n}\,\mathsf{role} &\equiv \leq\ \mathsf{n}\,\mathsf{role}.\top.\end{aligned}$$

That is, roles without range restrictions of roles can be seen as roles that are restricted to $\top$. This is actually no restriction (since every instance belongs to $\top$) and can be eliminated by the reasoner.
We will now take a look at a further example that shows us the usage of concepts using range restrictions of roles. The concept of parents all of whose daughters are students can be described as

$$\mathsf{ParentOfFemaleStudent} \equiv \mathsf{Parent} \sqcap \forall(\mathsf{has\text{-}child.Female}).\mathsf{Student}.$$

Note that the meaning of the concept above is different from the following circumscription using concept constructors without range restrictions of roles:

$$\mathsf{ParentOfFemaleStudent'} \equiv \mathsf{Parent} \sqcap \forall \mathsf{has\text{-}child}.(\mathsf{Female} \sqcap \mathsf{Student}),$$

means that parents belonging to ParentOfFemaleStudent' only have children who are daughters and students. The previous concept, however, defines parents whose daughters are students, making no assumptions about their sons.

What makes reasoning with range restrictions of roles hard is, among others, the higher complexity of the computation of subsumption. In [2] the bad performance is proven by reduction of subsumption to testing unsatisfiability of CNF formulae. Other reasons for the loss of good performance can be read in [5]. In contrast to these results the reasoning with range restrictions of roles is amenable to optimisations (cf. [5]) in order to provide reasoning with range restrictions of roles for common applications at a reasonable speed.

4.5 Using SWRL

The *Semantic Web Rule Language, SWRL* for short (cf. [4]), is a W3C Member Submission that proposes to combine OWL with the unary/binary datalog sublanguages of the Rule Markup Language (RuleML). With SWRL one adds horn-like rules to the set of OWL axioms. This project thesis deals with SWRL for the reason that it is more expressive than first-order logics

without function symbols and only two variables within the same scope. The expressivity of OWL is equal to that sub-family of first order logics. In other words, SWRL allows the use of arbitrarily many variables within the same scope whereas OWL does not. Before we look at the consequences of using SWRL for the modelling and the reasoning, we will give a short introduction of the syntax and semantics of SWRL.

As said before, SWRL is a rule language, where these rules correspond to the views commonly known from database systems. SWRL rules are of the form antecedent $\rightarrow$ consequent, where antecedent and consequent are both disjunctions of *atoms*. These atoms, in turn, can be of the form $\mathsf{C}(?x)$, $\mathsf{P}(?x, ?y)$, $\mathsf{sameAs}(?x, ?y)$, $\mathsf{differentFrom}(?x, ?y)$ or $\mathsf{builtIn}(\mathsf{r}, ?z1, \ldots, ?zn)$. Thereby, C denotes an OWL concept description, P denotes an OWL (datatype or object) property, builtIn denotes a built-in function of name r and ?x, ?y, ?z1 to ?zn denote variables (prefixed with "?"). Besides this human readable syntax, an abstract (declarative) syntax as well as a concrete, XML based syntax are provided (cf. [4]).

The semantics of SWRL extends the semantics of OWL in a way that variables are additionally bound to individuals or datatype values. Then, the atom $\mathsf{C}(?x)$ holds if the individual to which ?x is bound is element of that set C is interpreted as. Similarily, $\mathsf{P}(?x, ?y)$ holds if the tupel of individuals bound by ?x and ?y is an element of the relation P is interpreted as. The atom $\mathsf{sameAs}(?x, ?y)$ holds if ?x and ?y are bound to the same individual, otherwise $\mathsf{differentFrom}(?x, ?y)$ holds. The atom $\mathsf{builtIn}(\mathsf{r}, ?z1, \ldots, ?zn)$ holds if ?z1 is the result of the function, r is interpreted as, applied to the individuals to which the arguments ?z2 to ?zn are bound. An interpretation satisfies the antecedent if it satisfies all atoms of the antecedent or the antecedent is empty. In the same manner, an interpretation satisfies the consequent if it satisfies all atoms of the consequent and the consequent is not empty. As a first example consider the concept description TakenExam to which all exams belong that have been assigned at least one examination results.

$$\mathsf{TakenExam} \equiv \mathsf{Exam} \sqcap \exists \mathsf{has\text{-}result}.\mathsf{ExamResult}$$

In SWRL we would capture this notion with the following rule.

$$\mathsf{Exam}(?x) \wedge \mathsf{has\text{-}result}(?x, ?y) \wedge \mathsf{ExamResult}(?y) \rightarrow \mathsf{TakenExam}(?x)$$

Another example is the concept TooFewCreditsInSemester4 and its description

$$\begin{aligned}\mathsf{TooFewCreditsInSemester4} \equiv\ & \mathsf{Student} \sqcap \exists \mathsf{enrolled\text{-}in}.(\mathsf{Semester} \sqcap \exists \mathsf{has\text{-}number}.\ =_4 \sqcap \\ & \forall \mathsf{has\text{-}score}.(\mathsf{BonusAccount} \sqcap \exists \mathsf{crBelow4thSemester}.\ <_{54})).\end{aligned}$$

Since SWRL provides a variety of built-in functions including arithmetics and comparison operations (cf. [4]), one can express that the property crBelow4thSemester is the sum of all acquired bonus credits and that it must lie within certain boundaries.

$$\begin{aligned}&\mathsf{Student}(?st) \wedge \mathsf{enrolled\text{-}in}(?st, ?se) \wedge \mathsf{Semester}(?se) \wedge \mathsf{has\text{-}number}(?se, ?n) \wedge \\ &\quad \mathsf{builtIn}(\mathsf{equals}, ?n, 4) \wedge \mathsf{has\text{-}score}(?se, ?sc) \wedge \mathsf{BonusAccount}(?sc) \wedge \\ &\quad \mathsf{builtIn}(\mathsf{add}, ?res, \mathsf{crDefense}, \mathsf{crThesis}, \mathsf{crElectiveE}, \mathsf{crElectiveD}, \mathsf{crMandatoryC}, \\ &\qquad \mathsf{crObligatoryC}, \mathsf{crElectiveB}, \mathsf{crMandatoryA}, \mathsf{crObligatoryA}) \wedge \\ &\quad \mathsf{builtIn}(\mathsf{lowerThan}, ?res, 54) \wedge \mathsf{builtIn}(\mathsf{equals}, ?res, \mathsf{crBelow4thSemester}) \\ &\rightarrow \mathsf{TooFewCreditsInSemester4}(?st)\end{aligned}$$

Very similar to TakenExam we would describe the concept FailedExam to which those exams belong to that have only exam results being BadResults.

$$\mathsf{FailedExam} \equiv \mathsf{Exam} \sqcap \forall \mathsf{has\text{-}result}.\mathsf{BadResult}$$

To describe this concept (and the universal role quantification) with a SWRL rule we have to use the facility to describe OWL concepts in SWRL directly[4]:

$$\mathsf{Exam}(?x) \wedge (\forall \mathsf{has\text{-}result.BadResult})(?x) \rightarrow \mathsf{FailedExam}(?x).$$

As the latter example shows, OWL can be seen as a subset of SWRL. Any concept description of OWL can be used in SWRL rules. Since SWRL provides the usage of more than two variables within the same scope and also built-in functions, rules can be defined in SWRL that have no OWL counterpart. For example, consider the rule

$$\mathsf{father\text{-}of}(?dad,\ ?kid) \wedge \mathsf{brother\text{-}of}(?dad,\ ?uncle) \rightarrow \mathsf{uncle\text{-}of}(?uncle,\ ?kid).$$

It says that each brother of a father is an uncle. As another example (regarding the examination regulations), consider a rule that explicitly stipulates the transitivity property of the ordering of the semesters,

$$\mathsf{precedes}(?s1,\ ?s2) \wedge \mathsf{precedes}(?s2,\ ?s3) \rightarrow \mathsf{precedes}(?s1,\ ?s3).$$

Or imagine we want to formulate that the sender and the receiver as the endpoints of a communication channel must access the communication channel with the same frequency in order to guarantee correct transmissions:

$$\begin{aligned}\mathsf{Sender}(?s) \wedge \mathsf{Receiver}(?r) \wedge \mathsf{Channel}(?c) \wedge \mathsf{endpoint}(?c,\ ?s) \wedge \mathsf{endpoint}(?c,\ ?r) \wedge \\ \mathsf{frequency}(?s,\ ?f1) \wedge \mathsf{frequency}(?r,\ ?f2) \rightarrow \mathsf{builtIn}(\mathsf{equals},\ ?f1,\ ?f2)\end{aligned}$$

It is impossible to modify these rules so that less than three variables are needed within the same scope. As a consequence, the latter three rules can't be described in OWL[5].

From a modelling point of view, the usage of SWRL is critical, since OWL-DL extended with rules becomes undecidable [4]. Thus, the modeller should use SWRL only in a way that preserves decidability of his model, but enables him to take advantage of the higher expressivity provided by SWRL rules. In [4] it is thus proposed to model as much as possible in OWL. As soon as SWRL is available for the modeller, however, he might tend to use the rule language more extensively than it might be necessary because modelling in a rule language seems to be easier than modelling in a concept language. Hence, SWRL variables might be used extensively, too, what makes finding corresponding OWL-DL counterparts for these rules more difficult. Therefore, additional research on SWRL should be done in order to restrict this rule language to a decidable subset, provided that this subset is still usable and convenient.

From the modelling standpoint, a reasoner that offers OWL-DL reasoning as well as SWRL-reasoning should be considered, so that the modeller need not to be aware of the different reasoners. From a reasoning point of view one can think of different arrangements of these two reasoners. The first approach is a one-pass-reasoning of the knowledge base where the two reasoner process the knowledge subsequently and only once. As both reasoner operate isolated, a repeated processing of the knowledge base is not necessary. In the more difficult variant, the reasoners know each other and make their reasoning results available to each other. There, both reasoning services are invoked subsequently and the reasoning loops as long as the knowledge base changes. The difficulty there is to guarantee that such a *repletion process* terminates. In all these variants conflict resolution is necessary, because it is an easy task to define concept restrictions and SWRL rules that contradict each other. These inconsistencies have to be detected and repaired in a convenient and reasonable manner.

[4]The usage of "anonymous" concept descriptions is permitted according to the W3C member submission but it is not provided by current implementations

[5]Remember that OWL is equal to the subset of first order logics without function symbols and only two variables within the same scope.

In some cases appropriate syntactic transformations of the results are necessary. As SWRL extends OWL the SWRL reasoner should be able to read the computed results by the OWL reasoner. The computed results of the SWRL reasoner, in turn, can be added in OWL by extending the instances with the SWRL-computed affiliations. As SWRL provides rules similar to views in database systems it is impossible to generate new instance data with SWRL. Existent instances (defined in OWL) are thereby affected and need to be modified. Furthermore, it is proposed to separate the OWL ontology from SWRL rules by extending the OWL annotations to import declarations of pure SWRL documents. OWL reasoners being unable to handle SWRL rules can then ignore such imports. This approach permits standalone OWL reasoners to process also OWL documents that may include results computed by SWRL reasoners according to imported SWRL rules, thus increasing the portability of the ontologies.

5 Summary

In this project thesis we have discussed why we should model contracts in OWL. After given some use cases we have agreed on the modelling of examination regulations as a small and easy-to-understand contract. We shortly introduced OWL and the family of description logics to which OWL is transformed in order to provide a well defined semantics. We have then developed a diagrammatic model that has been used to build up a knowledge base. Furthermore, we have discussed several proposals that might improve OWL.
Firstly, it has been already examined in different papers that OWL needs to provide customised datatypes. Restricting values to lie between certain boundaries and describing concepts according to values that lie between such boundaries it is possible to model huge quantities effectively.
Secondly, we have pointed out that it would be helpful to provide the facility of specifying arithmetic interrelations between attributes in OWL. Although arithmetics make reasoning undecidable, we have proposed to include arithmetic functions for specifying arithmetic interrelations between attributes. These interrelations would be resolved by an evaluation step prior to the reasoning. In this way the reasoning would remain unaffected but the modeller could specify his model completely in OWL. This might increase the modelling convenience, enlarge the fields of application and minimise errors that occur because of an erroneous usage of models specified in OWL.
Thirdly, we have explored that concepts relying on enumerated classes can be very cumbersome to describe. That's why we also proposed finite quantification to provide a shorter notation of concepts in which conditions for each nominal occur. Finite quantification could be naively implemented by a macro expansion that is done prior to the reasoning. Alternatively, it could be given a semantics in way that the reasoner could involve finite quantification in order to improve the reasoning performance. In any case, finite quantification would not increase the worst case complexity of the reasoning. We have also proposed a "variable-free" form of finite quantification since in OWL variables don't occur explicitly. Finite quantification could help the modeller to keep the knowledge base concise and to avoid errors.
Fourthly, we have examined SWRL, which is currently a W3C Member Submission. We have discussed the usage of SWRL and also have shown that SWRL is more expressive than OWL, illustrated by a few examples. We have pointed out that there are different arrangements of the SWRL and the OWL reasoner. We have given several possibilities of how the reasoners could be arranged and what kinds of knowledge base processing could be possible. We also have discussed extensions of OWL that would permit an isolated operation of each reasoner.
Finally, we have discussed the usage of qualified number restrictions. We have shown that qualified number restrictions permit a more intuitive and effective modelling. To explain why qualified number restrictions can computationally behave badly, we have shown that they are based on range restrictions of roles (which also behave badly). As proprietary OWL constructs for qualified number restrictions are already available and work arounds that are necessary due to the lack of qualified number restrictions lead to models that are not correct in

general we have proposed to add qualified number restrictions to the future OWL standards. This would increase the approval of OWL and permits modelling without error-prone work arounds.
In a future diploma thesis an OWL (or SWRL, resp.) reasoner that includes the improvals that are proposed in this project thesis will be developed. This reasoner shall be able to answer queries of that types given in the section 3.3.1.

6 Appendix A: The Examination Regulations for this Project Thesis

The examination regulations used in this project thesis are composed of the examination regulations for the bachelor's program of computer science at the TU München and of the generic examination regulations for the bachelor's study programs at the Ludwig-Maximilians-Universität München. Actually for this project thesis it is not important, which concrete examination regulations are used. These examination regulations at hand were chosen because they are kept easy and they are also of regional significance. The composed examination regulations for this project thesis do not include the following regulations concerning administration because they are not relevant for the modelling of a query system.

- §1 Scope, academic degree, related study program
- §2 Examination Purpose
- §4 Examination Board
- §5 Recognition of Credits from other Institutions
- §8 Study Requirements not covered by these Regulations
- §11 Admission Requirements for the Bachelor's Examination
- §18 Coming into Effect of these Regulations

The examination regulations of the TU München are modified according to the generic examination regulations for the bachelor's study programs at the LMU München for two reasons. Firstly, the regulations of the TU München do not completely satisfy current obligations that examination regulations for bachelor's study programs should. Secondly, the examination regulations used in this project thesis should be similar to the examination regulations of the bachelor's study program of computer science at LMU München that will be introduced in the near future. But regulations that do not fit the obligations given in the generic examination regulations of the LMU München are accepted if they allow more interesting questions concerning modelling.

6.1 Relevant Aspects of the Examination Regulations

In this section the examination regulations used in this project thesis will be introduced in detail.

6.1.1 Acquiring Credits

The study program comprises 180 ECTS points (credits, abbreviated as cr). They are to be acquired in modules of the module-catalogue **A** (obligatory and mandatory modules in computer science), the module-catalogue **B** (elective modules in computer science for the advanced study period), module-catalogue **C** (obligatory and mandatory modules in mathematics), the module-catalgue **D** (elective modules in additional topics and soft skills), the module-catalogue **E** (elective modules in a chosen minor subject) and the bachelor's examination. The bachelor's examination again is divided into the bachelor's thesis and its defense. So, the ECTS points split up according to the scheme pictured below. The module-catalogues will be described in the following sections.

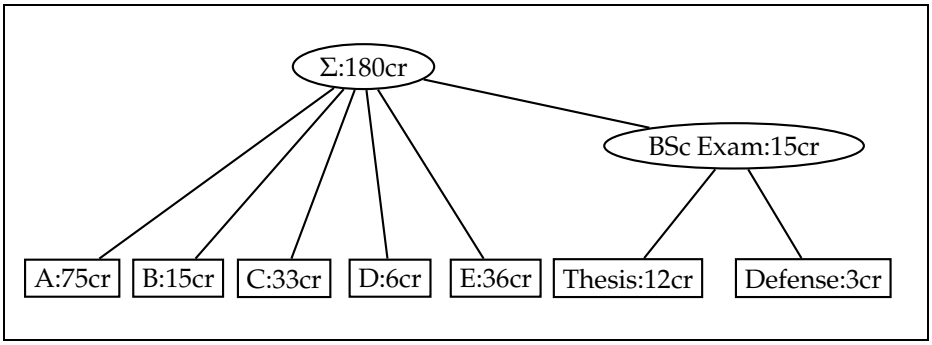


Figure 6.14: Acquisition Scheme

6.1.2 The Structure of the Module-Catalogue A

The module-catalogue **A** comprises the following modules for basic and introductory topics of computer science.

id	module	cr	type
IN0001	Foundations of Computer Science 1	9cr	obligatory, basic module
IN0003	Foundations of Computer Science 2	6cr	obligatory, basic module
IN0004	Technical Foundations of Computer Science	9cr	oligatory, basic module
IN0011	Introduction to Software Engineering	6cr	obligatory, basic module
IN0007	Foundations of Algorithms and Data Structures	9cr	obligatory, basic module
IN0008	Foundations of Databases	6cr	obligatory
IN0013	Proseminar	3cr	obligatory
IN0014	Seminar	9cr	obligatory

In addition, modules that amount to at least **18cr** have to be chosen from the following selection.

id	module	cr	type
IN0002	Practical: Foundations of Programming	6cr	mandatory
IN0005	Practical: Technical Foundations of Computer Science	9cr	mandatory
IN0011	Introduction to Theoretical Computer Science	9cr	mandatory
IN0009	Operating Systems and System Software	6cr	mandatory
IN0010	Computer Networks and Distributed Systems	6cr	mandatory

Students have to acquire at least **75cr** in modules of the module-catalogue **A**.

6.1.3 The Structure of the Module-Catalogue B

The module-catalogue **B** comprises elective modules of computer science for the advanced study period and is subdivided into so-called module-groups which are

B1 Software Engineering

B2 Databases and Information Systems

B3 Artificial Intelligence and Robotics

B4 Computer Graphics and Image Processing

B5 Computer Architecture

B6 Distributed Systems and Computer Networks

B7 Formal Methods and its Applications

B8 Algorithms and Scientific Computing

These module-groups partition modules according to their topics and chronology. The following tables give an overview of the module-groups.

id	module	cr	type
Modules of the module-group B1 - Software Engineering			
IN2078	Foundations of Program and System Development	6	elective
IN2079	IT Consulting	3cr	elective
IN2080	Modelling of Distributed Systems	4cr	elective
IN2081	Design Patterns in Software Engineering	5cr	elective
IN2082	Project Management	5cr	elective
IN2085	Software Engineering for Business Applications	5cr	elective
IN2088	Software Architectures	3cr	elective
IN2090	Distributed Problem Solving	3cr	elective
Modules of the module-group B2 - Databases and Information Systems			
IN2028	Business Analytics	5cr	elective
IN2029	Data Mining II: Clustering	3cr	elective
IN2030	Data Mining and Knowledge Discovery	3cr	elective
IN2031	Use and Realisation of Database Systems	6cr	elective
IN2032	Electronic Publishing	5cr	elective
IN2033	Information Management	5cr	elective
IN2038	Distributed Information Systems for the Web	6cr	elective
IN2039	XML Information Systems	6cr	elective
Modules of the module-group B3 - Artificial Intelligence and Robotics			
IN2057	3D Computer Vision	8cr	elective
IN2058	Application of Knowledge Based Methods	5cr	elective
IN2059	Autonomous Systems	4cr	elective
IN2060	Realtime Systems	6cr	elective
IN2061	Introduction to Digital Signal Processing	5cr	elective
IN2062	Foundations of Artifical Intelligence	5cr	elective
IN2067	Robotics	6cr	elective
IN2068	Sensor-controlled Robots	1cr	elective

id	module	cr	type
Modules of the module-group B4 - Computer Graphics and Image Processing			
IN2015	Image Synthesis	4cr	elective
IN2016	Computer Vision	4cr	elective
IN2017	Computer Graphics	6cr	elective
IN2018	Augmented Reality I	3cr	elective
IN2019	Augmented Reality II	3cr	elective
IN2021	Application of Computer Science in Medicine I	4cr	elective
IN2022	Application of Computer Science in Medicine II	4cr	elective
IN2025	Simulation and Animation	4cr	elective
Modules of the module-group B5 - Computer Architecture			
IN2072	Analysis of System Performance	6cr	elective
IN2073	Grid Computing	3cr	elective
IN2074	High Performance Computer	4cr	elective
IN2075	Microprocessors	4cr	elective
IN2076	Computer Architecture	4cr	elective
IN2077	Scalable Parallel Computers with Shared Memory	4cr	elective
Modules of the module-group B6 - Distributed Systems and Computer Networks			
IN2091	Automated Management of Networked and Concurrent Systems	4cr	elective
IN2093	eLearning - Techniques and Infrastructures	3cr	elective
IN2095	Internet Protocols	5cr	elective
IN2098	Mobile Distributed Systems	5cr	elective
IN2099	Network and System Management	4cr	elective
IN2100	Secure Computing Systems	5cr	elective
IN2101	Security of Networked Systems	3cr	elective
IN2102	Distributed Applications	4cr	elective
Modules of the module-group B7 - Formal Methods and its Applications			
IN2040	Abstract Machines	5cr	elective
IN2041	Automata and Formal Languages I	8cr	elective
IN2042	Automata and Formal Languages II	8cr	elective
IN2044	Compiler Construction	8cr	elective
IN2049	Logics	8cr	elective
IN2050	Model Checking	8cr	elective
IN2055	Semantics	8cr	elective
IN2056	Temporal Logics	3cr	elective
Modules of the module-group B8 - Algorithms and Scientific Computing			
IN2001	Algorithms of Scientific Computing I	8cr	elective
IN2002	Algorithms of Scientific Computing II	8cr	elective
IN2003	Efficient Algorithms and Data Structures I	8cr	elective
IN2004	Efficient Algorithms and Data Structures II	8cr	elective
IN2005	Introduction to Scientific Computing	3cr	elective
IN2007	Cryptology I	3cr	elective
IN2008	Cryptology II	3cr	elective

Students have to acquire at least **15cr** in modules of the module-catalogue **B**.

6.1.4 The Structure of the Module-Catalogue C

The module-catalogue **C** comprises the following modules for basic and introductory topics of mathematics.

id	module	cr	type
IN0015	Discrete Mathematics	9cr	obligatory, basic module
MA0901	Linear Algebra	9cr	obligatory, basic module
MA0902	Analysis	9cr	obligatory, basic module

In addition, one of the two following modules are to be selected.

id	module	cr	type
IN0018	Theory of Discrete Probability	6cr	mandatory
IN0019	Numeric Methods	6cr	mandatory

Students have to acquire at least **33cr** in modules of the module-catalogue **C**.

6.1.5 The Structure of the Module-Catalogue D

The module-catalogue **D** comprises the following modules for basic soft-skills and additional topics concerning computer science. In module-catalogue D one has to select at least two modules.

id	module	cr	type
IN9001	Communication in Team	3cr	elective
IN9002	Public Relations	3cr	elective
IN9003	Computer Science Law	3cr	elective
IN9004	Ethics and Responsibility	3cr	elective
IN9005	Business Plans, Marketing and Finances	3cr	elective
IN9006	Foundation and Management of Small Software Oriented Enterprises	3cr	elective
IN9008	Acting with Incomplete Information	3cr	elective
IN9010	Leading Oneself and Colleagues	3cr	elective
IN9011	Seminar: Scientists and Ethics	3cr	elective
IN9012	Seminar: Project Management	3cr	elective

Students have to acquire at least **6cr** in modules of the module-catalogue **D**.

6.1.6 The Structure of the Module-Catalogue E

The module-catalogue **E** comprises the modules for mandatory minor subjects. The minor subjects are neglected in this thesis. Within the first four semesters 15cr have to be acquired in modules of the module-catalogue **E**.

6.1.7 Further Remarks on the Examination Regulations

Examinations of modules that are denoted "basic module" have to be taken within the first four semesters. Otherwise they will be regarded as first-time-failures. Basic modules provide basic knowledge needed for subsequent modules, especially those of the advanced study period. To make sure that students have the necessary knowledge needed for those subsequent modules the examinations of basic modules must be passed within a certain time.

Furthermore, students must acquire at least 54cr within the first four semesters and at least 108cr within the first six semesters to make sure that they are able to finish their studies within nine semesters. If a student has not acquired 54cr within the first four semester or 108cr within the first six semesters he ultimately fails the bachelor's examination.
For each module examination passed by the student the examination board has to register the acquired credits as bonus credits. For each module re-examination the student fails these credits are to be registered as malus credits. If a student has acquired more than 90cr as malus credits he may no longer participate in second module re-examinations.
If a student fails a module examination he must participate in the re-examination to the next possible appointment. Otherwise the re-examination will be regarded as a failure. For each module at most two re-examinations are permitted. To make studying a bit more interesting (so one can expect more interesting queries) students must satisfy certain prerequisites for modules of the module-groups of the module-catalogue B. These prerequisites are listed below.

module-group (catalogue B)	**prerequisite**
Software-Engineering	Practical: Foundations of Programming Introduction to Theoretical Computer Science
Databases and Informaton Systems	Operating Systems and System Software Computer Networks and Distributed Systems
Artificial Intelligence and Robotics	Practical: Technical Foundations of Computer Science Introduction to Theoretical Computer Science Numeric Methods
Computer Graphics and Image Processing	Operating Systems and System Software Theory of Discrete Probability Numeric Methods
Computer Architecture	Practical: Technical Foundations of Computer Science Operating Systems and System Software
Distributed Systems and Computer Networks	Computer Networks and Distributed Systems Practical: Introduction to Programming
Formal Methods and its Applications	Practical: Introduction to Programming Introduction to Theoretical Computer Science
Algorithms and Scientific Computing	Practical: Introduction to Programming Numeric Methods Theory of Discrete Probability

Module examinations from the module catalogues **B** and **D** are replaceable, that is if a student fails a module examination he can replace this examination with another one from the same module catalogue.
Having acquired the necessary credits in modules from module-catalogue **B** the student may attend more modules which will then be registered as additional modules.

6.2 Recommended Generic Study Schedule

For a student it is always very useful to have a generic study schedule giving a student recommendations and preventing the student from making mistakes in planning his individual schedule. The recommended generic study schedule for this examination regulations at hand is as follows.

1st semester	Foundations of Computer Science I Technical Foundations of Computer Science Linear Algebra Discrete Mathematics
2nd semester	Foundations of Computer Science II Introduction to Software Engineering Foundations of Algorithms and Data Structures Analysis modules of module-catalogue D modules of module-catalogue E (approx. 9cr)
3rd semester	Practical: Foundations of Programming Numeric Methods Foundations of Databases Proseminar modules of the module-catalogue D modules of the module-catalogue E (approx. 9cr)
4th semester	Introduction to Theoretical Computer Science Practical: Technical Foundations of Computer Science Theory of Discrete Probability Operating Systems and System Software Computer Networks and Distributed Systems modules of the module-catalogue D modules of the module-catalgoue E (approx. 9cr)
5th semester	mandatory modules of module-catalogue B modules of the module-catalogue E Seminar
6th semester	mandatory modules of module-catalogue B Bachelor's Examination

6.3 Compatibility Constraints

The examination regulations of this project thesis do not satisfy all rules and proposals of the generic examination regulations for the bachelor's study programs at the Ludwig-Maximilians-Universität München. The most significant deviations are listed below.

- The bachelor's examination also comprises a defense of the bachelor's thesis.
- Examinations of basic modules may be repeated twice.
- The examination board also registers malus credits.
- Explicit prerequisites for the module-groups of the module-catalogue B are introduced.
- Students have to acquire certain amounts of credits within defined periods.

7 Appendix B: The Knowledge Base for the Examination Regulations

In this section we will give the knowledge base that was developed to specify the model of the examination regulations. For the development of the knowledge base we will use the improvals that are proposed in this thesis. Roles are defined according to figure 2.2, so that they can be directly translated into OWL. For concisement matter the instances of concepts and roles regarding exams are omitted. They can be found in the sections 6.1.2, 6.1.3, 6.1.4, 6.1.5 and 6.1.6.

7.1 The Concept Semester and the Roles Precedes and Dates-Result

The semester-concept can be defined as an eumerated class:

$$\mathsf{Semester} \equiv \{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9\}.$$

The role precedes defines an order of semesters:

$$\geq 1\,\mathsf{precedes} \sqsubseteq \mathsf{Semester}$$
$$\top \sqsubseteq \forall \mathsf{precedes}.\mathsf{Semester}$$

The following equations must hold:

$$\begin{aligned}
\mathsf{Semester} \sqcap \exists \mathsf{precedes}.\{s_1\} &\equiv \bot \\
\mathsf{Semester} \sqcap \exists \mathsf{precedes}.\{s_2\} &\equiv \{s_1\} \\
\mathsf{Semester} \sqcap \exists \mathsf{precedes}.\{s_3\} &\equiv \{s_2\} \\
\mathsf{Semester} \sqcap \exists \mathsf{precedes}.\{s_4\} &\equiv \{s_3\} \\
\mathsf{Semester} \sqcap \exists \mathsf{precedes}.\{s_5\} &\equiv \{s_4\} \\
\mathsf{Semester} \sqcap \exists \mathsf{precedes}.\{s_6\} &\equiv \{s_5\} \\
\mathsf{Semester} \sqcap \exists \mathsf{precedes}.\{s_7\} &\equiv \{s_6\} \\
\mathsf{Semester} \sqcap \exists \mathsf{precedes}.\{s_8\} &\equiv \{s_7\} \\
\mathsf{Semester} \sqcap \exists \mathsf{precedes}.\{s_9\} &\equiv \{s_8\} \\
\{s_9\} \sqcap \geq 1\,\mathsf{precedes} &\equiv \bot
\end{aligned}$$

The role has-score relates each semester to an instance of BonusAccount as well as to an instance of MalusAccount

$$\geq 1\,\mathsf{has\text{-}score} \sqsubseteq \mathsf{Semester}$$
$$\top \sqsubseteq \forall \mathsf{has\text{-}score}.(\mathsf{BonusAccount} \sqcup \mathsf{MalusAccount})$$

The role dates-result relates each semester to those instances of ExamResult that are considered to be acquired in the corresponding semester.

$$\geq 1\,\mathsf{dates\text{-}result} \sqsubseteq \mathsf{Semester}$$
$$\top \sqsubseteq \forall \mathsf{dates\text{-}result}.\mathsf{ExamResult}$$

7.2 The Credit Accounts

The concept BonusAccount is related to the following data-valued roles that model the acquired credits in each module catalogue.

$$\geq 1\,\mathsf{crObligatoryA} \sqsubseteq \mathsf{BonusAccount}$$
$$\top \sqsubseteq \forall \mathsf{crObligatoryA.xsd{:}NonNegativeInteger}$$

$$\geq 1\,\mathsf{crMandatoryA} \sqsubseteq \mathsf{BonusAccount}$$
$$\top \sqsubseteq \forall \mathsf{crMandatoryA.xsd{:}NonNegativeInteger}$$

$$\geq 1\,\mathsf{crElectiveB} \sqsubseteq \mathsf{BonusAccount}$$
$$\top \sqsubseteq \forall \mathsf{crElectiveB.xsd{:}NonNegativeInteger}$$

$$\geq 1\,\mathsf{crObligatoryC} \sqsubseteq \mathsf{BonusAccount}$$
$$\top \sqsubseteq \forall \mathsf{crObligatoryC.xsd{:}NonNegativeInteger}$$

$$\geq 1\,\mathsf{crMandatoryC} \sqsubseteq \mathsf{BonusAccount}$$
$$\top \sqsubseteq \forall \mathsf{crMandatoryC.xsd{:}NonNegativeInteger}$$

$$\geq 1\,\mathsf{crElectiveD} \sqsubseteq \mathsf{BonusAccount}$$
$$\top \sqsubseteq \forall \mathsf{crElectiveD.xsd{:}NonNegativeInteger}$$

$$\geq 1\,\mathsf{crElectiveE} \sqsubseteq \mathsf{BonusAccount}$$
$$\top \sqsubseteq \forall \mathsf{crElectiveE.xsd{:}NonNegativeInteger}$$

$$\geq 1\,\mathsf{crThesis} \sqsubseteq \mathsf{BonusAccount}$$
$$\top \sqsubseteq \forall \mathsf{crThesis.xsd{:}NonNegativeInteger}$$

$$\geq 1\,\mathsf{crDefense} \sqsubseteq \mathsf{BonusAccount}$$
$$\top \sqsubseteq \forall \mathsf{crDefense.xsd{:}NonNegativeInteger}$$

Additionally, the role crSum is modelled to contain the sum of all acquired bonus credits. For the modelling we use the proposal made in section 4.2.

$$\begin{aligned} \geq 1\,\mathsf{crSum} &\sqsubseteq \mathsf{BonusAccount} \\ \top &\sqsubseteq \forall \mathsf{crDefense.xsd{:}NonNegativeInteger} \\ \mathsf{crSum} &\equiv \mathsf{Sum}(\mathsf{crDefense}, \mathsf{crThesis}, \mathsf{crElectiveE}, \\ &\qquad \mathsf{crElectiveD}, \mathsf{crMandatoryC}, \mathsf{crObligatoryC}, \\ &\qquad \mathsf{crElectiveB}, \mathsf{crMandatoryA}, \mathsf{crObligatoryA}) \end{aligned}$$

For the concept MalusAccount only the role credits is needed:

$$\geq 1\,\mathsf{credits} \sqsubseteq \mathsf{Semester}$$
$$\top \sqsubseteq \forall \mathsf{credits.xsd{:}NonNegativeInteger}$$

7.3 The Concept of Exam Results

The concept ExamResult models examination results. Examination results can be subdivided into

$$\mathsf{ExamResult} \equiv \mathsf{GoodResult} \sqcup \mathsf{BadResult}.$$

where the following conditions hold.

$$\begin{aligned} \mathsf{BadResult} &\equiv \mathsf{ExamResult} \sqcap \exists \mathsf{grade}.\ >_4 \\ \mathsf{GoodResult} &\equiv \mathsf{ExamResult} \sqcap \exists \mathsf{grade}.\ \leq_4 \end{aligned}$$

In the definition above we used customised datatypes, see section 4.1.

The role grade models the grade a exam can have as a result.

$$\begin{aligned} \geq 1\,\mathsf{grade} &\sqsubseteq \mathsf{ExamResult} \\ \top &\sqsubseteq \forall \mathsf{grade.xsd{:}NonNegativeInteger} \end{aligned}$$

7.4 The Concept of Exams

The concept Exam is subdivided as follows.

$$\begin{aligned} \mathsf{Exam} &\equiv \mathsf{ModuleAExam} \sqcup \mathsf{ModuleBEam} \sqcup \mathsf{ModuleCExam} \sqcup \mathsf{ModuleDExam} \\ &\quad\ \mathsf{ModuleEExam} \sqcup \mathsf{BachelorsExam} \\ \\ \mathsf{ModuleAExam} &\equiv \mathsf{ObligatoryModuleAExam} \sqcup \mathsf{MandatoryModuleAExam} \\ \mathsf{ModuleBExam} &\equiv \mathsf{ModuleGroupB1Exam} \sqcup \mathsf{ModuleGroupB2Exam} \sqcup \\ &\quad\ \mathsf{ModuleGroupB3Exam} \sqcup \mathsf{ModuleGroupB4Exam} \sqcup \\ &\quad\ \mathsf{ModuleGroupB5Exam} \sqcup \mathsf{ModuleGroupB6Exam} \sqcup \\ &\quad\ \mathsf{ModuleGroupB7Exam} \sqcup \mathsf{ModuleGroupB8Exam} \\ \\ \mathsf{BachelorsExam} &\equiv \mathsf{Thesis} \sqcup \mathsf{Defense} \\ \\ \mathsf{Thesis} &\equiv \{\mathsf{thesis}\} \\ \mathsf{Defense} &\equiv \{\mathsf{defense}\} \end{aligned}$$

The instances of the exams are already given in the sections 6.1.2, 6.1.3, 6.1.4, 6.1.5 and 6.1.6. There the id is considered to be the name of the instance. For example, see the concept BasicModuleExam below.

The role course relates exams to their course-names.

$$\begin{aligned} \geq 1\,\mathsf{course} &\sqsubseteq \mathsf{Exam} \\ \top &\sqsubseteq \forall \mathsf{course.xsd{:}String} \end{aligned}$$

The role ultimo relates exams to their semesters in which the exam has to be taken at the latest.

$$\begin{aligned} \geq 1\,\mathsf{ultimo} &\sqsubseteq \mathsf{Exam} \\ \top &\sqsubseteq \forall \mathsf{course.xsd{:}NonNegativeInteger} \end{aligned}$$

The role credits relates exams to the credits that can be acquired by passing them.

$$\geq 1\,\text{credits} \sqsubseteq \text{Exam}$$
$$\top \sqsubseteq \forall \text{credits.xsd:NonNegativeInteger}$$

The role has-result relates exams to their exam results, if there are any.

$$\exists \text{has-result} \sqsubseteq \text{Exam}$$
$$\top \sqsubseteq \forall \text{has-result.ExamResult}$$

Additionally, the following sub-concepts are identified.

$$\begin{aligned} \text{TakenExam} &\sqsubseteq \text{Exam} \\ \text{RepeatableExam} &\sqsubseteq \text{Exam} \\ \text{BasicModuleExam} &\sqsubseteq \text{Exam} \\ \text{ReplaceableModuleExam} &\sqsubseteq \text{Exam} \end{aligned}$$

They can be described as follows.

$$\begin{aligned} \text{TakenExam} &\equiv \text{PassedExam} \sqcup \text{FailedExam} \\ \text{PassedExam} &\equiv \text{Exam} \sqcup \exists \text{has-result.GoodResult} \\ \text{FailedExam} &\equiv \text{Exam} \sqcup \exists \text{has-result.BadResult} \\ \text{RepeatableExam} &\equiv \text{Exam} \sqcap \leq 3\,\text{has-result} \\ \text{BasicModuleExam} &\equiv \{\text{in0001}, \text{in0003}, \text{in0004}, \text{in0006}, \\ &\quad \text{in0007}, \text{in0015}, \text{ma0901}, \text{ma0902}\}. \\ \text{ReplaceableModuleExam} &\equiv \text{ModuleBExam} \sqcup \text{ModuleDExam} \end{aligned}$$

7.5 The Concepts of Students and their Subsumption Hierarchy

The following concepts are used to structure students.

Student classes	**Concepts**
Students in the basic study period	BSP
Students in the advances study period	ASP
Students who only lack the bachelor's examination	OnlyBEMissing
Students who have finished studies	Finished
Students who have not taken all basic module exams within the first two semesters	NTakenBME-Before3rd
Students who still have not taken all exams within the first six semesters for the first time	NTakenExams-Before7th
Students in the 4th semester with less than 54 bonus credits acquired	LT-54cr-Sem4
Students in the 6th semester with less than 108 bonus credits acquired	LT-108cr-Sem6
Students not taking re-exams to the next appointment	NTaken-ReExams
Students who are excluded from second re-exams	Excluded-2nd-ReExams
Students who have not already passed all elective modules	EM-Remain-Sem9
Students who beginning the bachelor's exam in the 6th semester	BE-In-Sem6
Students who fail the bachelors's exam for the first time	BE-First-Failure
Students who fail the bachelors's exam for the second time	BE-Second-Failure
Students who ultimately fail the bachelor's exam	UF
Regular students	RS

The role obligatory-exam relates the student to its obligatory exams.

$$\geq 1\,\text{obligatory-exam} \sqsubseteq \text{Student}$$
$$\top \sqsubseteq \forall \text{obligatory-exam}.(\text{ObligatoryModuleAExam} \sqcup \text{ObligatoryModuleCExam} \sqcup \text{BachelorsExam})$$

The subsumption hierarchy is as follows.

$$
\begin{aligned}
\text{BSP} &\sqsubseteq \text{RS}\\
\text{ASP} &\sqsubseteq \text{RS}\\
\text{OnlyBE-Missing} &\sqsubseteq \text{ASP}\\
\text{Finished} &\sqsubseteq \text{ASP}\\
\text{NTakenBME-Until3rd} &\sqsubseteq \text{BSP}\\
\text{NTakenExams-Until7th} &\sqsubseteq \text{ASP}\\
\text{LT-54cr-Sem4} &\sqsubseteq \text{UF}\\
\text{LT-108cr-Sem6} &\sqsubseteq \text{UF}\\
\text{NTaken-ReExam} &\sqsubseteq \text{BSP} \sqcup \text{ASP}\\
\text{Excluded-2nd-ReExam} &\sqsubseteq \text{BSP} \sqcup \text{ASP}\\
\text{EM-ToBePassed-Sem9} &\sqsubseteq \text{ASP}\\
\text{BE-BegunIn-Sem6} &\sqsubseteq \text{ASP}\\
\text{BE-First-Failure} &\sqsubseteq \text{ASP}\\
\text{BE-Second-Failure} &\sqsubseteq \text{UF}
\end{aligned}
$$

$$
\begin{aligned}
\text{RS} &\equiv \text{ASP} \sqcup \text{BSP} \sqcup \text{Finished}\\
\text{Student} &\equiv \text{UF} \sqcup \text{RS}
\end{aligned}
$$

These concepts can be described as follows.

$$
\begin{aligned}
\text{BSP} &\equiv \text{Student} \sqcap \forall \text{enrolled-in.}(\text{Semester} \sqcap \text{has-value.} \leq_4)\\
\text{ASP} &\equiv \text{Student} \sqcap \exists \text{enrolled-in.}(\text{Semester} \sqcap \text{has-value.} >_4)\\
\text{OnlyBEMissing} &\equiv \text{ASP} \sqcap \exists \text{enrolled-in.}(\text{Semester} \sqcap\\
&\quad \exists \text{bonus.}(\text{BonusAccount} \sqcap\\
&\quad \exists \text{crObligatoryA.} \geq_{54} \sqcap \exists \text{crMandatoryA.} \geq_{15} \sqcap\\
&\quad \exists \text{crElectiveB.} \geq_{15} \sqcap \exists \text{crObligatoryC.} \geq_6 \sqcap\\
&\quad \exists \text{crMandatoryC.} \geq_6 \sqcap \exists \text{crElectiveD.} \geq_6 \sqcap\\
&\quad \exists \text{crElective.} \geq 36) \sqcap\\
&\quad \forall \text{bonus.}(\text{BonusAccount} \sqcap \exists \text{thesis.} =_0 \sqcap \exists \text{defense.} =_0))\\
\text{Finished} &\equiv \text{ASP} \sqcap \exists \text{enrolled-in.}(\text{Semester} \sqcap\\
&\quad \exists \text{bonus.}(\text{BonusAccount} \sqcap\\
&\quad \exists \text{crObligatoryA.} \geq_{54} \sqcap \exists \text{crMandatoryA.} \geq_{15} \sqcap\\
&\quad \exists \text{crElectiveB.} \geq_{15} \sqcap \exists \text{crObligatoryC.} \geq_6 \sqcap\\
&\quad \exists \text{crMandatoryC.} \geq_6 \sqcap \exists \text{crElectiveD.} \geq_6 \sqcap\\
&\quad \exists \text{crElective.} \geq 36 \sqcap \exists \text{crThesis.} =_{12}\\
&\quad \exists \text{crDefense.} =_3))\\
\text{NTakenBME-Until3} &\equiv \text{BSP} \sqcap \exists \text{enrolled-in.}(\text{Semester} \sqcap\\
&\quad \exists \text{has-value.} <_3 \sqcap \neg \text{HasTakenAllBasicModules},
\end{aligned}
$$

with the concept description (using the improval proposed in section 4.3)

$$
\begin{aligned}
\text{HasTakenAllBasicModules} \equiv \text{Student} \sqcap &\prod_{\text{BasicModuleExam}} \exists \text{enrolled-in.}(\text{Semester} \sqcap\\
&\exists \text{dates-result.}(\text{ExamResult} \sqcap \exists \text{has-result}^{-}.\Delta)).
\end{aligned}
$$

$$\begin{aligned}\text{NTaken-Exams-Until7} \equiv\ & \text{Student} \sqcap \forall \text{enrolled-in.}(\text{Semester} \sqcap \\ & \exists \text{has-number.} <_7) \sqcap \neg \text{AllExamsTaken}\end{aligned}$$

with the concept description (using the improval proposed in section 4.3)

$$\begin{aligned}\text{AllExamsTaken} \equiv\ & \text{Student} \sqcap \prod_{\text{Exam}} \exists \text{enrolled-in.}(\text{Semester} \sqcap \\ & \exists \text{dates-result.}(\text{ExamResult} \sqcap \exists \text{has-result}^{-}.\Delta))\end{aligned}$$

$$\begin{aligned}\text{LT-54cr-Sem4} \equiv\ & \text{Student} \sqcap \exists \text{enrolled-in.}(\text{Semester} \sqcap \exists \text{has-number.} =_4 \sqcap \\ & \exists \text{bonus.}(\text{BonusAccount} \sqcap \exists \text{crSum.} <_{54}))\end{aligned}$$

$$\begin{aligned}\text{LT-108cr-Sem6} \equiv\ & \text{Student} \sqcap \exists \text{enrolled-in.}(\text{Semester} \sqcap \exists \text{has-number.} =_6 \sqcap \\ & \exists \text{bonus.}(\text{BonusAccount} \sqcap \exists \text{crSum.} <_{108}))\end{aligned}$$

$$\begin{aligned}\text{Excluded-2nd-ReExams} \equiv\ & \text{Student} \sqcap \exists \text{enrolled-in.}(\text{Semester} \sqcap \\ & \exists \text{has-score.}(\text{MalusAccount} \sqcap \exists \text{credits.} >_{52}))\end{aligned}$$

$$\begin{aligned}\text{EM-Remain-Sem9} \equiv\ & \text{Student} \sqcap \exists \text{enrolled-in.}(\text{Semester} \sqcap \\ & \exists \text{has-number.} =_9 \sqcap \exists \text{has-score.}(\text{BonusAccount} \sqcap \\ & (\exists \text{crElectiveB.} <_{15} \sqcup \exists \text{crElectiveD.} <_6 \sqcup \exists \text{crElectiveE.} <_{36})))\end{aligned}$$

$$\begin{aligned}\text{BE-In-Sem6} \equiv\ & \text{ASP} \sqcap \exists \text{enrolled-in.}(\text{Semester} \sqcap \\ & \exists \text{has-number.} =_6 \sqcap \exists \text{dates-result.BachelorsExam})\end{aligned}$$

$$\text{BE-First-Failure} \equiv \text{Student} \sqcap \exists \text{obligatory-exam.}(\{\text{thesis}\} \sqcap = 1\, \text{has-result}^{-})$$

$$\text{BE-Second-Failure} \equiv \text{Student} \sqcap \exists \text{obligatory-exam.}(\{\text{thesis}\} \sqcap = 2\, \text{has-result}^{-})$$

8 Appendix C: Formalising a Concrete Study Schedule

In this section we will give a formalisation of a concrete study schedule in order to see how an instantiation of our model looks like. As the concrete schedule we will use the one of Hans, the student that was introduced in the section 3.3.1. As mentioned above, Hans has been already studying for three semesters and has passed the following module exams.

1st semester	Foundations of Computer Science I Technical Foundations of Computer Science Discrete Mathematics Linear Algebra *acquired credits: 36*
2nd semester	Foundations of Computer Science II Introduction to Software Engineering Foundations of Algorithms and Data Structures Analysis Computer Science Law *acquired credits: 33, total: 69*

Furthermore, we suppose that Hans has failed the following exams.

1st semester	Foundations of Databases Operating Systems and Systems Software *acquired malus credits: 0*
2nd semester	Foundations of Databases Operating Systems ans System Software *acquired malus credits: 12, total malus score: 12*

According to our knowledge about Hans, we might say that Hans is a student in the basic study period (this classification would be done by the reasoner).

Hans : BSP

Hans has to pass the following obligatory exams during his entire studies. Here again, we represent the obligatory exams by the identifiers of their corresponding modules (cf. 6.1.2 to 6.1.5). The list of obligatory exams Hans has to pass is necessary to allow propositions about exams that have not yet been taken by Hans. Without such a list our model allows only propositions about exams that Hans has already taken.

(Hans, in0001) : obligatory-exam
(Hans, in0003) : obligatory-exam
(Hans, in0004) : obligatory-exam
(Hans, in0006) : obligatory-exam
(Hans, in0007) : obligatory-exam
(Hans, in0008) : obligatory-exam
(Hans, in0013) : obligatory-exam
(Hans, in0014) : obligatory-exam
(Hans, in0015) : obligatory-exam
(Hans, ma0901) : obligatory-exam
(Hans, ma0902) : obligatory-exam
(Hans, thesis) : obligatory-exam
(Hans, defense) : obligatory-exam

Hans is currently in the third semester, that is he passed the first two semesters.

$(\mathsf{Hans}, \mathsf{s}_1)$: enrolled-in
$(\mathsf{Hans}, \mathsf{s}_2)$: enrolled-in

For each semester there is a bonus credit account as well as a malus credit account.

bonus_1 : BonusAccount
malus_1 : MalusAccount
bonus_2 : BonusAccount
malus_2 : MalusAccount

$(\mathsf{s}_1, \mathsf{bonus}_1)$: has-score
$(\mathsf{s}_1, \mathsf{malus}_1)$: has-score
$(\mathsf{s}_2, \mathsf{bonus}_2)$: has-score
$(\mathsf{s}_2, \mathsf{malus}_2)$: has-score

While in the first semester Hans has acquired 36 bonus credits he hasn't acquired any malus credits[6].

$(\mathsf{bonus}_1, 36)$: crAbove6thSemester
$(\mathsf{bonus}_1, 36)$: crBelow6thSemester
$(\mathsf{bonus}_1, 0)$: crDefense
$(\mathsf{bonus}_1, 0)$: crThesis
$(\mathsf{bonus}_1, 0)$: crElectiveD
$(\mathsf{bonus}_1, 0)$: crMandatoryC
$(\mathsf{bonus}_1, 18)$: crObligatoryC
$(\mathsf{bonus}_1, 0)$: crElectiveB
$(\mathsf{bonus}_1, 0)$: crMandatoryA
$(\mathsf{bonus}_1, 18)$: crObligatoryA

$(\mathsf{malus}_1, 0)$: credits

The scores of the accounts in the second semester are as follows.

$(\mathsf{bonus}_2, 69)$: crAbove6thSemester
$(\mathsf{bonus}_2, 69)$: crBelow6thSemester
$(\mathsf{bonus}_2, 0)$: crDefense
$(\mathsf{bonus}_2, 0)$: crThesis
$(\mathsf{bonus}_2, 3)$: crElectiveD
$(\mathsf{bonus}_2, 0)$: crMandatoryC
$(\mathsf{bonus}_2, 27)$: crObligatoryC
$(\mathsf{bonus}_2, 0)$: crElectiveB
$(\mathsf{bonus}_2, 0)$: crMandatoryA
$(\mathsf{bonus}_2, 39)$: crObligatoryA

$(\mathsf{malus}_2, 12)$: credits

The exam results are dated by the semester in which they were acquired. Additionally, these results are related to the exams to which they belong. But before we can instantiate the appropriate roles we have to define the instances of exam result. Again, we will classify the results to the most specialised classes, what is usually done by the reasoner.

[6] Malus credits are acquired by failing re-examinations.

res0-in0001 : GoodResult
res0-in0004 : GoodResult
res0-in0015 : GoodResult
res0-ma0901 : GoodResult
res0-in0008 : BadResult
res0-in0009 : BadResult

res0-in0003 : GoodResult
res0-in0006 : GoodResult
res0-in0007 : GoodResult
res0-in9003 : GoodResult
res0-ma0902 : GoodResult
res1-in0008 : BadResult
res1-in0009 : BadResult

Having defined the instances of examination result, we can relate the semesters to their exam results.

$(s_1, \text{res0-in0001})$: dates-result
$(s_1, \text{res0-in0004})$: dates-result
$(s_1, \text{res0-in0015})$: dates-result
$(s_1, \text{res0-ma0901})$: dates-result
$(s_1, \text{res0-in0008})$: dates-result
$(s_1, \text{res0-in0009})$: dates-result

$(s_2, \text{res0-in0003})$: dates-result
$(s_2, \text{res0-in0006})$: dates-result
$(s_2, \text{res0-in0007})$: dates-result
$(s_2, \text{res0-in9003})$: dates-result
$(s_1, \text{res0-ma0902})$: dates-result
$(s_2, \text{res1-in0008})$: dates-result
$(s_2, \text{res1-in0009})$: dates-result

Now we can relate these results to their corresponding exams.

(in0001, res0-in0001) : has-result
(in0003, res0-in0003) : has-result
(in0004, res0-in0004) : has-result
(in0006, res0-in0006) : has-result
(in0007, res0-in0007) : has-result
(in0008, res0-in0008) : has-result
(in0008, res1-in0008) : has-result
(in0009, res0-in0009) : has-result
(in0009, res1-in0009) : has-result
(in0015, res0-in0015) : has-result
(in9003, res0-in9003) : has-result
(ma0901, res0-ma0901) : has-result
(ma0902, res0-ma0902) : has-result

Because of these results the reasoner would classify the exams to additional classes as follows.

in0001	:	PassedExam
in0003	:	PassedExam
in0004	:	PassedExam
in0006	:	PassedExam
in0007	:	PassedExam
in0008	:	PassedExam
in0008	:	FailedExam
in0009	:	PassedExam
in0009	:	FailedExam
in0015	:	PassedExam
in9003	:	PassedExam
ma0901	:	PassedExam
ma0902	:	PassedExam

For completeness matter, the information of the taken exams can be given as follows.

(in0001, "Foundations of Computer Science 1")	:	course
(in0001, 2)	:	ultimo
(in0001, 9)	:	credits
(in0003, "Foundations of Computer Science 2")	:	course
(in0003, 2)	:	ultimo
(in0003, 6)	:	credits
(in0004, "Technical Foundations of Computer Science")	:	course
(in0004, 2)	:	ultimo
(in0004, 9)	:	credits
(in0006, "Introduction to Software Engineering")	:	course
(in0006, 2)	:	ultimo
(in0006, 6)	:	credits
(in0007, "Foundations of Algorithms and Data Structures")	:	course
(in0007, 2)	:	ultimo
(in0007, 9)	:	credits
(in0008, "Foundations of Databases")	:	course
(in0008, 6)	:	ultimo
(in0008, 6)	:	credits
(in0009, "Operating Systems ans System Software")	:	course
(in0009, 6)	:	ultimo
(in0009, 6)	:	credits
(in0015, "Discrete Mathematics")	:	course
(in0015, 2)	:	ultimo
(in0015, 9)	:	credits
(in9003, "Conmputer Science Law")	:	course
(in9003, 9)	:	ultimo
(in9003, 3)	:	credits
(ma0901, "Linear Algebra")	:	course

(ma0901, 2)	:	ultimo
(ma0901, 9)	:	credits
(ma0902, "Analysis")	:	course
(ma0902, 2)	:	ultimo
(ma0902, 9)	:	credits

References

[1] BAADER FRANK, SATTLER ULRIKE, Description Logics with Aggregates and Concrete Domains, 1997, Aachen, http://www.cs.man.ac.uk/~sattler/ulis-ps.html#1997

[2] BRACHMAN RONALD J., LEVESQUE HECTOR J., Knowledge Representation and Reasoning, Morgan Kaufmann Publishers, 2004, pp. 329-332

[3] GRIMM STEPHAN, MOTIK BORIS, Closed World Reasoning in the Semantic Web Through Epistemic Operators, Proceedings of the Workshop OWL - Experiences and Directions, 2005, Galway, Ireland, http://www.fzi.de/ipe/publikationen.php?id=1515

[4] HORROCKS IAN et al., SWRL: A Semantic Web Rule Language Combining OWL and RuleML, 2004, W3C Member Submission, http://www.w3.org/Submission/SWRL/

[5] HORROCKS IAN, SATTLER ULRIKE, A Tableaux Decision Procedure for $\mathcal{SHOIQ}$, 2005, Manchester, http://www.cs.man.ac.uk/~sattler/ulis-ps.html#2005

[6] PAN JEFF Z, HORROCKS IAN, Owl-Eu: Adding Customised Datatypes into OWL, 2005, http://www.cs.man.ac.uk/~horrocks/Publications/#2005

[7] PATEL-SCHNEIDER PETER F. et al., OWL Web Ontology Language Semantics and Abstract Syntax, W3C Recommendation, 2004, http://www.w3.org/TR/owl-semantics/

[8] SATTLER ULRIKE, Description Logic Reasoners, http://www.cs.man.ac.uk/~sattler/reasoners.html

[9] SCHREIBER GUUS, Qualified cardinality restrictions (QCRs): constraining the number of property values of a particular type, first draft, 2004, www.cs.vu.nl/~guus/public/qcr.html

[10] SMITH MICHAEL K. et al., OWL Web Ontology Language Guide, W3C Recommendation, 2004, deutsche Übersetzung, http://www.semaweb.org/dokumente/w3/TR/2004/REC-owl-guide-20040210-DE.html

[11] TOBIES STEPHAN, Complexity Results and Algorithms for Logics in Knowledge Representation, PhD Thesis, 2001, Essen, http://lat.inf.tu-dresden.de/research/phd/#Tobies-PhD-2001

[12] WOLSTENCROFT K. et al., A Little Semantic Web Goes a Long Way in Biology, in Gil Y. et al (Eds.), The Semantic Web - ISWC 2005, LNCS 3729, 2005, Springer, pp. 786-800

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Stand: 11. Oktober 2006

E N T W U R F

Prüfungs- und Studienordnung der Ludwig-Maximilians-Universität München für den Bachelorstudiengang

Vom

[auf der Grundlage der Muster-Prüfungs- und Studienordnung der Ludwig-Maximilians-Universität München für Bachelorstudiengänge]

Auf Grund von Art. 13 Abs. 1 Satz 2 in Verbindung mit Art. 58 Abs. 1 Satz 1 und 61 Abs. 2 Satz 1 des Bayerischen Hochschulgesetzes (BayHSchG) erlässt die Ludwig-Maximilians-Universität München folgende Satzung:

Inhaltsübersicht

I. Allgemeines

§ 1 Gegenstand des Studiengangs und Zweck der Bachelorprüfung
§ 2 Akademischer Grad
§ 3 Qualifikationsvoraussetzungen
§ 4 Zentrale Studienberatung und Fachstudienberatung

II. Dauer, Struktur und Ablauf des Studiums

§ 5 Studienbeginn, Regelstudienzeit, Semesterwochenstunden
§ 6 ECTS-Punkte # *[Option: , Nebenfach]*
§ 7 Modularisierung und Module
§ 8 Lehrveranstaltungen

III. Bachelorprüfung

1. Modulprüfungen, Modulteilprüfungen und Vorleistungen

§ 9 Modulprüfungen, Modulteilprüfungen und Vorleistungen als Bestandteile der Bachelorprüfung
§ 10 Bewertung der Modulprüfungen, Modulteilprüfungen und Vorleistungen
§ 11 Bestehen, Nichtbestehen und Wiederholung der Modulprüfungen, Modulteilprüfungen und Vorleistungen
§ 12 Kontoauszüge

2. Besondere Modulprüfungen, Modulteilprüfungen und Vorleistungen

§ 13 Grundlagen- und Orientierungsprüfung
§ 14 Bachelorarbeit
# *[Option: § 15 Abschlussprüfung]*

3. Prüfungsformen

§ 16 Mündliche Modulprüfungen, Modulteilprüfungen und Vorleistungen
§ 17 Klausuren und sonstige schriftliche Aufsichtsarbeiten
§ 18 Weitere Formen von Modulprüfungen, Modulteilprüfungen und Vorleistungen

4. Resultat der Bachelorprüfung

§ 19 Bestehen und Nichtbestehen der Bachelorprüfung
§ 20 Bescheid und Bescheinigung bei Nichtbestehen
§ 21 Bildung der Endnote
§ 22 Bachelor-Urkunde, Bachelor-Zeugnis, Transcript of Records und Diploma Supplement

IV. Prüfungsorgane und Prüfungsverwaltung

§ 23 Prüfungsausschuss und Prüfungsamt
§ 24 Prüfende und Beisitzende
§ 25 Studiengangskoordinatorin oder Studiengangskoordinator, Pflichten der Prüfenden
§ 26 Mitwirkungspflichten der Studierenden, Bestätigung von Mitteilungen

V. Durchführung der Prüfungen

§ 27 Anerkennung von Studien- und Prüfungsleistungen
§ 28 Belegung von Lehrveranstaltungen und Anmeldung zu Modulprüfungen, Modulteilprüfungen und Vorleistungen
§ 29 Versäumnis, Rücktritt
§ 30 Täuschung, Ordnungsverstoß, fehlende Teilnahmevoraussetzungen
§ 31 Schutzfristen nach dem Mutterschutzgesetz, Erziehungsurlaub
§ 32 Prüfungsvergünstigungen für Schwerbehinderte
§ 33 Mängel im Prüfungsverfahren
§ 34 Einsicht in die Prüfungsakten, Aufbewahrungsfristen

VI. Schlussbestimmungen

§ 35 Inkrafttreten und Übergangsvorschriften

Anlage 1: Beschreibung der Module und Lehrveranstaltungen

Anlage 2: Module, Lehrveranstaltungen, Modulprüfungen/ Modulteilprüfungen/ Vorleistungen

I. Allgemeines

§ 1
Gegenstand des Studiengangs und Zweck der Bachelorprüfung

(1) # [Studiengangsprofil]

(2) [1]Die studienbegleitend abzulegende Bachelorprüfung (§ 9 Abs. 1) bildet den berufsqualifizierenden Abschluss des Bachelorstudiengangs #. [2]Durch die Bachelorprüfung wird festgestellt, ob die oder der Studierende die Zusammenhänge des Faches überblickt und kritisch beurteilen kann, die Fähigkeit besitzt, dessen wissenschaftliche Methoden und Erkenntnisse anzuwenden und die für den Übergang in die Berufspraxis notwendigen gründlichen Fachkenntnisse erworben hat.

(3) [1]Im Rahmen der Lehrveranstaltungen dieses Bachelorstudiengangs werden auch Schlüsselqualifikationen vermittelt. [2]Schlüsselqualifikationen sind insbesondere

1. Fähigkeit, Wissen und Informationen zu recherchieren, zu bewerten, zu verdichten und zu strukturieren,
2. Überblickswissen zu maßgeblichen Wissensbereichen des jeweiligen Fachs,
3. vernetztes Denken,
4. Organisations- und Transferfähigkeit,
5. Informations- und Medienkompetenz,
6. Lern- und Präsentationstechniken,
7. Team- und Kommunikationsfähigkeit, auch unter genderspezifischen Gesichtspunkten,
8. Sprachkenntnisse sowie
9. EDV-Kenntnisse und Fähigkeiten.

§ 2
Akademischer Grad

Die Fakultät # verleiht denjenigen, die diesen Bachelorstudiengang erfolgreich abgeschlossen haben, den akademischen Grad # *[**Option:** „Bachelor of Arts“ (abgekürzt: „B.A.“) **oder** „Bachelor of Science“ (abgekürzt: „B.Sc.“) **oder** „Baccalaureus Artium“ (abgekürzt „B.A.“) **oder** „Baccalaureus Scientiae “ (abgekürzt „B.Sc.“)]*.

§ 3
Qualifikationsvoraussetzungen

(1) [1]Voraussetzung für die Aufnahme in diesen Bachelorstudiengang ist der Nachweis der Hochschulreife # *[**Option:** ; weitere Voraussetzungen werden ggf. in der Satzung über die Festsetzung von Zulassungszahlen für die im jeweiligen Studienjahr an der Ludwig-Maximilians-Universität München als Studienanfängerinnen und Studienanfänger sowie in höhere Fachsemester*

*aufzunehmenden Bewerberinnen und Bewerber (Zulassungszahlsatzung) in der jeweils geltenden Fassung **oder** in der Satzung über die Eignungsfeststellung für den Bachelorstudiengang # an der Ludwig-Maximilians-Universität München in der jeweils geltenden Fassung festgelegt].[2]Derselbe oder ein inhaltlich vergleichbarer Studiengang darf nicht endgültig nicht bestanden sein (Art. 46 Nr. 3 BayHSchG).]*

(2) Liegen die Voraussetzungen des Abs. 1 nicht vor, gilt eine Teilnahme an Modulprüfungen, Modulteilprüfung und Vorleistungen als nicht erfolgt, es sei denn ein späterer Nachweis der Voraussetzungen des Abs. 1 wurde ausdrücklich zugelassen und erfolgt fristgemäß.

§ 4
Zentrale Studienberatung und Fachstudienberatung

(1) [1]Die Zentrale Studienberatung an der Ludwig-Maximilians-Universität München erteilt Auskünfte und Ratschläge insbesondere bei fachübergreifenden Problemen. [2]Sie soll von den Studierenden insbesondere vor dem Studienbeginn, bei einem geplanten Wechsel des Studiengangs sowie bei allen Fragen in Bezug auf Zulassungsbeschränkungen in Anspruch genommen werden.

(2) [1]Die Fachstudienberatung wird in der Verantwortung der Fakultät von der zuständigen Fachstudienberaterin oder vom zuständigen Fachstudienberater durchgeführt. [2]Die Beratung erstreckt sich insbesondere auf Fragen der inhaltlichen und zeitlichen Studienplanung. [3]Auskünfte zu Fragen, die Prüfungen oder Anerkennungen von Studien- und Prüfungsleistungen betreffen, erteilen insbesondere die Mitglieder des Prüfungsausschusses und bzw. oder das Prüfungsamt.

II. Dauer, Struktur und Ablauf des Studiums

§ 5
Studienbeginn, Regelstudienzeit, Semesterwochenstunden

(1) Das Studium in diesem Bachelorstudiengang kann # *[**Option:** nur im Wintersemester **oder** im Wintersemester und im Sommersemester]* aufgenommen werden.

(2) [1]Die Regelstudienzeit beträgt einschließlich der Zeit für die Anfertigung der Bachelorarbeit sechs Semester. [2]Insgesamt sind höchstens # Semesterwochenstunden (SWS) erforderlich.

§ 6
ECTS-Punkte # *[Option: , Nebenfach]*

(1) [1]Im Rahmen dieses Bachelorstudiengangs sind insgesamt 180 Punkte nach dem European Credit Transfer System (ECTS-Punkte) zu erwerben # *[**Option:**, und zwar*
1. # *[**Option:** 165, 150, 120]* ECTS-Punkte im Hauptfach und
2. # *[**Option:** 15, 30, 60]* ECTS-Punkte in einem Nebenfach gemäß Abs. 3].

[2]ECTS-Punkte sind ein quantitatives Maß für die Gesamtarbeitsbelastung der oder des Studierenden. [3]Sie umfassen sowohl den unmittelbaren Unterricht in allen in § 8 Abs. 1 Satz 2 angegebenen Lehrveranstaltungen und Unterrichtsformen als auch die Zeit für die Vor- und Nachbereitung des Lehrstoffs (Präsenz- und Selbststudium), den Aufwand für die Prüfungsvorbereitungen und die erbrachten Prüfungsleistungen. [4]Ein ECTS-Punkt entspricht einer Arbeitsbelastung von ca. 30 Stunden, so dass die Gesamtarbeitsbelastung innerhalb der Regelstudienzeit (§ 5 Abs. 2 Satz 1) pro Semester ca. 900 Stunden beträgt und ca. 30 ECTS-Punkten entspricht.

(2) [1]In jedem Semester soll die oder der Studierende die sich aus Anlage 2/Spalte 18 ergebenden ECTS-Punkte erwerben. [2]ECTS-Punkte werden nur für bestandene Modulprüfungen, Modulteilprüfungen und Vorleistungen (§ 9) vergeben.

*# [**Option:** (3) Folgende Fächer sind als Nebenfach wählbar und auf der Grundlage der jeweils angegebenen Prüfungs- und Studienordnung für das Studium des Fachs im Nebenfach (Nebenfachsatzung) zu studieren:*

Nebenfach	***Nebenfachsatzung in der jeweils geltenden Fassung***

]

§ 7
Modularisierung und Module

(1) [1]Das Studium in diesem Bachelorstudiengang ist modular aufgebaut und in verbindlicher Weise in den Anlagen 1 und 2 geregelt. [2]Leeren Zellen der Tabellen in den Anlagen kommt kein Regelungsgehalt zu.

(2) [1]Das Studium in diesem Bachelorstudiengang umfasst Pflicht- und Wahlpflichtmodule. [2]Pflichtmodule sind ausnahmslos zu absolvieren; aus Wahlpflichtmodulen kann die oder der Studierende auswählen. # *[**Option:** [3]Es dürfen nicht mehr als die erforderliche Anzahl an Wahlpflichtmodulen gewählt werden.]* *# [**Option** 3 oder 4]*[4]Ein Wahlpflichtmodul wird spätestens durch Antreten einer dazugehörigen Modulprüfung, Modulteilprüfung oder Vorleistung gewählt; die Wahl

ist unwiderruflich. # *[**Option** 4 oder 5]*Ein Rechtsanspruch, dass jedes Wahlpflichtmodul in jedem Semester angeboten wird, besteht nicht.

(3) Ein Modul bezeichnet einen Verbund von thematisch und zeitlich aufeinander abgestimmten Lehrveranstaltungen sowie einer Modulprüfung oder einer oder mehreren Modulteilprüfungen, die entsprechend dem für eine erfolgreiche Teilnahme erforderlichen Zeitaufwand mit einer gemäß § 6 Abs. 1 bestimmten Anzahl an ECTS-Punkten bemessen werden.

(4) [1]Ein Modul erstreckt sich nach Maßgabe der Anlage 2 in der Regel über ein, höchstens über zwei Semester. [2]Der Umfang eines Moduls beträgt nach Maßgabe der Anlage 1/Spalte IV bzw. Anlage 2/Spalte 18 jeweils ein Vielfaches von drei ECTS-Punkten.

(5) Aus den Anlagen 1 und 2 ergeben sich

1. die Module,
2. deren Zuordnung zu einem oder mehreren Fachsemestern (Anlage 2/Spalte 1),
3. deren Zulassungsvoraussetzungen (Anlage 2/Spalte 2),
4. die Art der Module (Pflicht- oder Wahlpflichtmodul – Anlage 1/Spalte I und Anlage 2/Spalte 3), bei Wahlpflichtmodulen zusätzlich die Angabe, wie viele Wahlpflichtmodule aus welchen Wahlpflichtmodulen auszuwählen sind,
5. die Kurzbezeichnungen der Module (Anlage 2/Spalte 4),
6. die Bezeichnungen der Module in Deutsch (Anlage 1/Spalte I und Anlage 2/Spalte 5) und Englisch (Anlage 1/Spalte I),
7. die Beschreibungen (Inhalt und Lernziele) der Module in Deutsch und Englisch (Anlage 1/Spalte II),
8. der Angebotsturnus (semesterweise oder jährlich) der Module (Anlage 2/Spalte 6),
9. die dem Modul zugewiesenen ECTS-Punkte (Anlage 2/Spalte 18).

§ 8
Lehrveranstaltungen

(1) [1]Die Ziele und Inhalte des Studiums sowie Schlüsselqualifikationen (§ 1 Abs. 3) werden in den in der Anlage 1/Spalten II und III vorgesehenen Lehrveranstaltungen und Unterrichtsformen vermittelt. [2]In der Anlage 1/Spalte III bzw. in der Anlage 2/Spalte 9 können insbesondere folgende Lehrveranstaltungen und Unterrichtsformen vorgeschrieben werden:

1. Vorlesungen,
2. Übungen,
3. Proseminare,
4. Seminare,
5. Kolloquien,
6. Pflicht-Praktika
7. Exkursionen,
8. #,

9. #.

[3]Lehrveranstaltungen, in denen auch oder ausschließlich Schlüsselqualifikationen vermittelt werden, sind in der Anlage 1/Spalte II entsprechend gekennzeichnet.

(2) [1]Alle Lehrveranstaltungen sind Modulen zugeordnet. [2]Dieselbe Lehrveranstaltung kann von der- oder demselben Studierenden nicht mehrfach eingebracht werden.

(3) [1]Das Studium in diesem Bachelorstudiengang umfasst Pflicht- und Wahlpflichtlehrveranstaltungen. [2]Pflichtlehrveranstaltungen sind ausnahmslos zu absolvieren; aus Wahlpflichtlehrveranstaltungen kann die oder der Studierende auswählen. # *[**Option:** [3]Es dürfen nicht mehr als die erforderliche Anzahl an Wahlpflichtlehrveranstaltungen gewählt werden. **und/oder** [4]Eine Wahlpflichtlehrveranstaltung wird spätestens durch Antreten einer dazugehörigen Modulprüfung, Modulteilprüfung oder Vorleistung gewählt; die Wahl ist unwiderruflich.]* # *[**Option** 3, 4 oder 5* Wahlpflichtlehrveranstaltungen werden nach den Anlagen 1 und 2 ausschließlich Wahlpflichtmodulen zugeordnet.*# [**Option** 4, 5 oder 6]*Ein Rechtsanspruch, dass jede Wahlpflichtlehrveranstaltung in jedem Semester angeboten wird, besteht nicht.

(4) Aus den Anlagen 1 und 2 ergeben sich

1. die Lehrveranstaltungen,
2. die Art der Lehrveranstaltungen (Pflicht- oder Wahlpflichtlehrveranstaltung – Anlage 1/Spalte I und Anlage 2/Spalte 3), bei Wahlpflichtlehrveranstaltungen zusätzlich die Angabe, wie viele Wahlpflichtlehrveranstaltungen aus welchen Wahlpflichtlehrveranstaltungen auszuwählen sind,
3. deren Zuordnung zu einem oder mehreren Modulen,
4. deren Zuordnung zu einem oder mehreren Fachsemestern (Anlage 2/Spalte 1),
5. deren Zulassungsvoraussetzungen (Anlage 2/Spalte 7),
6. die Kurzbezeichnung der Lehrveranstaltungen (Anlage 2/Spalte 4),
7. die Bezeichnungen der Lehrveranstaltungen in Deutsch (Anlage 1/Spalte I und Anlage 2/Spalte 8) und in Englisch (Anlage 1/Spalte I),
8. die Beschreibungen (Inhalt und Lernziele) der Lehrveranstaltungen in Deutsch (Anlage 1/Spalte II) und Englisch (Anlage 1/Spalte II),
9. die Unterrichtsformen der Lehrveranstaltungen (Anlage 1/Spalte III und Anlage 2/Spalte 9),
10. die Semesterwochenstunden (Anlage 2/Spalte 10).

III. Bachelorprüfung

1. Modulprüfungen, Modulteilprüfungen und Vorleistungen

§ 9
Modulprüfungen, Modulteilprüfungen und Vorleistungen als Bestandteile der Bachelorprüfung

(1) Die Bachelorprüfung besteht aus Modulprüfungen, Modulteilprüfungen sowie Vorleistungen zu den beiden vorgenannten.

(2) [1]Jedes Modul schließt nach Maßgabe der Anlage 2 mit # *[**Option:** einer Modulprüfung **oder** einer bestimmten Anzahl an Modulteilprüfungen **oder** mit einer Modulprüfung oder einer bestimmten Anzahl an Modulteilprüfungen]* ab. [2]Die dem Modul zugewiesenen ECTS-Punkte werden dem persönlichen Konto (§ 12) der oder des Studierenden gutgeschrieben, wenn entweder die Modulprüfung bestanden ist oder alle Modulteilprüfungen bestanden sind. [3]Wird eine Modulprüfung durch mehrere Veranstaltungsleiterinnen und Veranstaltungsleiter gestellt, ohne dass es sich um Modulteilprüfungen handelt, finden die Vorschriften für Modulteilprüfungen entsprechende Anwendung.

*# [(3) **Option:** Die Teilnahme an Modulprüfungen, Modulteilprüfungen und Vorleistungen hängt von der Erfüllung von Zulassungsvoraussetzungen ab; das Nähere ergibt sich aus Anlage 2/Spalte 11. **oder:** Die Teilnahme an Modulprüfungen und bzw. oder Modulteilprüfungen hängt von Vorleistungen ab; das Nähere ergibt sich aus Anlage 2/Spalte 11.]*

(4) [1]In der Modulprüfung, der Modulteilprüfung oder in der Summe der Modulteilprüfungen des jeweiligen Moduls soll die oder der Studierende nachweisen, dass sie oder er die Kenntnisse und Fähigkeiten erworben hat, welche in der oder den dem Modul nach Anlage 1/Spalten I und II und nach Anlage 2/Spalten 7 bis 10 zugeordneten Lehrveranstaltungen vermittelt werden. [2]In Vorleistungen soll die oder der Studierende nachweisen, dass sie oder er die Voraussetzungen erfüllt, um an der jeweiligen Modulprüfung oder Modulteilprüfung teilzunehmen.

(5) Aus der Anlage 2 ergeben sich

1. die Modulprüfungen, Modulteilprüfungen und Vorleistungen,
2. deren Zuordnung zu einem Modul und ggf. einer Lehrveranstaltung,
3. deren Zuordnung zu einem Fachsemester (Regeltermin – Anlage 2/Spalte 1)
4. deren Zulassungsvoraussetzungen (Anlage 2/Spalte 11),
5. die Art der Modulprüfung, Modulteilprüfung oder Vorleistung (Anlage 2/Spalte 12),
6. die Prüfungsform (Anlage 2/Spalte 13),
7. die Prüfungsdauer (Anlage 2/Spalte 14),
8. die Art der Bewertung (Benotung bzw. „bestanden“ oder „nicht bestanden“ – Anlage 2/Spalte 15),
9. das Notengewicht (Anlage 2/Spalte 16),

10. die Wiederholbarkeit (Anlage 2/Spalte 17),
11. die ECTS-Punkte, die bei erfolgreichem Ablegen der Modulprüfungen, Modulteilprüfungen oder Vorleistungen vergeben werden (Anlage 2/Spalte 18).

§ 10
Bewertung der Modulprüfungen, Modulteilprüfungen und Vorleistungen

(1) Modulprüfungen und Modulteilprüfungen werden mit „bestanden“ oder „nicht bestanden“ bewertet oder benotet; Vorleistungen werden mit „bestanden“ oder „nicht bestanden“ bewertet.

(2) [1]Die Note für eine Modulprüfung oder für eine Modulteilprüfung wird von der oder dem jeweiligen Prüfenden festgesetzt. [2]Für die Bewertung der Modulprüfungen und Modulteilprüfungen sind folgende Noten zu verwenden:

Note 1	= „sehr gut“	=	hervorragende Leistung;
Note 2	= „gut“	=	Leistung, die erheblich über den Anforderungen liegt;
Note 3	= „befriedigend“	=	Leistung, die durchschnittlichen Anforderungen genügt;
Note 4	= „ausreichend“	=	Leistung, die trotz ihrer Mängel noch den Anforderungen genügt;
Note 5	= „nicht ausreichend“	=	Leistung, die wegen erheblicher Mängel den Anforderungen nicht mehr genügt.

[3]Zur differenzierten Bewertung der Modulprüfungen und Modulteilprüfungen können die Noten um 0,3 auf Zwischenwerte erhöht oder erniedrigt werden; die Noten 0,7; 4,3; 4,7 und 5,3 sind dabei ausgeschlossen. [4]Wird eine Modulprüfung oder Modulteilprüfung von mehreren Prüfenden benotet, errechnet sich die Gesamtnote der Modulprüfung oder Modulteilprüfung aus dem arithmetischen Mittel der Einzelbewertungen. [5]Dabei werden nur die ersten beiden Stellen hinter dem Komma berücksichtigt. [6]Die Notenbezeichnung nach Satz 4 lautet:

bei einem Durchschnitt bis einschließlich 1,50	=	„sehr gut“;
bei einem Durchschnitt von 1,51 bis einschließlich 2,50	=	„gut“;
bei einem Durchschnitt von 2,51 bis einschließlich 3,50	=	„befriedigend“;
bei einem Durchschnitt von 3,51 bis einschließlich 4,00	=	„ausreichend“.

(3) [1]Die Modulnote

1. ergibt sich bei einer Modulprüfung (§ 9 Abs. 2) aus Abs. 2 und
2. errechnet sich bei Modulteilprüfungen (§ 9 Abs. 2) aus dem arithmetischen Mittel der nach Anlage 2/Spalte 15 benoteten und nach Anlage 2/Spalte 16 gewichteten Einzelbewertungen in den zu dem jeweiligen Modul gehörenden Modulteilprüfungen.

[2]Soweit in Anlage 2/Spalte 16 keine andere Angabe erfolgt, gehen die Modulteilprüfungen mit den ihnen jeweils in Anlage 2/Spalte 18 zugeordneten ECTS-

Punkten in das nach Satz 1 Nr. 2 zu bildende arithmetische Mittel ein. [3]Abs. 2 Sätze 5 und 6 gelten entsprechend.

(4) [1]Werden innerhalb eines Moduls Modulprüfungen oder Modulteilprüfungen für mehr Lehrveranstaltungen erfolgreich absolviert als zum Bestehen des Moduls erforderlich sind, werden bei der Berechnung der Modulnote nur die für das Bestehen des Moduls erforderlichen ECTS-Punkte berücksichtigt. [2]Erforderlich für das Bestehen eines Moduls sind

1. alle den Pflichtlehrveranstaltungen zugeordneten Modulprüfungen oder Modulteilprüfungen in einer in den Anlagen 1 und 2 vorgesehenen Weise und
2. alle den erforderlichen Wahlpflichtlehrveranstaltungen zugeordneten Modulprüfungen oder Modulteilprüfungen in einer in den Anlagen 1 und 2 vorgesehenen Weise.

[3]Werden Modulprüfungen oder Modulteilprüfungen für mehr Wahlpflichtlehrveranstaltungen abgelegt, als nach Satz 2 Nr. 2 zu absolvieren sind, gilt vorbehaltlich des § 8 Abs. 3 die zeitlich zuerst erfolgreich abgelegte als erforderlich im Sinne des Satzes 2. [4]Es werden bei Wahlpflichtlehrveranstaltungen,

1. die in verschiedenen Semestern erfolgreich erbracht wurden, die früheren,
2. die im selben Semester erfolgreich erbracht wurden, die besseren

berücksichtigt.

§ 11
Bestehen, Nichtbestehen und Wiederholung der Modulprüfungen, Modulteilprüfungen und Vorleistungen

(1) [1]Eine Modulprüfung, Modulteilprüfung oder Vorleistung ist bestanden, wenn sie

1. mit „bestanden“ oder
2. mit mindestens „ausreichend“ (4,0)

bewertet ist. [2]Modulprüfungen, Modulteilprüfungen und Vorleistungen sollen vorbehaltlich des § 31 spätestens am Ende des in Anlage 2/Spalte 1 genannten Semesters bestanden sein (Regeltermin); Angaben in Klammern in Anlage 2/Spalte 1 sind nur Empfehlungen. [3]Modulprüfungen, Modulteilprüfungen und Vorleistungen sind bestanden, wenn vorbehaltlich des § 31 spätestens am Ende des auf den Ablauf des Regeltermins folgenden Fachsemesters alle erforderlichen Teilleistungen erbracht sind.

(2) [1]Enthält die Anlage 2/Spalte 1 für eine Modulprüfung, Modulteilprüfung oder Vorleistung keine oder eine Angabe in Klammern, gilt das Ende des sechsten Fachsemesters als Regeltermin. [2]Diese Modulprüfung, Modulteilprüfung oder Vorleistung ist bestanden, wenn sie vorbehaltlich des § 31 spätestens am Ende des siebten Fachsemesters erbracht ist.

(3) Modulprüfungen, Modulteilprüfungen und Vorleistungen sind auch bestanden, wenn die Voraussetzungen der Abs. 1 und 2 nicht im dort vorgesehenen Zeitraum erfüllt sind, aber im Rahmen einer nach dieser Prüfungs- und Studienordnung zulässigen Wiederholung erfüllt werden.

(4) [1]Modulprüfungen, Modulteilprüfungen oder Vorleistungen sind nicht bestanden, wenn sie ganz oder teilweise abgelegt, aber nicht bestanden wurden. [2]Modulprüfungen, Modulteilprüfungen oder Vorleistungen sind endgültig nicht bestanden, wenn sie ganz oder teilweise abgelegt, aber nicht bestanden wurden und keine Wiederholungsmöglichkeit mehr besteht.

(5) [1]Modulprüfungen, Modulteilprüfungen oder Vorleistungen gelten vorbehaltlich des § 31

1. als abgelegt und nicht bestanden, wenn sie am Ende des auf den Ablauf des Regeltermins folgenden Fachsemesters aus selbst zu vertretenden Gründen nicht erfolgreich abgelegt sind, und
2. als endgültig nicht bestanden, wenn sie aus selbst zu vertretenden Gründen am Ende des # *[**Option: – bei jährlichem Studienangebot –** dritten **oder – bei semesterweisem Studienangebot –** zweiten]* auf den Ablauf des Regeltermins folgenden Fachsemesters nicht erfolgreich abgelegt sind.

[2]Enthält die Anlage 2/Spalte 1 für eine Modulprüfung, Modulteilprüfung oder Vorleistung keine oder eine Angabe in Klammern, gilt diese Modulprüfung, Modulteilprüfung oder Vorleistung vorbehaltlich des § 31

1. als abgelegt und nicht bestanden, wenn sie am Ende des siebten Fachsemesters aus selbst zu vertretenden Gründen nicht erfolgreich abgelegt ist, und
2. als endgültig nicht bestanden, wenn sie aus selbst zu vertretenden Gründen am Ende des # *[**Option: – bei jährlichem Studienangebot –** neunten **oder – bei semesterweisem Studienangebot –** achten]* Fachsemesters nicht erfolgreich abgelegt ist.

[3]Gründe, die das Überschreiten einer der Fristen der Sätze 1 und bzw. oder 2 rechtfertigen sollen, müssen unverzüglich nach ihrem Auftreten beim Prüfungsamt schriftlich geltend und glaubhaft gemacht werden. [4]Bei Krankheit muss ein ärztliches Attest vorgelegt werden; die Vorlage einer Arbeitsunfähigkeitsbescheinigung genügt nicht. [5]Das Prüfungsamt kann im Einzelfall oder allgemein die Vorlage eines amtsärztlichen Attestes oder eines Attestes einer oder eines vom Prüfungsamt bestimmten Ärztin oder Arztes verlangen. [6]Wird der Grund anerkannt, so wird ein neuer Termin anberaumt. [7]Bei teilbaren Modulprüfungen, Modulteilprüfungen und Vorleistungen sind bereits vorliegende Prüfungsergebnisse anzurechnen. [8]Soweit Modulprüfungen, Modulteilprüfungen oder Vorleistungen in dem Semester erstmals nicht bestanden werden, an dessen Ende vorbehaltlich des § 31 die Frist nach Satz 1 Nr. 1, Satz 2 Nr. 1 oder die Frist nach Satz 1 Nr. 2, Satz 2 Nr. 2 abläuft, verlängert sich die jeweilige Frist soweit, dass im nächstmöglichen Termin eine Wiederholung möglich ist.

Variante a)

*(6) [1]Für jede Modulprüfung, Modulteilprüfung und Vorleistung, mit Ausnahme der # [**Option:** Grundlagen- und Orientierungsprüfung (§ 13) und der Bachelorarbeit (§ 14) **oder** Grundlagen- und Orientierungsprüfung (§ 13), der Bachelorarbeit (§ 14) und der Abschlussprüfung (§ 15)], werden insgesamt vier Versuche angeboten. [2]Der erste Versuch wird während oder unmittelbar im Anschluss an die Vorlesungszeit des letzten Semesters angeboten, in welchem die oder der Studierende an Lehrveranstaltungen teilnimmt, die Gegenstand der Modulprüfung, Modulteilprüfung oder Vorleistung sind. [3]Der zweite Versuch wird so rechtzeitig während des noch laufenden Semesters im Sinne des Satzes 2 angeboten, dass die Vorgaben des § 25 Abs. 2 erfüllt werden können. [4]Der dritte Versuch wird während oder unmittelbar im Anschluss an die Vorlesungszeit des letzten Semesters angeboten, in welchem die Lehrveranstaltungen, welche die oder der Studierende nicht bestanden hat, das nächste Mal angeboten werden. [5]Für den vierten Versuch gilt Satz 3 entsprechend.*

Variante b)

*(6) Eine nicht bestandene Modulprüfung, Modulteilprüfung oder Vorleistung, mit Ausnahme der # [**Option:** Grundlagen- und Orientierungsprüfung (§ 13) und der Bachelorarbeit (§ 14) **oder** Grundlagen- und Orientierungsprüfung (§ 13), der Bachelorarbeit (§ 14) und der Abschlussprüfung (§ 15)], kann, vorbehaltlich einer abweichenden Regelung in der Anlage 2/Spalte 17, beliebig oft wiederholt werden.*

Variante c)

(6) Jede nicht bestandene Modulprüfung, Modulteilprüfung oder Vorleistung kann, vorbehaltlich einer abweichenden Regelung in Anlage 2/Spalte 17, nur einmal im nächstmöglichen regulären Termin wiederholt werden.

Variante d)

(6) Jede nicht bestandene Modulprüfung, Modulteilprüfung oder Vorleistung kann, vorbehaltlich einer abweichenden Regelung in Anlage 2/Spalte 17, einmal in einem beliebigen regulären Termin wiederholt werden.

(7) Die Wiederholung einer bereits bestandenen Modulprüfung, Modulteilprüfung oder Vorleistung zur Notenverbesserung ist nicht möglich.

§ 12
Kontoauszüge

[1]Für die in diesen Bachelorstudiengang eingeschriebenen Studierenden wird beim Prüfungsamt ein persönliches Konto eingerichtet, in dem alle

1. bestandenen Modulprüfungen, Modulteilprüfungen und Vorleistungen (§ 11 Abs. 1 bis 3) jeweils mit dem Hinweis „bestanden“ bzw. mit der vergebenen Note und mit den erzielten ECTS-Punkten sowie

2. alle nicht bestandenen Modulprüfungen, Modulteilprüfungen und Vorleistungen (§ 11 Abs. 4 und 5) jeweils mit dem Hinweis „nicht bestanden“ bzw. mit der vergebenen Note erfasst werden.

[2]Zu Beginn des jeweils nächsten Semesters erhalten die Studierenden einen persönlichen Kontoauszug im Sinn von Satz 1 als Bescheid mit Rechtsbehelfsbelehrung.

2. Besondere Modulprüfungen, Modulteilprüfungen und Vorleistungen

§ 13
Grundlagen- und Orientierungsprüfung

(1) Die Grundlagen- und Orientierungsprüfung ist # *[**Option: falls in Abs. 3 Variante a) gewählt wird, je nach Inhalt der Anlage 2/Spalte 12 für das erste oder zweite Fachsemester** eine Modulprüfung **oder** eine Modulteilprüfung **oder** eine Vorleistung **oder falls in Abs. 3 Variante b) gewählt wird:** die in Anlage 2/Spalte 12 für das erste oder zweite Fachsemester vorgesehene # [Modulprüfung, Modulteilprüfung oder Vorleistung]]*.

(2) Die Grundlagen- und Orientierungsprüfung dient einer ersten und frühzeitigen Orientierung der oder des Studierenden darüber, ob sie oder er den Anforderungen dieses Bachelorstudiengangs voraussichtlich gerecht werden wird.

Variante a)

(3) Die Grundlagen- und Orientierungsprüfung ist bestanden, wenn mindestens eine der in der Anlage 2/Spalte 12 für das erste oder zweite Fachsemester vorgesehenen und als Grundlagen- und Orientierungsprüfung gekennzeichneten Modulprüfungen, Modulteilprüfungen oder Vorleistungen mit „bestanden“ bzw. „ausreichend“ (4,0) oder besser bewertet wurde.

Variante b)

*(3) Die Grundlagen- und Orientierungsprüfung ist bestanden, wenn die in der Anlage 2/Spalte 12 für das erste oder zweite Fachsemester vorgesehene und als Grundlagen- und Orientierungsprüfung gekennzeichnete # **[Option:** Modulprüfung, Modulteilprüfung oder Vorleistung] mit „bestanden“ bzw. „ausreichend“ (4,0) oder besser bewertet wurde.*

(4) [1]Die Grundlagen- und Orientierungsprüfung muss bis zum Ende des zweiten Fachsemesters bestanden sein. [2]Eine nicht bestandene Grundlagen- und Orientierungsprüfung kann einmal im nächstmöglichen Termin wiederholt werden. [3]Die Anordnung nach § 28 Abs. 2 Satz 3 gilt als erfolgt.

(5) [1]Die Grundlagen- und Orientierungsprüfung gilt vorbehaltlich des § 31

1. als abgelegt und nicht bestanden, wenn sie am Ende des zweiten Fachsemesters aus selbst zu vertretenden Gründen nicht erfolgreich abgelegt ist, und
2. als endgültig nicht bestanden, wenn sie aus selbst zu vertretenden Gründen im auf den nach Nr. 1 nächstmöglichen Termin nicht erfolgreich abgelegt wird.

[2]§ 11 Abs. 5 Sätze 3 bis 7 gelten entsprechend.

§ 14
Bachelorarbeit

(1) Die Bachelorarbeit ist eine # *[**Option:** Modulprüfung **oder** Modulteilprüfung]*.

(2) Die Bachelorarbeit soll zeigen, dass die oder der Studierende in der Lage ist, innerhalb einer vorgegebenen Frist (Abs. 7) ein Problem aus ihrem oder seinem Fach selbständig nach wissenschaftlichen Methoden zu bearbeiten.

(3) [1]Die Bachelorarbeit wird von einer nach § 24 Abs. 3 Nr. 3 zur ersten oder zum ersten Prüfenden bestellten Person betreut (Betreuerin oder Betreuer). [2]Soll die Bachelorarbeit in einer Einrichtung außerhalb der Ludwig-Maximilians-Universität München durchgeführt werden, bedarf es hierzu der Zustimmung der Vorsitzenden oder des Vorsitzenden des Prüfungsausschusses.

(4) [1]Das Verfahren der Themenvergabe wird # *[**Option:** und die Überprüfung der Zulassungsvoraussetzungen werden]* in den ersten beiden Wochen nach Semesterbeginn durch das Prüfungsamt ortsüblich bekannt gegeben; eine Bekanntgabe durch das Prüfungsamt ausschließlich im Internet ist ausreichend. [2]Thema und Zeitpunkt der Ausgabe der Bachelorarbeit werden beim Prüfungsamt aktenkundig gemacht. [3]Die oder der Studierende kann Themenwünsche äußern; die Betreuerin oder der Betreuer ist hieran nicht gebunden. [4]Das Thema kann nur einmal und nur innerhalb von zwei Wochen nach Ausgabe zurückgegeben werden. [5]Die Anordnung nach § 28 Abs. 2 Satz 3 gilt als erfolgt; § 28 Abs. 3 gilt entsprechend.

(5) [1]Die Betreuerin oder der Betreuer ist verpflichtet,

1. das Thema der Bachelorarbeit so rechtzeitig zu vergeben und
2. die Bachelorarbeit so rechtzeitig zu bewerten,

dass dem Prüfungsamt spätestens zwei Wochen vor Ende des laufenden Semesters die Bewertung vorliegt. [2]Für eine zweite Prüfende oder einen zweiten Prüfenden gilt Satz 1 Nr. 2 entsprechend.

(6) [1]Studierende, an die zu Beginn der Vorlesungszeit ihres letzten Fachsemesters noch kein Thema für eine Bachelorarbeit vergeben wurde, müssen sich unverzüglich bei der oder dem Vorsitzenden des Prüfungsausschusses melden. [2]Die oder der Vorsitzende des Prüfungsausschusses ist verpflichtet, für die Vergabe eines

Themas für eine Bachelorarbeit an jede Studierende oder jeden Studierenden Sorge zu tragen.

(7) [1]Die Bearbeitungsdauer der Bachelorarbeit beträgt # [*max. 10*] Wochen. [2]Für die Bachelorarbeit werden # [*mindestens 6 bis maximal 12*] ECTS-Punkte vergeben.

(8) [1]Die Bachelorarbeit ist fristgemäß in # Exemplaren beim Prüfungsamt abzuliefern; der Abgabezeitpunkt ist aktenkundig zu machen. [2]Bei der Abgabe hat der Prüfling schriftlich zu versichern, dass er seine Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt hat.

(9) [1]Die Bachelorarbeit ist durch die Betreuerin oder den Betreuer der Bachelorarbeit (Abs. 3 Satz 1) zu bewerten. [2]Bachelorarbeiten, die als „nicht bestanden“ bewertet werden sollen, sind durch eine weitere Prüfende oder einen weiteren Prüfenden (§ 24 Abs. 3 Nr. 3) zu bewerten.

(10) [1]Die Bachelorarbeit kann bei einer Bewertung, die schlechter als „ausreichend“ (4,0) ist, einmal wiederholt werden. [2]Eine Rückgabe des Themas der Bachelorarbeit in der in Abs. 4 Satz 4 genannten Frist ist jedoch nur zulässig, wenn die oder der Studierende bei der Anfertigung ihrer oder seiner ersten Arbeit von dieser Möglichkeit keinen Gebrauch gemacht hat.

 # [Option: § 15 Abschlussprüfung

*[1]Die Abschlussprüfung ist # [**Option:** eine Modulprüfung, welche in der Anlage 2/Spalte 12 als solche gekennzeichnet ist **oder** eine Modulteilprüfung, welche in der Anlage 2/Spalte 12 als solche gekennzeichnet ist **oder** besteht aus mehreren Modulteilprüfungen, welche in der Anlage 2/Spalte 12 als solche gekennzeichnet sind].*

Variante a)

[2]Prüfungsgegenstand der Abschlussprüfung ist die Bachelorarbeit.

Variante b)

*[2]Prüfungsgegenstand der Abschlussprüfung ist der Inhalt des Moduls mit der Kurzbezeichnung # **oder** der Module mit den Kurzbezeichnungen #.*

[3]Eine nicht bestandene Abschlussprüfung kann einmal im nächstmöglichen Termin wiederholt werden. [4]Die Anordnung nach § 28 Abs. 2 Satz 3 gilt als erfolgt. [5]Für die Abschlussprüfung werden insgesamt # [max. die Anzahl der ECTS-Punkte, welche der Bachelorarbeit zugeordnet sind] ECTS-Punkte vorgesehen.]

3. Prüfungsformen

§ 16
Mündliche Modulprüfungen, Modulteilprüfungen und Vorleistungen

(1) [1]Durch mündliche Modulprüfungen, Modulteilprüfungen und Vorleistungen soll die oder der Studierende nachweisen, dass sie oder er die Zusammenhänge des Prüfungsgebietes erkennt und spezielle Fragestellungen in diese Zusammenhänge einzuordnen vermag. [2]Ferner soll festgestellt werden, ob die oder der Studierende über ein dem Stand des Bachelorstudiums entsprechendes Grundlagenwissen verfügt.

(2) [1]Die Dauer einer mündlichen Modulprüfung, Modulteilprüfung oder Vorleistung beträgt für jeden Prüfling mindestens # und höchstens # Minuten. [2]Das Nähere wird in der Anlage 2/Spalte 14 geregelt.

(3) [1]Die wesentlichen Gegenstände und Ergebnisse der mündlichen Modulprüfungen, Modulteilprüfungen und Vorleistungen sind in einem Protokoll festzuhalten. [2]Das Ergebnis ist der oder dem Studierenden im Anschluss an die mündliche Modulprüfung, Modulteilprüfung oder Vorleistung bekannt zu geben.

§ 17
Klausuren und sonstige schriftliche Aufsichtsarbeiten

(1) [1]In den Klausuren und sonstigen schriftlichen Aufsichtsarbeiten soll die oder der Studierende nachweisen, dass sie oder er auf der Basis des notwendigen Grundlagenwissens in begrenzter Zeit und mit begrenzten Hilfsmitteln mit den gängigen Methoden ihres oder seines Faches Aufgaben lösen und Themen bearbeiten kann. [2]Der oder dem Studierenden können Themen zur Auswahl gegeben werden; ein Anspruch hierauf besteht nicht.

(2) [1]Die Dauer der Klausuren und sonstigen schriftlichen Aufsichtsarbeiten beträgt mindestens # und höchstens # Minuten. [2]Das Nähere wird in Anlage 2/Spalte 14 geregelt.

(3) [1]Schriftliche Modulprüfungen, Modulteilprüfungen und Vorleistungen können ganz oder teilweise auch in der Weise abgenommen werden, dass der Prüfling anzugeben hat, welche der mit den Aufgaben vorgelegten Antworten er für richtig hält (Antwort-Wahl-Verfahren). [2]Die Prüfungsaufgaben müssen zuverlässige Prüfungsergebnisse ermöglichen. [3]Dabei sind jeweils allen Prüflingen dieselben Prüfungsaufgaben zu stellen. [4]Bei der Aufstellung der Prüfungsaufgaben ist festzulegen, welche Antworten als zutreffend anerkannt werden. [5]Die Prüfungsaufgaben sind durch die Aufgabenstellerinnen oder die Aufgabensteller vor der Feststellung des Prüfungsergebnisses darauf zu überprüfen, ob sie gemessen an den Anforderungen des Satzes 2 fehlerhaft sind. [6]Ergibt diese Überprüfung, dass einzelne Prüfungsaufgaben fehlerhaft sind, sind diese bei der Feststellung des Prüfungsergebnisses nicht zu berücksichtigen. [7]Die Zahl der Aufgaben für die einzelnen Modulprüfungen, Modulteilprüfungen und Vorleistungen mindert sich

entsprechend. [8]Bei der Bewertung der schriftlichen Modulprüfung, Modulteilprüfung oder Vorleistung nach Satz 10 ist von der verminderten Zahl der Prüfungsaufgaben auszugehen. [9]Die Verminderung der Zahl der Prüfungsaufgaben darf sich nicht zum Nachteil eines Prüflings auswirken. [10]Schriftliche Modulprüfungen, Modulteilprüfungen und Vorleistungen nach Satz 1 gelten als bestanden, wenn

1. der Prüfling insgesamt mindestens 60 Prozent der gestellten Prüfungsfragen zutreffend beantwortet hat oder
2. der Prüfling insgesamt mindestens 50 Prozent der gestellten Prüfungsfragen zutreffend beantwortet hat und die Zahl der vom Prüfling zutreffend beantworteten Fragen um nicht mehr als 15 Prozent die durchschnittlichen Prüfungsleistungen der Prüflinge unterschreitet, die erstmals an der entsprechenden Prüfung teilgenommen haben.

[11]Wird Satz 10 Nr. 2 angewendet, ist die Studiendekanin oder der Studiendekan zu unterrichten. [12]Bei schriftlichen Modulprüfungen, Modulteilprüfungen oder Vorleistungen, die nur teilweise im Antwort-Wahl-Verfahren abgenommen werden, gelten die Sätze 2 bis 11 nur für diesen Teil.

(4) [1]Eine schriftliche Modulprüfung, Modulteilprüfung oder Vorleistung kann auch in elektronischer Form abgenommen werden. [2]Art und Umfang der elektronischen Leistungserhebung werden zu Beginn der Lehrveranstaltung von der Veranstaltungsleiterin oder dem Veranstaltungsleiter bekannt gegeben. [3]Den Studierenden wird vor der Prüfung im Rahmen der Lehrveranstaltung ausreichend Gelegenheit gegeben, sich mit dem elektronischen Prüfungssystem vertraut zu machen. [4]Die datenschutzrechtlichen Bestimmungen sind einzuhalten.

§ 18
Weitere Formen von Modulprüfungen, Modulteilprüfungen und Vorleistungen

(1) [1]Eine Hausarbeit ist in schriftlicher Form als fortlaufender Text im Umfang von ca. # Zeichen zu erbringen. [2]Die Bearbeitungsdauer soll # Wochen nicht überschreiten.

(2) [1]Ein Referat ist ein eigenständig vorbereiteter Vortrag, der durch geeignete visuelle Hilfsmittel unterstützt werden soll. [2]Die Dauer des Vortrags soll zwischen # und # Minuten betragen. [3]An das Referat kann sich ein Fachgespräch anschließen.

(3) [1]Wissenschaftliche Protokolle beinhalten die schriftliche, systematische Aufarbeitung einer fachlich geeigneten Veranstaltung einschließlich einer kritischen Diskussion der Inhalte. [2]Die Dauer der zu Grunde liegenden Veranstaltung soll # [für das jeweilige Fach üblicher Zeitraum] nicht überschreiten.

(4) [1]Die Durchführung von Fallstudien basiert auf praxisbezogenen Problemstellungen. [2]Mit der Fallstudie soll der Nachweis erbracht werden, in fundierter Weise Theorien, Modelle und Konzepte anwenden zu können. [3]Zur Bewertung gelangt die Darstellung der Ergebnisse der Fallstudie.

(5) Das Lösen von Übungsaufgaben erfolgt in einem regelmäßigen Turnus über die Dauer des Semesters.

(6) Auf einem Poster sollen wissenschaftliche Sachverhalte mittels Text und mit Hilfe von Illustrationen dargestellt werden.

(7) Ein Exkursionsbericht ist eine schriftliche Zusammenfassung eines Exkursionstages im Umfang von ca. # Zeichen.

*[**Option**: (8) #]*

4. Resultat der Bachelorprüfung

§ 19
Bestehen und Nichtbestehen der Bachelorprüfung

(1) Die Bachelorprüfung soll bis zum Abschluss des sechsten Fachsemesters bestanden sein.

(2) [1]Die Bachelorprüfung ist bestanden, wenn spätestens bis zum Abschluss des siebten Fachsemesters

1. die Grundlagen- und Orientierungsprüfung bestanden ist,
2. alle Modulprüfungen oder Modulteilprüfungen der Pflichtmodule und der erforderlichen Wahlpflichtmodule in einer in den Anlagen 1 und 2 vorgesehenen Weise bestanden sind und
3. die erforderliche Anzahl an 180 ECTS-Punkten erbracht ist.

[2]Die Bachelorprüfung ist auch bestanden, wenn die Voraussetzungen des Satzes 1 nicht im dort vorgesehenen Zeitraum erfüllt sind, aber im Rahmen einer nach dieser Prüfungs- und Studienordnung zulässigen Wiederholung erfüllt werden.

(3) Die Bachelorprüfung ist endgültig nicht bestanden, wenn

1. die Grundlagen- und Orientierungsprüfung oder
2. die Modulprüfung oder eine Modulteilprüfung eines der in den Anlagen 1 und 2 vorgesehenen Pflichtmodule oder erforderlichen Wahlpflichtmodule

abgelegt, aber nicht bestanden wurde und keine Wiederholungsmöglichkeit mehr besteht.

(4) [1]Die Bachelorprüfung gilt vorbehaltlich des § 31

1. als erstmals abgelegt und nicht bestanden, wenn die in Abs. 1 genannte Frist aus selbst zu vertretenden Gründen um mehr als ein Semester überschritten wird, und
2. als endgültig nicht bestanden, wenn die in Abs. 1 genannte Frist aus selbst zu vertretenden Gründen um mehr als # *[**Option: – bei jährlichem Studienangebot –** drei **oder – bei semesterweisem Studienangebot –** zwei]* Semester überschritten wird.

[2]§ 11 Abs. 5 Sätze 3 bis 7 gelten entsprechend.

§ 20
Bescheid und Bescheinigung bei Nichtbestehen

(1) Wenn die Bachelorprüfung

1. gemäß § 19 Abs. 3 endgültig nicht bestanden wurde oder
2. gemäß § 19 Abs. 4 Satz 1 Nr. 2 als endgültig nicht bestanden gilt,

erlässt die oder der Vorsitzende des Prüfungsausschusses hierüber einen schriftlichen und mit einer Rechtsbehelfsbelehrung versehenen Bescheid; § 23 Abs. 5 gilt entsprechend.

(2) Wurde die Bachelorprüfung nicht bestanden oder gilt sie als nicht bestanden, wird auf Antrag und gegen Vorlage der Exmatrikulationsbescheinigung eine Bescheinigung ausgestellt, welche die erfolgreich erbrachten Modulprüfungen, Modulteilprüfungen und Vorleistungen, die zugeordneten ECTS-Punkte und Noten, sowie eine Erklärung enthält, dass die Bachelorprüfung nicht bestanden ist.

§ 21
Bildung der Endnote

[1]Ist die Bachelorprüfung nach § 19 Abs. 2 bestanden, errechnet sich die Endnote aus dem arithmetischen Mittel der nach Anlage 2/Spalte 16 gewichteten Modulnoten; § 10 Abs. 3 Sätze 2 und 3 gelten für die Berechnung der Endnote aus den Modulnoten entsprechend. [2]Werden in der Bachelorprüfung mehr als 180 ECTS-Punkte erworben, werden nur die für das Bestehen der Bachelorprüfung erforderlichen 180 ECTS-Punkte berücksichtigt. [3]Erforderlich für das Bestehen der Bachelorprüfung sind

1. alle Vorleistungen, Modulprüfungen und Modulteilprüfungen der Pflichtmodule in einer in den Anlagen 1 und 2 vorgesehenen Weise und
2. alle Vorleistungen, Modulprüfungen und Modulteilprüfungen der Wahlpflichtmodule in einer in den Anlagen 1 und 2 vorgesehenen Weise.

[4]Werden mehr Wahlpflichtmodule abgelegt, als nach Satz 3 Nr. 2 zu absolvieren sind, gilt vorbehaltlich des § 7 Abs. 2 Sätze 3 und 4 das zeitlich zuerst erfolgreich abgelegte als erforderlich im Sinne des Satzes 2. [5]Es werden bei Wahlpflichtmodulen,

1. die in verschiedenen Semestern erfolgreich erbracht wurden, die früheren,
2. die im selben Semester erfolgreich erbracht wurden, die besseren

berücksichtigt. [6]Dasjenige Wahlpflichtmodul, mit dessen Vorleistung, Modulprüfung oder Modulteilprüfung erstmalig 180 ECTS-Punkte überschritten werden, wird mit

der ihm zugeschriebenen ECTS-Punkte-Zahl nur insoweit berücksichtigt, als 180 ECTS-Punkte nicht überschritten werden.

§ 22
Bachelor-Urkunde, Bachelor-Zeugnis, Transcript of Records und Diploma Supplement

(1) [1]Nach bestandener Bachelorprüfung erhält die oder der Studierende eine Bachelor-Urkunde, die das Datum des Tages trägt, an dem die letzte Modulprüfung oder Modulteilprüfung erbracht worden ist. [2]Darin wird die Verleihung des akademischen Grades gemäß § 2 beurkundet.

(2) [1]Gleichzeitig mit der Bachelor-Urkunde erhält die oder der Studierende das Bachelor-Zeugnis mit dem Datum der Bachelor-Urkunde. [2]In das Bachelor-Zeugnis sind das Thema der Bachelorarbeit und deren Note sowie die Endnote aufzunehmen.

(3) [1]Das Prüfungsamt stellt zusätzlich ein Transcript of Records aus, das alle absolvierten Module und die ihnen zugeordneten Modulprüfungen, Modulteilprüfungen und Vorleistungen einschließlich der dafür vergebenen ECTS-Punkte und Noten beinhaltet. [2]Modulprüfungen, Modulteilprüfungen und Vorleistungen, die nach §§ 19 und 21 nicht in die Bachelorprüfung eingehen, werden nachrichtlich aufgenommen.

(4) Das Prüfungsamt stellt darüber hinaus ein Diploma Supplement mit Informationen über Art und Ebene des Bachelorabschlusses, den Status der Ludwig-Maximilians-Universität München sowie detaillierten Informationen über das Studienprogramm des Bachelorstudiengangs aus.

(5) [1]Die Bachelor-Urkunde wird durch die Dekanin oder den Dekan und durch die Vorsitzende oder den Vorsitzenden des Prüfungsausschusses, das Bachelor-Zeugnis wird durch die Vorsitzende oder den Vorsitzenden des Prüfungsausschusses, das Transcript of Records und das Diploma Supplement werden durch die Leiterin oder den Leiter des Prüfungsamts unterzeichnet. [2]Bachelor-Urkunde, Bachelor-Zeugnis, Transcript of Records und Diploma Supplement werden mit dem Siegel der Ludwig-Maximilians-Universität München versehen.

(6) [1]Ergibt sich nach Ausstellung und Aushändigung einer Bachelor-Urkunde, eines Bachelor-Zeugnisses, eines Transcript of Records, eines Diploma Supplement, eines sonstigen Zeugnisse, einer sonstigen Urkunde oder eines Kontoauszuges, dass unerlaubte Hilfsmittel benutzt wurden oder eine Täuschung begangen wurde, so kann der Prüfungsausschuss nachträglich die betreffenden Noten berichtigen und die Prüfung ganz oder teilweise für nicht bestanden erklären. [2]Die unrichtige Bachelor-Urkunde, das unrichtige Bachelor-Zeugnis, das unrichtige Transcript of Records, das unrichtige Diploma Supplement, ein sonstiges unrichtiges Zeugnis, eine sonstige unrichtige Urkunde oder ein unrichtiger Kontoauszug sind einzuziehen. [3]Falls die Voraussetzungen erfüllt sind, ist eine korrekte Bachelor-Urkunde, ein korrektes Bachelor-Zeugnis, ein korrektes Transcript of Records, ein

korrektes Diploma Supplement, ein korrektes sonstiges Zeugnis, eine korrekte sonstige Urkunde oder ein korrekter abschließender Kontoauszug zu erteilen. [4]Eine derartige Entscheidung ist nach einer Frist von fünf Jahren ab dem Datum des Bachelor-Zeugnisses ausgeschlossen. [5]Vor einer Entscheidung nach Satz 1 und bzw. oder Satz 2 ist Gelegenheit zur Äußerung zu geben. [6]Belastende Entscheidungen sind der oder dem Studierenden unverzüglich schriftlich mitzuteilen, zu begründen und mit einer Rechtsbehelfsbelehrung zu versehen.

IV. Prüfungsorgane und Prüfungsverwaltung

§ 23
Prüfungsausschuss und Prüfungsamt

[Hinweis: Sinnvollerweise wird kein gesonderter Prüfungsausschuss kreiert, sondern ein bereits bestehender für zuständig erklärt oder für alle Bachelor-, Master- und PhD-Studiengänge ein gemeinsamer Prüfungsausschuss geschaffen.]

(1) [1]Der Prüfungsausschuss # *[ggf. Bezeichnung des Prüfungsausschusses]* besteht aus # *[mindestens 3]* Mitgliedern, denen nach Art. 62 Abs. 1 Satz 2 BayHSchG in Verbindung mit der Hochschulprüferverordnung in der jeweils geltenden Fassung (HSchPrüferV) Prüfungsberechtigung zukommen muss. [2]Die Mitglieder des Prüfungsausschusses werden durch den Fakultätsrat bestellt. # *[**Option:** # Mitglieder werden durch den Fakultätsrat der Fakultät #, # Mitglieder durch den Fakultätsrat der Fakultät # bestellt.]* [3]Die Amtszeit der Mitglieder des Prüfungsausschusses beträgt # Jahre. [4]Wiederbestellung ist zulässig.

(2) [1]Die Mitglieder bestellen aus ihrer Mitte eine Vorsitzende oder einen Vorsitzenden sowie deren oder dessen Stellvertreterin oder Stellvertreter. [2]Die Amtszeit der oder des Vorsitzenden und der Stellvertreterin oder des Stellvertreters beträgt # Jahre. [3]Wiederbestellung ist zulässig.

(3) [1]Der Prüfungsausschuss ist beschlussfähig, wenn sämtliche Mitglieder mindestens eine Woche vor der Sitzung schriftlich oder elektronisch unter Angabe der Tagesordnung geladen wurden und die Mehrheit der Mitglieder anwesend und stimmberechtigt ist. [2]Er beschließt mit der Mehrzahl der abgegebenen Stimmen in Sitzungen; Stimmenthaltung, geheime Abstimmung und Stimmrechtsübertragung sind nicht zulässig. [3]Bei Stimmengleichheit gibt die Stimme der oder des Vorsitzenden den Ausschlag. [4]Der Ausschluss eines Mitglieds des Prüfungsausschusses von Beratung und Abstimmung im Prüfungsausschuss und von einer Prüfungstätigkeit bestimmt sich nach Art. 41 Abs. 2 BayHSchG.

(4) [1]Für die Organisation der Prüfungen, die Bestellung der Prüfenden und Beisitzenden (§ 24 Abs. 3) sowie die Entscheidungen in Prüfungssachen ist der Prüfungsausschuss zuständig. [2]Der Prüfungsausschuss wird bei der Erfüllung seiner Aufgaben durch das Prüfungsamt unterstützt. [3]Der Prüfungsausschuss achtet darauf, dass die Bestimmungen dieser Prüfungs- und Studienordnung eingehalten werden. [4]Er berichtet regelmäßig der Studiendekanin oder dem Studiendekan über

die Entwicklung der Prüfungen und Studienzeiten und gibt Anregungen zur Reform und zweckdienlichen Fortschreibung dieser Prüfungs- und Studienordnung.

(5) [1]Der Prüfungsausschuss kann in widerruflicher Weise die Erledigung von bestimmten Aufgaben auf die oder den Vorsitzenden des Prüfungsausschusses, deren oder dessen Stellvertreterin oder Stellvertreter sowie das Prüfungsamt übertragen. [2]Im Übrigen ist die oder der Vorsitzende des Prüfungsausschusses befugt, unaufschiebbare Entscheidungen anstelle des Prüfungsausschusses allein zu treffen; hierüber hat sie oder er den Prüfungsausschuss unverzüglich zu informieren.

(6) Der Prüfungsausschuss soll sich eine Geschäftsordnung geben.

(7) Die Mitglieder des Prüfungsausschusses haben das Recht, den Prüfungen beizuwohnen.

§ 24
Prüfende und Beisitzende

(1) [1]Bei Modulprüfungen, Modulteilprüfungen und Vorleistungen, die nur eine Lehrveranstaltung betreffen und mit Ausnahme der Bachelorarbeit, ist vorbehaltlich Abs. 4 Satz 1 Prüfende oder Prüfender die oder der für die Lehrveranstaltung verantwortliche Veranstaltungsleiterin oder Veranstaltungsleiter. [2]Bei Modulprüfungen, Modulteilprüfungen und Vorleistungen, die mehrere Lehrveranstaltungen unterschiedlicher Veranstaltungsleiterinnen und Veranstaltungsleiter betreffen, bestellt der Prüfungsausschuss allgemein oder im Einzelfall eine Veranstaltungsleiterin oder einen Veranstaltungsleiter als Prüfende oder Prüfenden. [3]Satz 2 gilt entsprechend, wenn die Veranstaltungsleiterin oder der Veranstaltungsleiter nicht prüfungsberechtigt ist (Abs. 4 Satz 1).

(2) [1]Mündliche Modulprüfungen, Modulteilprüfungen und Vorleistungen sind von # *[**Option:** mindestens von einer oder einem Prüfenden in Gegenwart einer oder eines sachkundigen Beisitzenden (Abs. 3 Nr. 1) abzunehmen **oder – falls mehr als eine Prüfende oder ein Prüfender gewollt sind –** # Prüfenden abzunehmen]*. [2]Nicht bestandene Modulprüfungen, Modulteilprüfungen und Vorleistungen müssen von zwei Prüfenden (Abs. 3 Nr. 2) bewertet werden.

(3) Der Prüfungsausschuss bestellt allgemein oder im Einzelfall

1. bei mündlichen Modulprüfungen, Modulteilprüfungen und Vorleistungen die Beisitzenden,
2. bei nicht bestandenen Modulprüfungen, Modulteilprüfungen und Vorleistungen eine zweite Prüfende oder einen zweiten Prüfenden,
3. für die Bachelorarbeit eine Prüfende oder einen Prüfenden (§ 14 Abs. 3) bzw. mehrere Prüfende (§ 14 Abs. 9). *[**Option**: und*
4. *für die Abschlussprüfung (§ 15) eine Prüfende oder einen Prüfenden bzw. mehrere Prüfende.]*

(4) [1]Prüfende können nur diejenigen sein, die nach Art. 62 Abs. 1 Satz 2 BayHSchG in Verbindung mit der HSchPrüferV prüfungsberechtigt sind. [2]Beisitzende müssen sachkundige Personen sein, die mindestens einen Bachelorstudiengang erfolgreich absolviert haben.

(5) Die Durchführung des Prüfungsverfahrens obliegt den einzelnen Prüfenden und Aufsichtspersonen.

§ 25
Studiengangskoordinatorin oder Studiengangskoordinator, Pflichten der Prüfenden

(1) [1]Die Studiengangskoordinatorin oder der Studiengangskoordinator für diesen Bachelorstudiengang wird durch die Fakultät bestellt. [2]Solange keine Bestellung erfolgt ist, nimmt die Studiendekanin oder der Studiendekan die Aufgaben wahr. [3]Die Studiengangskoordinatorin oder der Studiengangskoordinator erfüllt in Zusammenarbeit mit dem Prüfungsausschuss, dem Prüfungsamt und der Zentralen Universitätsverwaltung folgende Aufgaben

1. bei der Einrichtung und eventuellen Änderungen dieses Bachelorstudiengangs:

   a) die Überprüfung der Modellierung dieser Prüfungs- und Studienordnung aus fachlicher Sicht,
   b) die Erstellung der erforderlichen Informationen über diesen Bachelorstudiengang für Studierende und Prüfende,

   # *[**Option:** c) die Koordination dieses Bachelorstudiengangs mit den Studiengangskoordinatorinnen und Studiengangskoordinatoren der Nebenfächer]*.

2. danach: die Koordination und Organisation der Lehrveranstaltungen, Modulprüfungen, Modulteilprüfungen und Vorleistungen, namentlich

   a) die Einberufung einer jährlichen Lehrplankonferenz,
   b) die Zuordnung der konkret stattfindenden Lehrveranstaltungen zu den in dieser Prüfungs- und Studienordnung vorgeschriebenen abstrakten Lehrveranstaltungen,
   c) die Ankündigung der Lehrveranstaltungen im Vorlesungsverzeichnis,
   d) die Eingabe der Lehrveranstaltungen in die Elektronische Datenverarbeitung,
   e) die Terminierung und Raumzuordnung der Lehrveranstaltungen, Modulprüfungen, Modulteilprüfungen und Vorleistungen und
   f) die Eingabe der Benotung bzw. Bewertung in die Elektronische Datenverarbeitung.

Variante a)

(2) [1]Die Prüfenden (§ 24) sind verpflichtet, der Studiengangskoordinatorin oder dem Studiengangskoordinator unverzüglich, spätestens vier Wochen vor Ende des laufenden Semesters, in einer vom Prüfungsamt vorgegebenen standardisierten

Form mitzuteilen, welche Studierenden an ihrer Lehrveranstaltung mit welchem Ergebnis teilgenommen haben. [2]Die Studiengangskoordinatorin oder der Studiengangskoordinator ist verpflichtet, diese Mitteilungen unverzüglich zu überprüfen und sie unverzüglich in der vom Prüfungsamt vorgeschriebenen standardisierten Form an dieses weiterzuleiten. [3]Die Mitteilungen müssen spätestens zwei Wochen vor Ende des laufenden Semesters in korrekter Form im Prüfungsamt vorliegen.

Variante b)

(2) [1]Die Prüfenden (§ 24) sind verpflichtet, der Studiengangskoordinatorin oder dem Studiengangskoordinator unverzüglich in der vom Prüfungsamt vorgegebenen standardisierten Form mitzuteilen, welche Studierenden an ihrer Lehrveranstaltung mit welchem Ergebnis teilgenommen haben. [2]Die Studiengangskoordinatorin oder der Studiengangskoordinator ist verpflichtet, diese Mitteilungen unverzüglich zu überprüfen und sie unverzüglich in der vom Prüfungsamt vorgeschriebenen standardisierten Form an dieses weiterzuleiten. [3]Die Mitteilungen müssen rechtzeitig in korrekter Form im Prüfungsamt vorliegen; das Prüfungsamt gibt spätestens zu Beginn eines jeden Semesters bekannt, wann die Mitteilungen der Studiengangskoordinatorin oder dem Studiengangskoordinator und bzw. oder dem Prüfungsamt vorliegen müssen.

Variante c)

(2) [1]Die Prüfenden (§ 24) sind verpflichtet, dem Prüfungsamt unverzüglich in einer von diesem vorgegebenen standardisierten Form mitzuteilen, welche Studierenden an ihrer Lehrveranstaltung mit welchem Ergebnis teilgenommen haben. [2]Die Mitteilungen müssen rechtzeitig in korrekter Form im Prüfungsamt vorliegen; das Prüfungsamt gibt spätestens zu Beginn eines jeden Semesters bekannt, wann die Mitteilungen dem Prüfungsamt vorliegen müssen.

*# [**Option** 3 oder 4]*Werden die Anforderungen des Satzes *# [Satz 3 in den Varianten a) und b); Satz 2 in der Variante c)]* nicht erfüllt, finden die betreffenden Veranstaltungen in den aktuellen Kontoauszügen (§ 12) keine Berücksichtigung. *# [**Option** 4 oder 5]# [**Option:** Die Studiengangskoordinatorin oder der Studiengangskoordinator **oder** Die oder der Prüfende*] ist verpflichtet, diese Mitteilungen schnellstmöglich dem Prüfungsamt nachzureichen und allen betroffenen Studierenden Einzelbescheinigungen in Bescheidsform mit Rechtsbehelfsbelehrung als Postzustellungsaufträge zu übersenden.

§ 26
Mitwirkungspflichten der Studierenden, Bestätigung von Mitteilungen

[1]Die oder der Studierende ist verpflichtet, den Eingang an sie oder ihn übersandter, den Erhalt ihr oder ihm ausgehändigter oder von ihr oder ihm elektronisch abgerufener Informationen, Mitteilungen und Verwaltungsakte des Prüfungsausschusses oder Prüfungsamtes in der geforderten Form auf ihre oder seine Kosten zu bestätigen (Empfangsbestätigung). [2]Auf dem Gelände der Ludwig-Maximilians-Universität München kann die Empfangsbestätigung kostenlos erfolgen.

[3]Das Prüfungsamt gibt in den ersten beiden Wochen der Vorlesungszeit ortsüblich bekannt, ab wann Informationen, Mitteilungen und Verwaltungsakte ausgehängt oder versandt werden bzw. elektronisch abgerufen oder abgeholt werden können. [4]Für die Zustellung solcher Informationen, Mitteilungen und Verwaltungsakte gelten die allgemeinen gesetzlichen Vorschriften. [5]Gegenüber Studierenden, welche von ausgehängten Informationen, Mitteilungen und Verwaltungsakten keine Kenntnis nehmen, bereit gestellte nicht elektronisch abrufen oder abholen und versandte nicht entgegen nehmen bzw. durch ein Versandunternehmen hinterlegte nicht abholen, gelten diese Informationen, Mitteilungen und Verwaltungsakte einen Monat nach Aushang, Bereitstellung zum elektronischen Abruf oder zur Abholung oder dem Versand als zugegangen und bekannt gegeben. [6]Übermittelt das Prüfungsamt Informationen, Mitteilungen und Verwaltungsakte erneut, weil die oder der Studierende die in Satz 1 vorgesehene Empfangsbestätigung nicht übermittelt und bzw. oder von ausgehängten Informationen, Mitteilungen und Verwaltungsakten keine Kenntnis nimmt, bereit gestellte nicht elektronisch abruft oder abholt und versandte nicht entgegen nimmt bzw. durch ein Versandunternehmen hinterlegte nicht abholt, trägt die oder der Studierende die durch die erneute Übermittlung entstehenden Kosten. [7]Das Prüfungsamt ist zu einem erneuten Übermittlungsversuch nicht verpflichtet.

V. Durchführung der Prüfungen

§ 27
Anerkennung von Studien- und Prüfungsleistungen

(1) [1]Studienzeiten, Studien- und Prüfungsleistungen, die im gleichen Studiengang an anderen staatlichen oder staatlich anerkannten Hochschulen in der Bundesrepublik Deutschland erbracht worden sind, werden anerkannt, es sei denn, dass diese nicht gleichwertig sind. [2]Studienzeiten, Studien- und Prüfungsleistungen, die in einem anderen Studiengang an der Ludwig-Maximilians-Universität München oder an anderen staatlichen oder staatlich anerkannten Hochschulen in der Bundesrepublik Deutschland erbracht worden sind, werden anerkannt, es sei denn, dass diese nicht gleichwertig sind.

(2) [1]Studienzeiten, Studien- und Prüfungsleistungen werden auch durch die erfolgreiche Teilnahme an einer entsprechenden Fernstudieneinheit nachgewiesen, soweit die Einheit dem entsprechenden Lehrangebot des Präsenzstudiums inhaltlich gleichwertig ist; dies gilt entsprechend für die erfolgreiche Teilnahme an Lehrangeboten der Virtuellen Hochschule Bayern. [2]Studienzeiten, Studien- und Prüfungsleistungen propädeutischer Lehrveranstaltungen werden auch durch eine einschlägige, gleichwertige Berufs- oder Schulausbildung nachgewiesen; nach Inhalt und Niveau gleichwertige Studien- und Prüfungsleistungen einer mit Erfolg abgeschlossenen Ausbildung an Fachschulen und Fachakademien werden anerkannt.

(3) Studienzeiten, Studien- und Prüfungsleistungen, die an ausländischen Hochschulen erbracht worden sind, werden in der Regel anerkannt, außer sie sind nicht gleichwertig.

(4) [1]Studienzeiten, Studien- und Prüfungsleistungen sind gleichwertig, wenn sie in Inhalt, Umfang und in den Anforderungen denjenigen dieses Bachelorstudiengangs an der Ludwig-Maximilians-Universität München im Wesentlichen entsprechen. [2]Dabei ist kein schematischer Vergleich, sondern eine Gesamtbetrachtung und Gesamtbewertung vorzunehmen. [3]Bei der Anrechnung von Studienzeiten, Studien- und Prüfungsleistungen, die außerhalb der Bundesrepublik Deutschland erbracht wurden, sind die von der Kultusministerkonferenz und der Hochschulrektorenkonferenz gebilligten Äquivalenzvereinbarungen sowie Absprachen im Rahmen von Hochschulpartnerschaften zu beachten. [4]Bei Zweifeln an der Gleichwertigkeit kann die Zentralstelle für ausländisches Bildungswesen gehört werden.

(5) [1]Die Anerkennung einzelner Studien- oder Prüfungsleistungen nach den vorstehenden Absätzen kann nur in Höhe von maximal # ECTS-Punkten des Umfangs der angestrebten Bachelorprüfung an der Ludwig-Maximilians-Universität München erfolgen. [2]Dabei ist eine Anerkennung der Bachelorarbeit ausgeschlossen. [3]Außerhalb des Hochschulbereichs erworbene Kenntnisse und Fähigkeiten dürfen höchstens die Hälfte des vorgeschriebenen Hochschulstudiums ersetzen.

(6) [1]Werden Studien- oder Prüfungsleistungen anerkannt, sind die Noten – soweit die Notensysteme übereinstimmen – zu übernehmen und nach Maßgabe dieser Prüfungs- und Studienordnung in die Berechnung der Modul- und Endnote einzubeziehen. [2]Die übernommenen Noten werden gekennzeichnet und die Tatsache der Übernahme im Zeugnis vermerkt. [3]Stimmen die Notensysteme nicht überein, wird durch die Vorsitzende oder durch den Vorsitzenden des Prüfungsausschusses für die anerkannte Studien- und Prüfungsleistung unter Zugrundelegung der Bewertungsstufen nach § 10 Abs. 2 eine Note festgesetzt und nach den Sätzen 1 und 2 verfahren. [4]Die Sätze 1 bis 3 gelten für die Zuordnung von ECTS-Punkten entsprechend.

(7) [1]Die für die Anrechnung erforderlichen Unterlagen sind von den Studierenden spätestens am Ende des ersten nach der Immatrikulation in diesen Bachelorstudiengang an der Ludwig-Maximilians-Universität München verbrachten Semesters beim Prüfungsausschuss einzureichen, sofern Studienzeiten und Studien- oder Prüfungsleistungen angerechnet werden sollen, die bereits vor der Immatrikulation an der Ludwig-Maximilians-Universität München in diesen Bachelorstudiengang erbracht wurden. [2]Für die Anrechnung von Studienzeiten und Studien- und Prüfungsleistungen, die nach der Immatrikulation an der Ludwig-Maximilians-Universität München in diesen Bachelorstudiengang erbracht werden, sind die Unterlagen im jeweils auf den Erwerb folgenden Semester einzureichen. [3]Der Nachweis von anzurechnenden Studienzeiten wird im Regelfall durch Vorlage des Studienbuchs der Hochschule, an der die Studienzeit zurückgelegt wurde, erbracht. [4]Für die Anerkennung von Studien- und Prüfungsleistungen ist eine Bescheinigung derjenigen Hochschule, an der die Prüfungsleistungen erbracht wurden, vorzulegen, aus der sich ergeben muss,

1. welche Einzelprüfungen (mündlich und/oder schriftlich) in welchen Prüfungsfächern im Rahmen der Gesamtprüfung abzulegen waren,
2. welche Prüfungen tatsächlich abgelegt wurden,

3. die Bewertung der Prüfungsleistungen sowie ggf. die Fachnote,
4. das der Bewertung zu Grunde liegende Notensystem,
5. bei Studiengängen mit Leistungspunktesystemen die für die einzelnen Lehrveranstaltungen, in denen die anzuerkennenden Studien- und Prüfungsleistungen erbracht wurden, vergebenen Leistungspunkte sowie die Anzahl der Leistungspunkte, welche für einen erfolgreichen Abschluss des Studiengangs erforderlich ist,
6. der Umfang der einzelnen Lehrveranstaltungen, in denen die anzuerkennenden Prüfungsleistungen erbracht wurden, in Semesterwochenstunden und
7. ob eine Gesamtprüfung auf Grund der vorliegenden Ergebnisse nicht bestanden ist oder auf Grund anderer Umstände als nicht bestanden gilt.

(8) Bei Zeugnissen und Unterlagen, die nicht in deutscher Sprache ausgestellt sind, kann die Vorlage einer beglaubigten deutschen Übersetzung verlangt werden.

(9) Über die Anrechnung von Studienzeiten, Studien- und Prüfungsleistungen entscheidet der Prüfungsausschuss, in Zweifelsfällen nach Anhörung der zuständigen Fachvertreterin oder des zuständigen Fachvertreters.

§ 28
Belegung von Lehrveranstaltungen und Anmeldung zu Modulprüfungen, Modulteilprüfungen und Vorleistungen

(1) [1]Der Prüfungsausschuss kann für einzelne oder alle Lehrveranstaltungen vorschreiben, dass für eine Teilnahme an der Lehrveranstaltung eine Belegung erforderlich ist sowie deren Form und Frist regeln. [2]Studierende, die eine Lehrveranstaltung, für die nach Satz 1 eine Belegung vorgeschrieben wurde, nicht oder nicht form- und bzw. oder nicht fristgerecht belegt haben, haben keinen Anspruch auf Teilnahme an dieser Lehrveranstaltung. [3]Die Lehrveranstaltungen, für welche eine Belegung erforderlich ist, sowie die Form und Frist der jeweiligen Belegung werden in den ersten beiden Wochen nach Semesterbeginn durch das Prüfungsamt ortsüblich bekannt gegeben; eine Bekanntgabe durch das Prüfungsamt ausschließlich im Internet ist ausreichend.

(2) [1]Der Prüfungsausschuss kann für einzelne oder alle Modulprüfungen, Modulteilprüfungen und Vorleistungen eine Anmeldung sowie deren Form und Frist vorschreiben. [2]Studierende, die sich zu einer Modulprüfung, Modulteilprüfung oder Vorleistung, für die nach Satz 1 eine Anmeldung vorgeschrieben wurde, nicht oder nicht form- und bzw. oder nicht fristgerecht angemeldet haben, haben keinen Anspruch auf Teilnahme an dieser Modulprüfung, Modulteilprüfung oder Vorleistung. [3]Der Prüfungsausschuss kann darüber hinaus allgemein anordnen, dass eine Modulprüfung, Modulteilprüfung oder Vorleistung, für die nach Satz 1 eine Anmeldung vorgeschrieben wurde, als nicht bestanden gilt, wenn die oder der Studierende aus selbst zu vertretenden Gründen nicht antritt oder von der angetretenen Modulprüfung, Modulteilprüfung oder Vorleistung zurücktritt. [4]Abs. 1 Sätze 2 und 3 gelten für die Modulprüfungen, Modulteilprüfungen und Vorleistungen, für welche eine Anmeldung erforderlich ist, sowie die Form und Frist der jeweiligen Anmeldung entsprechend.

(3) [1]Über die Bekanntgaben nach Abs. 1 Satz 3 und Abs. 2 Satz 4 wird ein schriftliches Protokoll erstellt, das insbesondere Angaben über den Inhalt der Festlegungen sowie Zeit, Art und Ort von deren Bekanntgabe enthält. [2]Das Protokoll wird durch die Vorsitzende oder den Vorsitzenden des Prüfungsausschusses unterschrieben und durch das Prüfungsamt mindestens fünf Jahre aufbewahrt.

§ 29
Versäumnis, Rücktritt

(1) [1]Eine Modulprüfung, Modulteilprüfung oder Vorleistung gilt als „nicht bestanden“ bzw. mit „nicht ausreichend“ (5,0) bewertet, wenn die oder der Studierende

1. bei einer Modulprüfung, Modulteilprüfung oder Vorleistung, für die er oder sie sich angemeldet hat und der Prüfungsausschuss eine Anordnung nach § 28 Abs. 2 Satz 3 getroffen hat, einen Prüfungstermin aus einem selbst zu vertretenden Grund versäumt oder
2. von einer Modulprüfung, Modulteilprüfung oder Vorleistung, die sie oder er angetreten hat, aus einem selbst zu vertretenden Grund zurücktritt oder
3. eine schriftliche Modulprüfung, Modulteilprüfung oder Vorleistung nicht innerhalb der vorgegebenen Bearbeitungszeit erbracht hat.

(2) [1]Der Grund für den Rücktritt oder das Versäumnis muss beim Prüfungsamt unverzüglich schriftlich geltend und glaubhaft gemacht werden. [2]§ 11 Abs. 5 Sätze 4 bis 7 gelten entsprechend.

§ 30
Täuschung, Ordnungsverstoß, fehlende Teilnahmevoraussetzungen

(1) Versucht die oder der Studierende, das Ergebnis einer Modulprüfung, Modulteilprüfung oder Vorleistung durch Täuschung oder Benutzung nicht zugelassener Hilfsmittel zu eigenem oder fremden Vorteil zu beeinflussen, wird die betreffende Modulprüfung, Modulteilprüfung oder Vorleistung mit „nicht bestanden“ bzw. „nicht ausreichend“ (5,0) bewertet; als Versuch gilt bei schriftlichen Modulprüfungen, Modulteilprüfungen und Vorleistungen bereits der Besitz nicht zugelassener Hilfsmittel während und nach Ausgabe der Prüfungsunterlagen.

(2) Eine Studierende oder ein Studierender, die oder der den ordnungsgemäßen Ablauf des Prüfungstermins stört, kann von der oder dem jeweiligen Prüfenden oder Aufsichtführenden von der Fortsetzung der Modulprüfung, Modulteilprüfung oder Vorleistung ausgeschlossen werden; in diesem Fall wird die Modulprüfung, Modulteilprüfung oder Vorleistung mit „nicht bestanden“ bzw. „nicht ausreichend“ (5,0) bewertet.

(3) In schwerwiegenden oder wiederholten Fällen des Abs. 1 und bzw. oder des Abs. 2 kann der Prüfungsausschuss die Studierende oder den Studierenden von der Erbringung einzelner oder aller weiteren Modulprüfungen, Modulteilprüfungen und

Vorleistungen ausschließen; im letzteren Fall wird die oder der Studierende gemäß Art. 49 Abs. 2 Nr. 3 BayHSchG exmatrikuliert.

(4) Waren die Voraussetzungen für die Teilnahme an einer Modulprüfung, Modulteilprüfung oder Vorleistung nicht erfüllt, gilt sie als nicht abgelegt.

(5) § 22 Abs. 6 Sätze 5 und 6 gelten entsprechend.

§ 31
Schutzfristen nach dem Mutterschutzgesetz, Erziehungsurlaub

(1) Die Inanspruchnahme der Schutzfristen entsprechend den §§ 3, 4, 6 und 8 des Mutterschutzgesetzes sowie entsprechend den Fristen des Gesetzes zum Erziehungsgeld und zur Elternzeit (Bundeserziehungsgeldgesetz – BErzGG) in der Fassung der Bekanntmachung vom 9. Februar 2004 (BGBl I S. 206) in der jeweils geltenden Fassung wird ermöglicht.

(2) [1]Der Prüfungsausschuss legt fest, welche Lehrveranstaltungen für schwangere oder stillende Studierende mit überdurchschnittlichen Gefahren verbunden sind und verbindet dies mit einer entsprechenden Warnung. [2]Der Prüfungsausschuss untersagt die Teilnahme schwangerer oder stillender Studierende an Lehrveranstaltungen, die mit erheblich über dem Durchschnitt liegenden Gefahren für Mutter und bzw. oder Kind verbunden sind. [3]Der Prüfungsausschuss legt fest, ob und wie schwangere oder stillende Studierende die Kenntnisse und Fähigkeiten, die in Lehrveranstaltungen vermittelt werden, an denen sie nicht teilnehmen dürfen, anderweitig erwerben können. [4]Ein Rechtsanspruch auf die Zurverfügungstellung eines besonderen Lehrangebots für schwangere oder stillende Studierende besteht nicht. [5]Die Lehrveranstaltungen, Warnungen und Untersagungen nach den Sätzen 1 und 2 sowie die Möglichkeit eines anderweitigen Erwerbs der Kenntnisse und Fähigkeiten nach Satz 3 werden durch das Prüfungsamt ortsüblich bekannt gegeben; eine Bekanntgabe durch das Prüfungsamt ausschließlich im Internet ist ausreichend.

§ 32
Prüfungsvergünstigungen für Schwerbehinderte

[1]Schwerbehinderten werden vom Prüfungsausschuss auf schriftlichen, spätestens bei der Anmeldung zu einer Modulprüfung, Modulteilprüfung oder Vorleistung oder spätestens einen Monat vor der jeweiligen Modulprüfung, Modulteilprüfung oder Vorleistung zu stellenden Antrag Prüfungsvergünstigungen gewährt. [2]Insbesondere ist, falls die Art der Behinderung es rechtfertigt, die Bearbeitungszeit bei schriftlichen Modulprüfungen, Modulteilprüfungen und Vorleistungen um bis zu einem Viertel zu verlängern. [3]Sofern die Art der Behinderung es erforderlich macht, kann der Prüfungsausschuss im Einzelfall weitere Prüfungsvergünstigungen gewähren. [4]Behindert ist, wer wegen einer körperlichen Behinderung nicht in der Lage ist, die Modulprüfung, Modulteilprüfung oder Vorleistung ganz oder teilweise in der vorgesehenen Form abzulegen. [5]Die Behinderung ist glaubhaft zu machen. [6]Das

Prüfungsamt kann fordern, dass die Glaubhaftmachung durch ein ärztliches Attest erfolgt. [7]§ 11 Abs. 5 Sätze 4 und 5 gelten entsprechend.

§ 33
Mängel im Prüfungsverfahren

(1) Erweist sich, dass ein Prüfungsverfahren mit wesentlichen Mängeln behaftet war, die das Prüfungsergebnis beeinflusst haben können, so ist auf Antrag einer Teilnehmerin oder eines Teilnehmers oder von Amts wegen anzuordnen, dass von bestimmten oder von allen Teilnehmerinnen und Teilnehmern die gesamte Modulprüfung, Modulteilprüfung oder Vorleistung oder ein einzelner Teil derselben wiederholt wird.

(2) [1]Angebliche Mängel im Prüfungsverfahren oder eine vor oder während der Modulprüfung, Modulteilprüfung oder Vorleistung eingetretene Prüfungsunfähigkeit müssen unverzüglich, spätestens jedoch vor Bekanntgabe des Prüfungsergebnisses, beim Aufsichtsführenden, bei der Prüfenden oder dem Prüfenden, beim Prüfungsamt oder bei der oder dem Vorsitzenden des Prüfungsausschusses geltend und glaubhaft gemacht werden. [2]Mündlich geltend und glaubhaft gemachte Gründe im Sinn von Satz 1 sind unverzüglich auch schriftlich beim Prüfungsamt oder bei der oder dem Vorsitzenden des Prüfungsausschusses geltend und glaubhaft zu machen. [3]Die Geltend- und Glaubhaftmachung ist in jedem Fall ausgeschlossen, wenn seit dem Tag, an dem die Modulprüfung, Modulteilprüfung oder Vorleistung erbracht wurde, ein Monat verstrichen ist. [4]§ 11 Abs. 5 Sätze 3 bis 7 gelten entsprechend.

§ 34
Einsicht in die Prüfungsakten, Aufbewahrungsfristen

[1]Innerhalb # *[**Option:** {mindestens drei Monate} oder eines durch das Prüfungsamt ortsüblich bekannt gegebenen Zeitraums]* nach Abschluss einer Modulprüfung, Modulteilprüfung oder Vorleistung wird der oder dem Studierenden beim Prüfungsamt auf Antrag in angemessener Frist Einsicht in dieselbe, die darauf bezogenen Gutachten und Protokolle gewährt # *[**Option:**; eine Bekanntgabe des Zeitraums durch das Prüfungsamt ausschließlich im Internet ist ausreichend]*. [2]Die vollständigen Prüfungsakten werden mindestens fünf Jahre aufbewahrt. [3]Die Grundakte, die aus Abschriften des Bachelor-Zeugnisses, der Bachelor-Urkunde und des Transcript of Records besteht, wird unbegrenzte Zeit aufbewahrt. [4]Die Aufbewahrung kann in elektronischer Form erfolgen.

VI. Schlussbestimmungen

§ 35
Inkrafttreten und Übergangsvorschriften

Diese Satzung tritt am # in Kraft.

Fakultät für Informatik
der Technischen Universität München

Prüfungsausschuss der Fakultät für Informatik

- Home
- Aktuelle Mitteilungen
- Studienberatung
- Studiensekretariat
- Bachelor-Studiengang Informatik
  - Eignungsfeststellung
  - Prüfungsordnung
  - Anwendungsfach
  - Bachelor-Arbeit
  - FAQ
- Alle Studiengänge
- Anerkennungen
- Eignungsfeststellung
- Bafög

FACHPRÜFUNGSORDNUNG FÜR DEN BACHELORSTUDIENGANG INFORMATIK

an der Technischen Universität München

Vom 24. Oktober 2005

Bitte beachten:

Rechtsverbindlich ist allein die in der Hochschule am 24. Oktober 2005 niedergelegte Fassung der Fachprüfungsordnung für den Bachelorstudiengang Informatik

---

Aufgrund von Art. 6 in Verbindung mit Art. 81 Abs. 1 und Art. 86 a des Bayerischen Hochschulgesetzes (BayHSchG) erlässt die Technische Universität München folgende Fachprüfungsordnung.

Vorbemerkung zum Sprachgebrauch: Nach Art. 3 Abs. 2 des Grundgesetzes sind Frauen und Männer gleichberechtigt. Alle maskulinen Personen- und Funktionsbezeichnungen in dieser Satzung gelten daher für Frauen und Männer in gleicher Weise.

INHALTSÜBERSICHT

I. Allgemeine Bestimmungen

§ 1 Geltungsbereich, akademischer Grad, verwandte Studiengänge
§ 2 Gliederung des Studiums, Regelstudienzeit, ECTS, Modularisierung
§ 3 Zweck der Prüfungen
§ 4 Prüfungsausschuss
§ 5 Anrechnung von Prüfungsleistungen
§ 6 Studienbegleitendes Prüfungsverfahren
§ 7 Punktekontensystem
§ 8 Studienleistungen
§ 9 Anmeldung zu Prüfungen, Prüfungsfristen, ECTS-Hürden
§ 10 Wiederholung von Prüfungen

II. Bachelorprüfung

§ 11 Zulassung zur Bachelorprüfung
§ 12 Umfang der Bachelorprüfung
§ 13 Bachelor's Thesis

§ 14 Zusatzprüfungen
§ 15 Freier Prüfungsversuch
§ 16 Bestehen und Bewertung der Bachelorprüfung
§ 17 Zeugnis, Urkunde, Diploma Supplement

III. Schlussbestimmung

§ 18 In-Kraft-Treten

Anlagen

Anlage 1: Prüfungsfächer
Anlage 2: Anwendungsfächer

I. ALLGEMEINE BESTIMMUNGEN

§ 1 Geltungsbereich, akademischer Grad, verwandte Studiengänge

(1) Soweit nachfolgend nichts anderes bestimmt ist, gelten die Regelungen der Allgemeinen Diplomprüfungsordnung (ADPO) der Technischen Universität München in der jeweils geltenden Fassung entsprechend.

(2) [1]Aufgrund der bestandenen Bachelorprüfung wird der akademische Grad „Bachelor of Science” („B.Sc.”) verliehen. [2]Der akademische Grad kann mit dem Hochschulzusatz „(TUM)” geführt werden.

(3) [1]Der Bachelorstudiengang Informatik und der Diplomstudiengang Informatik an der Technischen Universität München sind verwandte Studiengänge. [2]Beim Wechsel von einer anderen Universität an die Technische Universität München entscheidet der zuständige Prüfungsausschuss über die Verwandtheit des Studienganges.

§ 2 Gliederung des Studiums, Regelstudienzeit, ECTS, Modularisierung

(1) Das Bachelorstudium gliedert sich in ein viersemestriges Grundstudium und ein zweisemestriges Hauptstudium.

Das Bachelorstudium wird mit der Bachelorprüfung abgeschlossen.

(2) [1]Der Höchstumfang der für die Erlangung des Bachelorgrades erforderlichen Lehrveranstaltungen im Pflicht-, Wahlpflicht- und Wahlbereich beträgt 165 Credits (122 SWS), wovon 31 Credits auf ein Anwendungsfach entfallen. [2]Hinzu kommen 15 Credits für die Erstellung der Bachelor's Thesis mit einem Vortrag mit fachlicher Ausprache im Rahmen eines Kolloquiums (12+3 Credits). [3]Die Regelstudienzeit für das Bachelorstudium beträgt damit insgesamt sechs Semester.

(3) [1]Der Umfang der zu erbringenden Prüfungsleistungen errechnet sich

aufgrund der Anzahl der in Credits gemessenen Lehrveranstaltungsstunden gemäß des European Credit Transfer System (ECTS). [2]Das System erfordert neben der Feststellung der erfolgreichen Teilnahme auch eine Bewertung oder eine Benotung. [3]Die Lehrveranstaltungsstunde wird mit einer Gewichtung von mindestens einem, höchstens zwei Credits umgerechnet, wobei als Zwischenwert nur eine Vergabe von 1,5 Credits möglich ist. [4]Der Regelumfang eines Semesters beträgt 30 Credits.

(4) Der Umfang der zu erbringenden Prüfungsleistungen im Pflicht-, Wahlpflicht- und Wahlbereich gemäß Anlage 1 und 2 im Bachelorstudiengang Informatik beträgt 180 Credits.

(5) [1]Das Bachelorstudium ist modular aufgebaut. [2]Ein Modul im Sinne dieser Prüfungsordnung zeichnet einen Verbund von thematisch und zeitlich aufeinander abgestimmten Lehrveranstaltungen aus. [3]Module können sich aus verschiedenen Lehr- und Lernformen (wie z.B. Vorlesungen, Übungen, Praktika und ähnliches) zusammensetzen. [4]Ein Modul kann Inhalte eines einzelnen Semesters oder eines Studienjahres umfassen, sich aber auch über mehrere Semester erstrecken.

§ 3 Zweck der Prüfungen

[1]Die Bachelorprüfung bildet einen ersten berufsbefähigenden Abschluss des Studiums der Informatik. [2]Durch die Prüfung wird festgestellt, ob der Student die wissenschaftlichen Grundlagen des Fachgebiets beherrscht, Methodenkompetenz sowie berufsfeldbezogene Qualifikationen erworben hat, auf die im Masterstudiengang oder in einer beruflichen Tätigkeit aufgebaut werden kann.

§ 4 Prüfungsausschuss

Die für Entscheidungen in Prüfungsangelegenheiten zuständige Stelle gemäß § 5 ADPO ist für die Bachelorprüfung der Prüfungsausschuss der Fakultät für Informatik.

§ 5 Anrechnung von Prüfungsleistungen

(1) [1]Prüfungsleistungen, die an einer ausländischen Hochschule erbracht worden sind, werden in der Regel angerechnet, außer sie sind nicht gleichwertig. 2Über die Gleichwertigkeit entscheidet der Prüfungsausschuss. 3Über die Anerkennung von Studien- und Prüfungsleistungen entscheidet der Prüfungsausschuss.

(2) [1]Die Bachelor's Thesis muss im Rahmen des Bachelor-Studiums Informatik an der Technischen Universität München erstellt werden. [2]Darüber hinaus müssen Prüfungsleistungen im Umfang von mindestens weiteren 75 Credits im Rahmen der Bachelorprüfung an der Technischen Universität München erbracht werden.

(3) Ein Antrag auf Anerkennung sämtlicher Prüfungsleistungen aus

früheren Studien kann nur einmal beim Prüfungsausschuss gestellt werden.

§ 6 Studienbegleitendes Prüfungsverfahren

(1) Die Fachprüfungen der Bachelorprüfung werden in der Regel studienbegleitend abgelegt.

(2) [1]Fachprüfungen werden schriftlich oder mündlich in Form einer Abschlussprüfung oder geteilt abgehalten. [2]Zusätzlich können auch Zulassungsvoraussetzungen in Form von Hausaufgaben gefordert werden. [3]Studenten, die einmal an einer Fachprüfung teilgenommen haben, sind damit ohne Nachweis etwaiger Zulassungsvoraussetzungen für Wiederholungsprüfungen in diesem Fach zugelassen. [4]Die Entscheidung, auf welche Art eine Fachprüfung durchgeführt wird und ob Zulassungsvoraussetzungen in Form von Hausaufgaben gefordert werden, treffen die fachlich zuständigen Prüfer in Abstimmung mit dem Prüfungsausschuss. [5]Für ein Fach können Prüfungsleistungen in mehreren Formen verlangt werden.

[6]Findet die Prüfung im selben Semester wie die Lehrveranstaltung statt, so sind dem Studenten die Prüfungsart und die Prüfungsdauer zu Beginn der Lehrveranstaltung in geeigneter Weise bekannt zu geben.
[7]Ansonsten müssen die Prüfungsart und die Prüfungsdauer spätestens 14 Tage vor der betreffenden Prüfung bekannt gegeben werden.

[8]Mündliche Einzelprüfungen dauern mindestens 20 und höchstens 60 Minuten, schriftliche Prüfungen mindestens 45 und höchstens 300 Minuten. [9]Mündliche Mehrfachprüfungen dauern mindestens 15 Minuten je Kandidat.

(3) [1]Jedem Prüfungsfach werden die in Anlage 1 und 2 jeweils aufgeführten Credits zugeordnet, deren Festlegung unter Beachtung des § 2 Abs. 3 zu erfolgen hat. [2]Diese sind ein Maß für den Arbeitsaufwand, der für die Studenten mit der Belegung dieses Faches verbunden ist.
[3]Die Credits sind erbracht, wenn die entsprechende Fachprüfung mindestens mit „ausreichend” (4,0) bewertet worden ist.

(4) Auf Antrag des Studenten und mit Zustimmung der Prüfer können bei deutschsprachigen Lehrveranstaltungen Prüfungen in englischer Sprache abgelegt werden.

(5) Die Bewertung einer Fachprüfung durch den Prüfer muss spätestens vier Wochen nach Ablegung der letzten Prüfungsleistung abgeschlossen sein und muss per Aushang oder Bescheid dem Prüfling bekannt gegeben werden.

§ 7 Punktekontensystem

(1) 1Über die Teilnahme an Fachprüfungen werden Punktekonten geführt. [2]Für jeden zur Bachelorprüfung zugelassenen Studenten wird

beim Prüfungsausschuss ein Bonus- und ein Maluspunktekonto eingerichtet.

(2) [1]Das Bonuspunktekonto enthält die Summe aller Credits der im Rahmen des Bachelorstudiengangs Informatik bestandenen Fachprüfungen. [2]Das Bonuspunktekonto wächst während der gesamten Studiendauer an.

(3) [1]Das Maluspunktekonto enthält die Summe an Credits aller nicht bestandenen Wiederholungsprüfungen. [2]Der Stand des Maluspunktekontos entscheidet über die Zulassung zur zweiten Wiederholung von Fachprüfungen.

§ 8 Studienleistungen

Im Bachelorstudiengang Informatik sind vorbehaltlich der Regelung in § 6 Abs. 2 Satz 2 keine Studienleistungen zu erbringen.

§ 9 Anmeldung zu Prüfungen, Prüfungsfristen, ECTS-Hürden

(1) [1]Zur Teilnahme an einer Fachprüfung ist eine Meldung in der durch Aushang bekannt gegebenen Form spätestens drei Wochen vor dem Termin der ersten Prüfungsleistung beim zuständigen Prüfungsausschuss erforderlich. [2]Diese Meldung gilt zugleich als bedingte Meldung zu der entsprechenden Wiederholungsprüfung zum nächstmöglichen Prüfungstermin. [3]Ist eine Prüfung in mehrere Prüfungsleistungen aufgeteilt, so gilt die Anmeldung zu einer Prüfungsleistung zugleich als Meldung zu allen anderen zu dieser Prüfung gehörenden Prüfungsleistungen.

(2) [1]Eine Meldung gilt als vorzeitig im Sinne von § 13 Abs.1 Nr. 4 Satz 2 ADPO, wenn die Meldung zu dem in Abs. 3 Sätze 1 und 4 genannten Zeitpunkt erfolgt ist. [2]Bei einer vorzeitigen Meldung ist ein Rücktritt zulässig, wenn dieser bei einer mündlichen Prüfung spätestens sieben Tage und bei schriftlichen Prüfungen spätestens einen Tag vor dem Tag der ersten Prüfungsleistung, für die er erklärt wird, dem Prüfungsausschuss zugeht.

(3) [1]Ein Student soll sich so rechtzeitig zu den Fachprüfungen, die in Anlage 1 als Grundlagenprüfung gekennzeichnet sind, anmelden, dass er diese bis spätestens zum Ende des zweiten Semesters ablegen kann. [2]Diese Prüfungen müssen bis spätestens Ende des zweiten Semesters erstmals abgelegt werden. [3]Anderenfalls gelten diese Prüfungen als erstmals abgelegt und nicht bestanden sofern keine nach § 13 Abs. 2 ADPO anerkannten Gründe vorliegen. [4]Zu den übrigen Fachprüfungen der Bachelorprüfung soll er sich so rechtzeitig anmelden, dass er diese bis spätestens zum Ende des sechsten Semesters ablegen kann. [5]Die übrigen Fachprüfungen der Bachelorprüfung müssen bis spätestens Ende des neunten Semesters erstmals abgelegt werden. [6]Satz 3 gilt

entsprechend.

(4) Hat ein Student ohne gemäß § 13 Abs. 2 ADPO anerkannte Gründe
nach vier Semestern weniger als 51 Credits oder
nach sechs Semestern weniger als 102 Credits
erreicht, so hat er die Bachelor-Prüfung endgültig nicht bestanden.

§ 10 Wiederholung von Prüfungen

(1) Eine Fachprüfung ist bestanden, wenn sie mindestens mit „ausreichend" (4,0) bewertet worden ist.

(2) [1]Ist die Fachprüfung nicht bestanden, so kann sie in den betroffenen Fächern wiederholt werden. [2]Die Wiederholungsprüfung ist zum nächstmöglichen Prüfungstermin abzulegen. [3]Geschieht dies nicht, so gilt die Wiederholungsprüfung als abgelegt und nicht bestanden. [4]Nicht bestandene Prüfungen in einem Pflicht- oder Wahlpflichtfach können nicht durch eine bestandene Fachprüfung in einem anderen Fach ersetzt werden.

(3) [1]Für studienbegleitende Prüfungen des Grundstudiums muss jedes Semester eine Wiederholungsmöglichkeit angeboten werden. [2]Für studienbegleitende Prüfungen des Hauptstudiums muss spätestens jedes zweite Semester eine Wiederholungsmöglichkeit angeboten werden. [3]Wiederholungsprüfungen können in einer anderen Prüfungsart durchgeführt werden.

(4) Für den Fall, dass die Prüfung nicht bestanden wird, gilt jede Meldung zu einer Prüfung zugleich als Meldung zur entsprechenden Wiederholungsprüfung zum nächsten Prüfungstermin.

(5) [1]Eine nicht bestandene Prüfung kann bis zu zweimal wiederholt werden. [2]Eine zweite Wiederholung ist nur bis zu einem Maluspunktekontostand von 90 Credits zulässig.

II. BACHELORPRÜFUNG

§ 11 Zulassung zur Bachelorprüfung

Voraussetzung für die Zulassung zur Bachelorprüfung ist die Immatrikulation als Student des Bachelor-Studiengangs Informatik der Technischen Universität München in dem Semester, dem der Prüfungstermin zugerechnet wird.

§ 12 Umfang der Bachelorprüfung

(1) Die Bachelorprüfung umfasst:

1. Die Fachprüfungen in den Pflichtfächern Informatik gemäß Anlage 1A (83 Credits);
2. die Fachprüfungen in den Wahlfächern Informatik gemäß Anlage

1B (10 Credits);
3. die Fächprüfungen in den Pflichtfächern Mathematik gemäß Anlage 1C (35 Credits);
4. die Fächprüfungen in den Wahlfächern Überfachliche Grundlagen gemäß Anlage 1D (6 Credits);
5. die Fachprüfungen in den Pflicht-, Wahl- und Wahlpflichtfächern im Anwendungsfach gemäß Absatz 2 und Anlage 1E (31 Credits);
6. die Bachelor's Thesis gemäß § 13.

(2) [1]Die Fachprüfungen im Anwendungsfach umfassen 31 Credits. [2]Das Anwendungsfach muss eine mit dem Ziel der Ausbildung und Prüfung zu vereinbarende sinnvolle Fächerkombination ergeben. [3]In Anlage 2 sind verschiedene Anwendungsfächer aufgelistet. [4]Spätestens zu Beginn des dritten Fachsemesters entscheidet sich der Student für ein Anwendungsfach. [5]Auf Antrag kann der Prüfungsausschuss weitere Anwendungsfächer genehmigen.

(3) [1]Eine nicht bestandene Prüfung im Wahlfachbereich kann innerhalb der Meldefristen durch eine bestandene Fachprüfung in einem anderen Fach des gleichen Katalogs ersetzt werden. [2]Sind am Ende des neunten Semesters die Prüfungen in den Wahlfächern noch nicht bestanden, so muss der Student dem Prüfungsausschuss schriftlich mitteilen, in welchem nicht bestandenen Wahlfach er die Wiederholungsprüfung ablegen möchte.

§ 13 Bachelor's Thesis

(1) Jeder Student hat im Rahmen der Bachelorprüfung eine Bachelor's Thesis anzufertigen und diese mit einem Vortrag mit fachlicher Aussprache im Rahmen eines Kolloquiums vorzustellen.

(2) [1]Betreuer einer Bachelor's Thesis kann jede prüfungsberechtigte Person der Fakultät für Informatik der Technischen Universität München sein. [2]Die Bachelor's Thesis darf mit Zustimmung des Betreuers ganz oder teilweise an einer anderen Fakultät der Technischen Universität München oder in einer Einrichtung außerhalb der Hochschule durchgeführt werden.

(3) [1]Die Bachelor's Thesis muss im sechsten Semester innerhalb des in Abs. 4 Satz 1 genannten Zeitraums gefertigt werden. [2]Sie muss so rechtzeitig begonnen werden, dass sie spätestens im neunten Fachsemester fertiggestellt werden kann. [3]Anderenfalls gilt sie als erstmals abgelegt und nicht bestanden.

(4) [1]Die Zeit von der Ausgabe bis zur Ablieferung der Bachelor's Thesis darf vier Monate nicht überschreiten. [2]Auf schriftlichen Antrag des Studenten kann die Bearbeitungsfrist in besonders begründeten Ausnahmefällen und mit Genehmigung des Prüfungsausschusses im Einvernehmen mit dem Themensteller um höchstens einen Monat verlängert werden.

(5) [1]Die Bachelor's Thesis kann in deutscher oder englischer Sprache angefertigt werden. [2]Es muss eine Zusammenfassung in der jeweils anderen Sprache vorangestellt sein.

(6) [1]Die Bewertung der Bachelor's Thesis erfolgt innerhalb von zwei Monaten in der Regel durch den Betreuer und einen weiteren Prüfer. [2]Von der Bestellung eines zweiten Prüfers kann auf Beschluss des Prüfungsausschusses in Ausnahmefällen abgesehen werden, wenn kein zweiter fachkundiger Prüfer zur Verfügung steht oder seine Bestellung das Prüfungsverfahren unangemessen verzögern würde. [3]Wird die Arbeit vom Betreuer als nicht bestanden bewertet, so muss sie von einem zweiten, dem Fach der Bachelor's Thesis möglichst nahestehenden Prüfer bewertet werden.

(7) [1]Die Bachelor's Thesis ist erfolgreich abgeschlossen, wenn sie mit mindestens „ausreichend" (4,0) bewertet wird. [2]Die Note für die Bachelor's Thesis wird als ungewichteter arithmetischer Mittelwert aus den Einzelnoten der Prüfer gebildet und an die Notenskala des § 16 Abs. 1 und 2 ADPO angeglichen. [3]Für die bestandene Bachelor's Thesis einschließlich Vortrag mit fachlicher Aussprache (siehe Abs. 1) werden 15(=12+3) Credits vergeben. [4]Durch die Bewertung des Vortrages mit fachlicher Aussprache durch den Betreuer kann die Note auf den nächsten Wert der Notenskala des § 16 Abs. 1 und 2 ADPO erhöht oder erniedrigt werden.

(8) Ist die Bachelor's Thesis nicht bestanden, so kann sie einmal mit neuem Thema wiederholt werden.

§ 14 Zusatzprüfugen

(1) Ab dem sechsten Fachsemester kann der Student Prüfungen zu zusätzlichen Wahlfächern aus den Modulkatalogen Wahlfächer Informatik (Anlage 1B) und Überfachliche Grundlagen (Anlage 1D) ablegen, die für die nach § 12 Abs. 1 Nrn. 2 und 4 geforderten Credits nicht mehr benötigt werden.

(2) [1]Die Ergebnisse der Zusatzprüfungen fließen nicht in die Gesamtnote der Bachelorprüfung ein und werden nicht im Bachelor-Zeugnis vermerkt. [2]Die Zusatzprüfungen werden mit den erzielten Ergebnissen jedoch im Diploma Supplement ausgewiesen.

(3) [1]Wird in einem Semester nur ein Teil der bestandenen Wahlfachprüfungen zum Erreichen der Credits nach § 12 Abs. 1 Nrn. 3 und 4 benötigt, so zählen für die Bachelorprüfung die Fachprüfungen mit den besseren Ergebnissen. [2]Die anderen Prüfungen werden als Zusatzprüfungen gewertet. [3]Ist nach diesem Verfahren mehr als eine Aufteilung möglich, so wählt der Prüfungsausschuss zufällig eine aus. [4]Der Student kann bis spätestens zwei Wochen nach Bekanntgabe der Prüfungsergebnisse eine davon abweichende Aufteilung beantragen.

(4) Nicht bestandene Zusatzprüfungen können einmal wiederholt werden.

§ 15 Freier Prüfungsversuch

(1) Studienbegleitende Prüfungen zu Grundlagenfächern, für die gemäß Anlage 1 ein freier Prüfungsversuch vorgesehen ist, gelten nach § 18 Abs. 1 ADPO als nicht abgelegt, wenn sie im ersten Semester abgelegt und erstmals nicht bestanden werden.

(2) Im Anwendungsfach gelten nach § 18 Abs. 1 ADPO studienbegleitende Prüfungen, die bis zum dritten Fachsemester erstmals abgelegt und nicht bestanden sind, als nicht abgelegt.

(3) Bei länger währender Krankheit oder in anderen begründeten Fällen kann der Prüfungsausschuss im Einzelfall auf Antrag eine Übertragung von freien Prüfungsversuchen auf spätere Semester genehmigen.

§ 16 Bestehen und Bewertung der Bachelorprüfung

(1) Die Bachelorprüfung ist bestanden, wenn alle Fachprüfungen bestanden sind und die Bachelor's Thesis mindestens mit „ausreichend” (4,0) bewertet wurde.

(2) [1]Die Gesamtnote der Bachelorprüfung wird als gewichtetes Notenmittel der Fachprüfungen gemäß § 12 Abs. 1 und der Bachelor's Thesis errechnet. [2]Die Notengewichte der einzelnen Prüfungsleistungen entsprechen den zugeordneten Credits. [3]Dabei werden die in Anlage 1 als Grundlagenfächer gekennzeichneten Prüfungsfächer nur mit 50 % der Credits gewichtet.

Das Gesamturteil wird durch das Prädikat gemäß § 16 ADPO ausgedrückt.

§ 17 Zeugnis, Urkunde, Diploma Supplement

(1) Ist die Bachelorprüfung bestanden, so ist ein Zeugnis auszustellen, das die einzelnen Prüfungsfächer und die in diesen Fächern erzielten Noten, das Thema und die Note der Bachelor's Thesis sowie die Gesamtnote enthält.

(2) [1]Mit dem Zeugnis wird eine Urkunde ausgehändigt, in der die Verleihung des akademischen Grades „Bachelor of Science” (B.Sc.) beurkundet wird. [2]Die Bachelorurkunde wird vom Präsidenten der Technischen Universität München unterzeichnet, das Zeugnis vom Vorsitzenden des Prüfungsausschusses oder dessen Stellvertreter. [3]Als Datum des Zeugnisses ist der Tag anzugeben, an dem alle Prüfungsleistungen erfüllt sind.

(3) Außerdem wird ein englischsprachiges Diploma Supplement ausgehändigt.

III. SCHLUSSBESTIMMUNG

§ 18 In-Kraft-Treten

(1) [1]Diese Satzung tritt mit Wirkung vom 1. Oktober 2005 in Kraft. [2]Sie gilt für alle Studenten, die ab dem Wintersemester 2005/06 ihr Fachstudium an der Technischen Universität München aufnehmen.

(2) Gleichzeitig tritt die Fachprüfungsordnung für den Bachelor-Studiengang Informatik an der Technischen Universität München vom 3. November 2000 (KWMBl II 2001 S. 454), zuletzt geändert durch Satzung vom 29. September 2003 (KWMBl II 2004 S. 1451), vorbehaltlich der Regelung in Abs. 1 außer Kraft.

Anlage 1: Prüfungsfächer

A Modulkatalog Pflichtfächer Informatik

B Modulkatalog Wahlfächer Informatik und Zusatzfächer

C Modulkatalog Mathematik

D Modulkatalog Überfachliche Grundlagen und Zusatzfächer

E Modulkatalog Anwendungsfach

[1]Die zu belegenden Module in den einzelnen Anwendungsfächern sind in Anlage 2 aufgelistet. [2]Von den 31 Credits für ein Anwendungsfach entfallen mindestens 20 Credits auf Pflichtfächer, die übrigen Credits sind je nach Anwendungsfach in Wahlpflicht- oder Wahlfächern zu erbringen.

F Anmerkungen

- [1]Die in den Modulkatalogen angegebene Prüfungsdauer gilt nur für schriftliche Prüfungen. [2]Im Falle einer Midterm-Klausur erhöht sich die maximal zulässige Prüfungsdauer um 25 %.
- [1]Seminare und Praktika werden in der Regel nicht in Form einer Klausur geprüft (gekennzeichnet durch ***). [2]Prüfungsleistungen sind hier beispielsweise Vorträge und Ausarbeitungen. [3]Werden Seminare und Praktika zusätzlich schriftlich geprüft, liegt die Prüfungsdauer zwischen Credits*15 Minuten und Credits*25 Minuten unter Berücksichtigung von § 6 Abs. 2.
- Die Liste der Wahlfächer (gekennnzeichnet durch W) in den Modulkatalogen Informatik und Überfachliche Grundlagen wird vom Prüfungsausschuss laufend fortgeschrieben und aktualisiert.

G Erklärung der Fächerkennzeichnungen (FKZ)

P:	Pflichtfach
WP:	Wahlpflichtfach
W:	Wahlfach
G:	Grundlagenfach; die zugehörige Fachprüfung muss spätestens im zweiten Fachsemester zum ersten Mal abgelegt werden.
F:	Fächer mit freiem Prüfungsversuch gemäß § 15 Abs. 1

Anlage 2: Anwendungsfächer

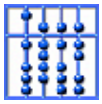
Ausgefertigt aufgrund des Senatsbeschlusses der Technischen Universität München vom 16. Februar 2005 sowie der Genehmigung des Bayerischen Staatsministeriums für Wissenschaft, Forschung und Kunst vom 6. Oktober 2005 Nr. X/4-3/41b16-10b/12 360.

München, den 24. Oktober 2005
Technischen Universität München

Wolfgang A. Herrmann
Präsident

Diese Satzung wurde am 24. Oktober 2005 in der Hochschule niedergelegt; die Niederlegung wurde am 24. Oktober 2005 durch Anschlag in der Hochschule bekannt gemacht. Tag der Bekanntmachung ist daher der 24. Oktober 2005.

Claudia Gold, Christian Herzog, 31.10.2005



Fakultät für Informatik
der Technischen Universität München

Prüfungsausschuss der Fakultät für Informatik

- Home
- Aktuelle Mitteilungen
- Studienberatung
- Studiensekretariat
- Bachelor-Studiengang Informatik
  - Eignungsfeststellung
  - Prüfungsordnung
  - Anwendungsfach
  - Bachelor-Arbeit
  - FAQ
- Alle Studiengänge
- Anerkennungen
- Eignungsfeststellung
- Prüfungsrücktritt
- Vertrauensärzte
- Bafög

Bachelor-Studiengang Informatik

Fachprüfungsordnung von 2005

Modulkatalog Pflichtfächer Informatik (Anlage 1A)

Dieser Modulkatalog enthält die Pflichtfächer aus dem Bereich Informatik im Umfang von 83 Credits.

ID	Veranstaltung	Turnus	FKZ	Umfang	Credits	Prüfungsdauer
IN0001	Einführung in die Informatik 1	WS	P/G/F	4V	6	90 - 150 min
IN0002	Praktikum: Grundlagen der Programmierung	WS,SS	P	1Ü + 3P	6	90 - 150 min
IN0003	Einführung in die Informatik 2	SS	P/G	2V + 1Ü	4	60 - 100 min
IN0004	Einführung in die Technische Informatik	WS	P/G/F	4V + 2Ü	8	120 - 200 min
IN0005	Praktikum: Technische Informatik	WS,SS	P	4P	8	120 - 200 min
IN0006	Einführung in die Softwaretechnik	SS	P/G	3V + 1Ü	5	75 - 125 min
IN0007	Grundlagen: Algorithmen und Datenstrukturen	SS	P/G	3V + 1Ü	5	75 - 125 min
IN0008	Grundlagen: Datenbanken	WS	P	3V + 1Ü	5	75 - 125 min
IN0009	Grundlagen: Betriebssysteme und Systemsoftware	WS	P	3V + 1Ü	5	75 - 125 min
IN0010	Grundlagen: Rechnernetze und Verteilte Systeme	SS	P	3V + 1Ü	5	75 - 125 min
IN0011	Einführung in die Theoretische Informatik	SS	P	4V + 2Ü	8	120 - 200 min
IN0012	Projekt: Systementwicklung	WS,SS	P	6P	10	***

IN0013	Proseminar	WS,SS	P	2S	4	***
IN0014	Seminar	WS,SS	P	2S	4	***

Anmerkungen:

- Die in den Modulkatalogen angegebene Prüfungsdauer gilt nur für schriftliche Prüfungen. Im Falle einer Midterm-Klausur erhöht sich die maximal zulässige Prüfungsdauer um 25%.
- Seminare und Praktika werden in der Regel nicht in Form einer Klausur geprüft (gekennzeichnet durch ***). Prüfungsleistungen sind hier beispielsweise Vorträge und Ausarbeitungen. Werden Seminare und Praktika zusätzlich schriftlich geprüft, liegt die Prüfungsdauer zwischen Credits*15 Minuten und Credits*25 Minuten unter Berücksichtigung von § 6 Abs. 2.

Erklärung der Fächerkennzeichnungen (FKZ):

P: Pflichtfach

WP: Wahlpflichtfach

W: Wahlfach

G: Grundlagenfach; die zugehörige Fachprüfung muss spätestens im zweiten Fachsemester zum ersten Mal abgelegt werden.

F: Fächer mit freiem Prüfungsversuch gemäß § 15 Abs. 1

---

generiert am 21.10.2006, Christian Herzog

Fakultät für Informatik
der Technischen Universität München

Prüfungsausschuss der Fakultät für Informatik

- Home
- Aktuelle Mitteilungen
- Studienberatung
- Studiensekretariat
- Bachelor-Studiengang Informatik
  - Eignungsfeststellung
  - Prüfungsordnung
  - Anwendungsfach
  - Bachelor-Arbeit
  - FAQ
- Alle Studiengänge
- Anerkennungen
- Eignungsfeststellung
- Prüfungsrücktritt
- Vertrauensärzte
- Bafög

Bachelor-Studiengang Informatik

Fachprüfungsordnung von 2005

Modulkatalog Wahlfächer Informatik und Zusatzfächer (Anlage 1B)

Aus diesem Katalog sind für die Bachelorprüfung Module im Umfang von 10 Credits zu wählen. Module, die über diese 10 Credits hinaus belegt werden, zählen als Zusatzfächer (vgl. § 14).

ID	Veranstaltung	Turnus	FKZ	Umfang	Credits	Prüfungsdauer
Wahlfächer aus dem Fachgebiet "Software Engineering" (SE):						
IN2114	Automotive Software - Methoden und Technologien	SS	W	2V + 1Ü	4	60 - 100 min
IN2078	Grundlagen der Programm- und Systementwicklung	WS	W	3V + 2Ü	6	90 - 150 min
IN2079	IT und Unternehmensberatung	WS	W	2V	3	45 - 75 min
IN2080	Modellierung verteilter Systeme	SS	W	2V + 1Ü	4	60 - 100 min
IN2081	Muster in der Softwaretechnik	SS	W	2V + 2Ü	5	75 - 125 min
IN2082	Projektmanagement	SS	W	2V + 2Ü	5	75 - 125 min
IN2083	Projektorganisation und -management in der Softwaretechnik	WS	W	2V + 2Ü	5	75 - 125 min
IN2084	Prozesse und Methoden beim Testen von Software	SS	W	2V	3	45 - 75 min
IN2088	Softwarearchitekturen	WS	W	2V	3	45 - 75 min
IN2087	Software Engineering für betriebliche Anwendungen - Masterkurs	SS	W	2V + 2Ü	5	75 - 125 min
IN2126	Software Engineering I: Softwaretechnik (vertieft)	WS	W	3V + 2Ü	6	90 - 150 min
IN2086	Software Technologien für Kommunikationsnetze	WS	W	2V	3	45 - 75 min
IN2089	Strategisches IT-Management	azyklisch	W	2V	3	45 - 75 min
IN2090	Verteiltes Problemlösen	SS	W	2V	3	45 - 75 min
Wahlfächer aus dem Fachgebiet "Datenbanken und Informationssysteme" (DBI):						
IN2027	Betriebliche Informationssysteme und ihre Rolle im Unternehmen	unregelmäßig	W	2V	3	45 - 75 min
IN2028	Business Analytics	WS	W	2V + 2Ü	5	75 - 125 min
IN2030	Data Mining und Knowledge Discovery	WS	W	2V	3	45 - 75 min
IN2118	Datenbanksysteme und moderne CPU-Architekturen	unregelmäßig	W	3V + 2Ü	6	90 - 150 min
IN2031	Einsatz und Realisierung von Datenbanksystemen	SS	W	3V + 2Ü	6	90 - 150 min

IN2032	Elektronisches Publizieren / Document Engineering und das World-Wide Web	WS,SS	W	3V + 1Ü	5	75 - 125 min
IN2033	Informationsmanagement	SS	W	2V + 2Ü	5	75 - 125 min
IN2034	Internetbasierte Geschäftssysteme	WS	W	2V	3	45 - 75 min
IN2035	Management des Softwarelebenszyklus und des organisatorischen Wandels	WS	W	3V	4	60 - 100 min
IN2036	Multimedia-Datenbanken und Content-Management	SS	W	2V	3	45 - 75 min
IN2037	Referenzmodelle für Informationssysteme	SS	W	3V + 1Ü	5	75 - 125 min
IN2038	Verteilte und Web-Informationssysteme	unregelmäßig	W	3V + 2Ü	6	90 - 150 min
IN2039	XML-Informationssysteme	unregelmäßig	W	3V + 2Ü	6	90 - 150 min
Wahlfächer aus dem Fachgebiet "Künstliche Intelligenz und Robotik" (KIR):						
IN2057	3D Computer Vision	WS	W	3V + 2Ü	6	90 - 150 min
IN2123	3D Computer Vision II	SS	W	2V + 2Ü	5	75 - 125 min
IN2058	Anwendungen Wissenbasierter Methoden	SS	W	3V + 1Ü	5	75 - 125 min
IN2060	Echtzeitsysteme	WS	W	3V + 2Ü	6	90 - 150 min
IN2061	Einführung in die digitale Signalverarbeitung	SS	W	3V + 3Ü	7	115 - 175 min
IN2062	Grundlagen der Küstlichen Intelligenz	WS	W	3V + 1Ü	5	75 - 125 min
IN2124	Grundlegende Mathematische Methoden für Imaging und Visualisierung	WS	W	2V + 2Ü	5	75 - 125 min
IN2063	Integrierte (embedded) Intelligente Systeme	SS	W	3V + 1Ü	5	75 - 125 min
IN2064	Maschinelles Lernen	WS	W	3V	4	60 - 100 min
IN2065	Maschinelles Lernen II	SS	W	3V	4	60 - 100 min
IN2066	Neuronale Netze	SS	W	3V	4	60 - 100 min
IN2067	Robotik	WS	W	3V + 2Ü	6	90 - 150 min
IN2068	Sensorgeführte Roboter	WS,SS	W	1V	1	45 min
IN2069	Statistische und neuronale Lernverfahren	WS	W	2V	3	45 - 75 min
IN2070	Verteilte Künstliche Intelligenz	SS	W	2V	3	45 - 75 min
IN2071	Wissensbasierte Systeme für industrielle Anwendungen	SS	W	3V	4	60 - 100 min
Wahlfächer aus dem Fachgebiet "Computergrafik und Bildverstehen" (CGBV):						
IN2014	Auswertung von Bildfolgen	SS	W	2V	3	45 - 75 min
IN2015	Bildsynthese	WS	W	3V	4	60 - 100 min
IN2016	Bildverstehen	WS	W	3V	4	60 - 100 min
IN2017	Computer Grafik	SS	W	4V	6	90 - 150 min
IN2112	Desktop-basierte Nutzerschnittstellen für Computeranwendungen	SS	W	2V	3	45 - 75 min
IN2111	Dreidimensionale Nutzerschnittstellen	WS	W	2V	3	45 - 75 min
IN2018	Erweiterte Realität	WS	W	2V	3	45 - 75 min
IN2019	Erweiterte Realität II	SS	W	2V	3	45 - 75 min

IN2020	Geometrieverarbeitung	SS	W	3V	4	60 - 100 min
IN2021	Informatikanwendungen in der Medizin	WS	W	4V	6	90 - 150 min
IN2022	Informatikanwendungen in der Medizin II	SS	W	2V + 2Ü	5	75 - 125 min
IN2116	Informationsdarstellungsgeräte	nur SS 06	W	2V	3	45 - 75 min
IN2023	Methoden der industriellen Bildverarbeitung	SS	W	2V	3	45 - 75 min
IN2024	Modellbasierte Auswertung von Bildern und Bildfolgen	SS	W	2V	3	45 - 75 min
IN2025	Simulation und Animation	WS	W	3V	4	60 - 100 min
IN2026	Wissenschaftliche Visualisierung	WS	W	3V	4	60 - 100 min
Wahlfächer aus dem Fachgebiet "Rechnerarchitektur" (RA):						
IN2072	Analyse von Systemperformanz	SS	W	4V	6	90 - 150 min
IN2073	Grid Computing	SS	W	2V	3	45 - 75 min
IN2074	Hochleistungsarchitekturen	SS	W	2V + 1Ü	4	60 - 100 min
IN2075	Rechensysteme in Einzeldarstellungen: Mikroprozessoren	SS	W	2V + 1Ü	4	60 - 100 min
IN2076	Rechnerarchitektur	WS	W	3V	4	60 - 100 min
IN2077	Skalierbare Parallelrechner mit gemeinsamem Speicher	WS	W	2V + 1Ü	4	60 - 100 min
IN2125	Virtualisierungstechniken	WS	W	2V + 1Ü	4	60 - 100 min
Wahlfächer aus dem Fachgebiet "Verteilte Systeme und Rechnernetze" (VRN):						
IN2091	Automatisiertes Management von vernetzten, nebenläufigen Systemen	SS	W	3V	4	60 - 100 min
IN2092	Computergestützte Gruppenarbeit	WS	W	2V	3	45 - 75 min
IN2119	Computergestützte Gruppenarbeit 2	WS	W	2V	3	45 - 75 min
IN2093	eLearning - Techniken und Infrastrukturen	WS	W	2V	3	45 - 75 min
IN2094	Fehlertoleranz	SS	W	2V + 1Ü	4	60 - 100 min
IN2095	Internetprotokolle	SS	W	3V + 1Ü	5	75 - 125 min
IN2096	Komponenten zum Aufbau von Rechnernetzen	unregelmäßig	W	3V	4	60 - 100 min
IN2097	Masterkurs Rechnernetze	WS	W	3V + 1Ü	5	75 - 125 min
IN2098	Mobile verteilte Systeme	SS	W	3V + 1Ü	5	75 - 125 min
IN2099	Netz- und Systemmanagement	SS	W	3V	4	60 - 100 min
IN2100	Sichere Rechensysteme	WS	W	3V + 1Ü	5	75 - 125 min
IN2101	Sicherheit vernetzter Systeme	SS	W	2V	3	45 - 75 min
IN2102	Verteilte Anwendungen	SS	W	3V	4	60 - 100 min
Wahlfächer aus dem Fachgebiet "Formale Methoden und ihre Anwendungen" (FMA):						
IN2040	Abstrakte Maschinen	SS	W	2V + 2Ü	5	75 - 125 min
IN2041	Automaten und formale Sprachen	WS	W	4V + 2Ü	8	120 - 200 min
IN2042	Automaten und formale Sprachen II	unregelmäßig	W	2V	3	45 - 75 min
IN2043	Automatisches Beweisen	unregelmäßig	W	2V	3	45 - 75 min
IN2044	Compilerbau	SS	W	4V + 2Ü	8	120 - 200 min

IN2045	Diskrete Simulation	SS	W	3V	4	60 - 100 min
IN2046	Entscheidbarkeit und Berechenbarkeit	unregelmäßig	W	2V	3	45 - 75 min
IN2047	Formale Methoden der Computersicherheit	WS	W	2V	3	45 - 75 min
IN2048	Gleichungslogik und Lambda-Kalkül	WS	W	4V + 2Ü	8	120 - 200 min
IN2049	Logik	WS	W	4V + 2Ü	8	120 - 200 min
IN2050	Model Checking	SS	W	4V + 2Ü	8	120 - 200 min
IN2051	Nichtsequentielle Systeme und nebenläufige Prozesse	SS	W	4V + 2Ü	8	120 - 200 min
IN2052	Petrinetze	SS	W	2V	3	45 - 75 min
IN2113	Programmiersprachen	unregelmäßig	W	2V + 2Ü	5	75 - 125 min
IN2053	Programmoptimierung	WS	W	4V + 2Ü	8	120 - 200 min
IN2054	Quantitative Modelle für Rechensysteme	SS	W	3V	4	60 - 100 min
IN2055	Semantik	SS	W	4V + 2Ü	8	120 - 200 min
IN2056	Temporale Logik	Unregelmäßig	W	2V	3	45 - 75 min
Wahlfächer aus dem Fachgebiet "Algorithmen und Wissenschaftliches Rechnen" (AWR):						
IN2001	Algorithmen des Wissenschaftlichen Rechnens	SS	W	4V + 2Ü	8	120 - 200 min
IN2002	Algorithmen des Wissenschaftlichen Rechnens II	unregelmäßig	W	2V + 1Ü	4	60 - 100 min
IN2127	Ausgewählte Themen in algorithmischer Bioinformatik	SS	W	3V + 1Ü	5	75 - 125 min
IN2003	Effiziente Algorithmen und Datenstrukturen	WS	W	4V + 2Ü	8	120 - 200 min
IN2004	Effiziente Algorithmen und Datenstrukturen II	SS	W	4V + 2Ü	8	120 - 200 min
IN2005	Einführung in das Wissenschaftliche Rechnen	WS	W	2V	3	45 - 75 min
IN2006	Internet-Algorithmik	SS	W	4V + 2Ü	8	120 - 200 min
IN2007	Komplexitätstheorie	WS	W	4V + 2Ü	8	120 - 200 min
IN2008	Kryptologie	WS	W	2V	3	45 - 75 min
IN2009	Kryptologie II	SS	W	2V	3	45 - 75 min
IN2010	Modellbildung und Simulation	SS	W	4V + 2Ü	8	120 - 200 min
IN2115	Netzwerkalgorithmen	WS	W	4V + 2Ü	8	120 - 200 min
IN2011	Parallele Algorithmen	WS	W	4V + 2Ü	8	120 - 200 min
IN2012	Parallele Numerik	WS	W	2V + 2Ü	5	75 - 125 min
IN2013	Parallele Programmierung und Hochleistungsrechnen	SS	W	2V + 1Ü	4	60 - 100 min

Anmerkungen:

- Die in den Modulkatalogen angegebene Prüfungsdauer gilt nur für schriftliche Prüfungen. Im Falle einer Midterm-Klausur erhöht sich die maximal zulässige Prüfungsdauer um 25%.
- Seminare und Praktika werden in der Regel nicht in Form einer Klausur geprüft (gekennzeichnet durch ***). Prüfungsleistungen sind hier beispielsweise Vorträge und Ausarbeitungen. Werden Seminare und Praktika zusätzlich schriftlich geprüft, liegt die Prüfungsdauer zwischen Credits*15 Minuten und Credits*25 Minuten unter Berücksichtigung von § 6 Abs. 2.
- Die Liste der Wahlfächer (gekennzeichnet durch W) in den Modulkatalogen Informatik und Überfachliche Grundlagen wird vom Prüfungsausschuss laufend fortgeschrieben und aktualisiert

Erklärung der Fächerkennzeichnungen (FKZ):

P: Pflichtfach

WP: Wahlpflichtfach

W: Wahlfach

G: Grundlagenfach; die zugehörige Fachprüfung muss spätestens im zweiten Fachsemester zum ersten Mal abgelegt werden.

F: Fächer mit freiem Prüfungsversuch gemäß § 15 Abs. 1

---

generiert am 21.10.2006, Christian Herzog

Fakultät für Informatik
der Technischen Universität München

Prüfungsausschuss der Fakultät für Informatik

- Home
- Aktuelle Mitteilungen
- Studienberatung
- Studiensekretariat
- Bachelor-Studiengang Informatik
  - Eignungsfeststellung
  - Prüfungsordnung
  - Anwendungsfach
  - Bachelor-Arbeit
  - FAQ
- Alle Studiengänge
- Anerkennungen
- Eignungsfeststellung
- Prüfungsrücktritt
- Vertrauensärzte
- Bafög

Bachelor-Studiengang Informatik

Fachprüfungsordnung von 2005

Modulkatalog Mathematik (Anlage 1C)

Dieser Modulkatalog enthält die Pflichtfächer aus dem Bereich Mathematik im Umfang von 35 Credits.

ID	Veranstaltung	Turnus	FKZ	Umfang	Credits	Prüfungsdauer
IN0015	Diskrete Strukturen	WS	P/G/F	4V + 2Ü	8	120 - 200 min
MA0901	Lineare Algebra für Informatiker	SS	P/G	4V + 2Ü	8	120 - 200 min
MA0902	Analysis für Informatiker	WS	P	4V + 2Ü	8	120 - 200 min
IN0018	Diskrete Wahrscheinlichkeitstheorie	SS	P	3V + 1Ü	5	75 - 125 min
IN0019	Numerisches Programmieren	WS	P	2V + 3Ü	6	90 - 150 min

Anmerkung:

- Die in den Modulkatalogen angegebene Prüfungsdauer gilt nur für schriftliche Prüfungen. Im Falle einer Midterm-Klausur erhöht sich die maximal zulässige Prüfungsdauer um 25%.

Erklärung der Fächerkennzeichnungen (FKZ):

P: Pflichtfach
WP: Wahlpflichtfach
W: Wahlfach
G: Grundlagenfach; die zugehörige Fachprüfung muss spätestens im zweiten Fachsemester zum ersten Mal abgelegt werden.
F: Fächer mit freiem Prüfungsversuch gemäß § 15 Abs. 1

generiert am 21.10.2006, Christian Herzog

Fakultät für Informatik
der Technischen Universität München

Prüfungsausschuss der Fakultät für Informatik

- Home
- Aktuelle Mitteilungen
- Studienberatung
- Studiensekretariat
- Bachelor-Studiengang Informatik
  - Eignungsfeststellung
  - Prüfungsordnung
  - Anwendungsfach
  - Bachelor-Arbeit
  - FAQ
- Alle Studiengänge
- Anerkennungen
- Eignungsfeststellung
- Prüfungsrücktritt
- Vertrauensärzte
- Bafög

Bachelor-Studiengang Informatik

Fachprüfungsordnung von 2005

Modulkatalog Überfachliche Grundlagen und Zusatzfächer (Anlage 1D)

Zu wählen sind Module im Umfang von 6 Credits aus dem folgenden Wahlkatalog. Module, die über diese 6 Credits hinaus belegt werden, zählen als Zusatzfächer (vgl. § 14).

ID	Veranstaltung	Turnus	FKZ	Umfang	Credits	Prüfungsdauer
IN9001	Kommunikation und Team	WS,SS	W	2S	4	***
IN9002	Public Relations	WS,SS	W	2V	3	45 - 75 min
IN9003	Informatikrecht	SS	W	2V	3	45 - 75 min
IN9004	Ethik und Verantwortung	unregelmäßig	W	2S	4	***
IN9013	Geschäftsidee und Markt: Businessplan-Grundlagenseminar	WS	W	2S	4	***
IN9005	Geschäftsmodell, Vertrieb und Finanzen: Businessplan-Aufbauseminar	SS	W	2S	4	***
IN9006	Gründung und Führung kleiner softwareorientierter Unternehmen	SS	W	1S	2	***
IN9007	SET-Tutor	WS	W	1S	2	***
IN9008	Innovation und Risiko	SS	W	2S	4	***
IN9009	Sich und andere führen	SS	W	2S	4	***
IN9010	Seminar Wissenschaftler und Ethik	unregelmäßig	W	2S	4	***
IN9011	Seminar Projektmanagement	WS	W	2S	4	***
IN9014	Medizin, Körper und Konsum	SS	W	2Ü	2	***
IN9015	Informationskompetenz	SS	W	2V	3	45 - 75 min
IN9016	Didaktische Grundlagen für Informatiker: Konzepte der Didaktik und Psychologie für Bildungsmaßnahmen und E-Learning Systeme	SS	W	2V	3	45 - 75 min
IN9017	Existenzgründung	WS, SS	W	2S	4	***
IN9018	Technik, Wirtschaft, Gesellschaft	WS, SS	W	2V	3	***
IN9019	Unternehmerische Medienkompetenz	WS,SS	W	2V	3	***
IN9020	Einführung in das Wirtschaftsprivatrecht	WS	W	2V	3	***
IN9021	Grundzüge der Volkswirtschaftslehre	WS	W	2V	3	***
IN9022	Geschichte der Informatik 1: Computing before Computers	WS	W	2V	3	***
IN9023	Geschichte der Informatik 2: Von der Erfindung des Computers zum Internet	WS	W	2V	3	***
IN9024	Die Entwicklung der modernen Informationsgesellschaft	WS	W	2S	4	***

Anmerkungen:

- Die in den Modulkatalogen angegebene Prüfungsdauer gilt nur für schriftliche Prüfungen. Im Falle einer Midterm-Klausur erhöht sich die maximal zulässige Prüfungsdauer um 25%.
- Seminare und Praktika werden in der Regel nicht in Form einer Klausur geprüft (gekennzeichnet durch ***). Prüfungsleistungen sind hier beispielsweise Vorträge und Ausarbeitungen. Werden Seminare und Praktika zusätzlich schriftlich geprüft, liegt die Prüfungsdauer zwischen Credits*15 Minuten und Credits*25 Minuten unter Berücksichtigung von § 6 Abs. 2.
- Die Liste der Wahlfächer (gekennzeichnet durch W) in den Modulkatalogen Informatik und Überfachliche Grundlagen wird vom Prüfungsausschuss laufend fortgeschrieben und aktualisiert

Erklärung der Fächerkennzeichnungen (FKZ):

P:	Pflichtfach
WP:	Wahlpflichtfach
W:	Wahlfach
G:	Grundlagenfach; die zugehörige Fachprüfung muss spätestens im zweiten Fachsemester zum ersten Mal abgelegt werden.
F:	Fächer mit freiem Prüfungsversuch gemäß § 15 Abs. 1

---

generiert am 21.10.2006, Christian Herzog