Concept: SE with AI (Jack Heseltine, concept document)

Content:

* SE note (skills)
* AI curricula and content (core content and theoretical)
* Symbolic Computation course list (long and deep direction)
* note

The skills focus (and basis) for this self-designed curriculum in the JKUniverse (Johannes Kepler Universität (JKU) in Linz and Hagenberg Campus, Fachhochschule Upper Austria, possibly in the future: [Institute of Digital Sciences Austria](https://www.bmbwf.gv.at/en/Topics/Higher-education---universities/Institute-of-Digital-Sciences-Austria-(IDSA)-%E2%80%93-A-new%2C-innovative-University-of-Technology-for-Digitalisation-and-Digital-Transformation-in-Austria.html)) is the Hagenberg Software Engineering (SE) curriculum, see [Software Engineering Studies Documentation and Research-Projects/Interests - heseltime](https://heseltime.github.io/rDse) with a personal focus of Software for Math (as thesis project), which makes sense especially in this context.

Overlapping or building on SE, at JKU, the AI curriculum to Masters level is:

A close-up of a document

Description automatically generated with low confidence

SE should be finished as far as possible to start on this track (symbolic and math version, in terms of JKU tracks). Theoretical and especially math background needs building in the SE phase as well as on an as-needed basis in the Masters:

A picture containing text, diagram, parallel, font

Description automatically generated

The Masters surely similarly follows a dependency graph, both within itself but also wrt the Bachelor’s.

In terms of content, both levels of the JKU degree are distributed as follows.

A picture containing text, screenshot, circle, diagram

Description automatically generated

This was the Masters, where the focus is on independent work and the area of specialization – core content in the Bachelors:

A picture containing text, diagram, screenshot, circle

Description automatically generated

For an AI-SE integration, a Masters Thesis that builds on the SE thesis is preferable, where the JKU-track (symbolic/math) should be respected.

The motivation for this studies should lie in the content and its application themselves, but also a view to the following doctoral level studies at RISC (or another institute) but where this institute is certainly notable within the JKUniverse, see the following PhD overview and course list for Symbolic Computation to wrap.

A picture containing text, font, diagram, screenshot

Description automatically generated

The following is copied from [Studying Symbolic Computation - RISC - Johannes Kepler University (jku.at)](https://risc.jku.at/studying-symbolic-computation/) – without course descriptions, my italics.

RISC offers three kinds of courses:

**Curriculum Courses**

*These are the core courses of the curriculum; every Ph.D. student performs course work in the amount of at least 30 ECTS points. The curriculum courses are offered annually or biennially; additionally Special Topics courses with varying content may be offered.*

**Seminars**

In seminars, the RISC faculty discusses the state of the art in research and ongoing project work. In the first year, every Ph.D. student attends (as a listener) several seminars; later the student attends (as an active participant) *at least one seminar per semester.* Seminar topics vary from semester to semester, the ones listed below are just examples.

**Standard Courses**

These are courses offered by RISC for *bachelor/master* students in mathematics or computer science. They are actually not part of the RISC curriculum *but they may help in exceptional cases to supplement the education of new Ph.D. students*.

According to the general working areas of RISC, the courses are structured into three categories. Short course descriptions are given in the appendix.

Computer **Algebra**

Curriculum Courses

|  |  |  |
| --- | --- | --- |
| **Title** | **ECTS** | **Period** |
| Computer Algebra | 4.5 | annually |
| Gröbner Bases | 3 | biennially |
| Fast Arithmetic and Factorization | 3 | biennially |
| Elimination Theory | 3 | biennially |
| Commutative Algebra and Algebraic Geometry | 7.5 | biennially |
| Algorithmic Algebraic Geometry | 3 | biennially |
| Computer Analysis | 3 | biennially |
| Symbolic Linear Algebra | 3 | biennially |
| Computer Algebra for Concrete Mathematics | 4.5 | annually |
| Algorithmic Combinatorics | 4.5 | annually |
| Symbolic Summation and Special Functions 1 | 3 | biennially |
| Symbolic Summation and Special Functions 2 | 3 | biennially |
| Category Theory for Symbolic Computation | 3 | biennially |
| Algebraic Methods in Kinematics | 3 | biennially |
| Algebraic Topology | 3 | biennially |
| Homological Algebra | 3 | biennially |
| Special Topics in Computer Algebra | 3 | on demand |

Seminars (Examples)

* Seminar “Computer Algebra”
* Seminar “Algorithmic Algebra”
* Seminar “Algorithmic Combinatorics
* Seminar “Algebraic Geometry”
* Seminar “Symbolic Summation for Particle Physics”
* Seminar “Symbolic Summation and Special Functions”
* Seminar “Fundamentals of Numerical Analysis and Symbolic Computation”

Standard Courses

* Algebra
* Analysis
* Linear Algebra 1 and 2
* Linear Algebra for Physicists 1 and 2
* Linear Algebra and Analytic Combinatorics 1 and 2
* Linear Algebra and Analytic Geometry 1 and 2
* Ordinary Differential Equations and Dynamical Systems 1
* Algebraic and Discrete Methods for Biology
* Preparatory Course Mathematics for Beginners of Business Informatics

Computational **Logic**

Curriculum Courses

|  |  |  |
| --- | --- | --- |
| **Title** | **ECTS** | **Period** |
| Thinking, Speaking, Writing 1+2 | 6 | annually |
| Mathematical Logic | 7.5 | annually |
| Topics in Mathematical Logic | 3 | biennially |
| Automated Reasoning | 3 | biennially |
| Advanced Automated Reasoning | 3 | biennially |
| Computability Theory | 3 | biennially |
| Decidability and Complexity Classes | 3 | biennially |
| Decidable Logical Theories | 3 | biennially |
| Rewriting in Computer Science and Logic | 3 | biennially |
| Unification Theory | 3 | biennially |
| Gödel’s Incompleteness Theorems | 3 | biennially |
| Formal Methods in Software Development | 4.5 | annually |
| Formal Semantics of Programming Languages | 3 | biennially |
| Formal Specification of Abstract Datatypes | 3 | biennially |
| Formal Models of Parallel a. Distributed Systems | 3 | biennially |
| Formal Languages and Formal Grammars 1 | 3 | biennially |
| Formal Languages and Formal Grammars 2 | 3 | biennially |
| Fixpoint Theory of Functional Programs | 3 | biennially |
| Special Topics in Computational Logic | 3 | on demand |

Seminars (Examples)

* Seminar “Automated Reasoning (Theorema)”
* Seminar “Set Theory and Logical Foundations”
* Seminar “Formal Methods”

Standard Courses

* Logic as a Working Language
* Logic for Computer Science
* Mathematics and Logic
* Computability and Complexity
* Formal Foundations in Business Informatics

Mathematical **Software**

Curriculum Courses

|  |  |  |
| --- | --- | --- |
| **Title** | **ECTS** | **Period** |
| Programming Project Symbolic Computation 1+2 | 6 | annually |
| Design and Analysis of Algorithms | 3 | biennially |
| Introduction to Parallel and Distributed Computing | 3 | biennially |
| Logic Programming | 3 | annually |
| Functional Programming | 3 | annually |
| Programming in Mathematica | 3 | annually |
| Chess Programming | 3 | biennially |
| Computer Algebra Systems | 3 | biennially |
| Automated Reasoning Systems | 3 | biennially |
| Computer-based Working Environments | 1.5 | annually |
| Special Topics in Mathematical Software | 3 | on demand |

Seminars (Examples)

* Bachelor Seminar “Selected Algorithms in Symbolic Computation”

Standard Courses

* Practical Software Technology
* Software Engineering
* Project Engineering
* Algorithms and Data Structures
* Information Systems
* Computer Systems
* Algorithmic Methods 1

As a personal side, I think this concept and PhD can accommodate varied backgrounds and interests, managing to strike a balance between applied skills and theoretical and math background, with flexibility to allow math content and proving later in one’s studies as well, which actually is of interest to lots of different people coming to this kind of studies.

I think the particular Masters combination allows for taking SE in an AI direction, with enough depth, and to integrate math and symbolic approaches as an application direction in the long and deep run.

Finally, the course, especially at PhD level and once a certain cruising altitude has been reached, might allow for math or computer science/software exchanges or intercalations externally from the JKUniverse/abroad as well (incl. summer schools). Accommodation for the JKUniverse content in Linz/Hagenberg mit make sense, at least some of the time. The distance learning mode for JKU AI studies and the part-time “berufsbgegleitend” mode for SE in Hagenberg are worth mentioning, for further flexibility, to the point of allowing work during studies: Software Engineering work would be the obvious choice.

But of course also: potential long/deep topics? For me: Symbolic/connectionist theme, LLMs in Theorema and similar applications, symbolic verification of LLM output, natural language to theorem to proof. To be determined.