Domain-Specific Languages of Mathematics
Course codes: DAT326 / DIT983

Patrik Jansson

2024-06-04

Contact	Patrik Jansson, 072 985 2033.
Results	Announced within 15 workdays
Exam check	2024-06-17, 12.15-12.45 in EDIT 6128
Aids	One textbook of your choice (Domain-Specific Languages of Mathematics, or Beta - Mathematics Handbook, or Rudin, or Adams and Essex, or ...). No printouts, no lecture notes, no notebooks, etc.
Grades	To pass you need **a minimum of 5p on each question (1 to 4)** and also reach these grade limits: 3: >=48p, 4: >=65p, 5: >=83p, max: 100p

Remember to write legibly. Good luck!

For reference: the learning outcomes. Some are tested by the hand-ins, some by the written exam.

- Knowledge and understanding
  - design and implement a DSL (Domain-Specific Language) for a new domain
  - organize areas of mathematics in DSL terms
  - explain main concepts of elementary real and complex analysis, algebra, and linear algebra
- Skills and abilities
  - develop adequate notation for mathematical concepts
  - perform calculational proofs
  - use power series for solving differential equations
  - use Laplace transforms for solving differential equations
- Judgement and approach
  - discuss and compare different software implementations of mathematical concepts

1. [25p] **Algebraic structure**: Boolean algebra (lightly edited from wikipedia):

   > A *Boolean algebra* is a six-tuple consisting of a set A, equipped with two binary operations $\wedge$ (called "meet"), $\vee$ (called "join"), a unary operation $\neg$ (called "not") and two elements 0 and 1 in A (called "bottom" and "top"), such that for all elements a of A, the following axioms hold: $\vee$ and $\wedge$ are associative, commutative and distribute over each other; $a \vee 0 = a$ and $a \wedge 1 = a$; $a \vee \neg a = 1$ and $a \wedge \neg a = 0$.

   (a) Define a type class *BoolAlg* that corresponds to the Boolean algebra structure.

   (b) Define a datatype *BA v* for the language of Boolean algebra expressions (with variables of type v) and define a *BoolAlg* instance for it. (These are expressions formed from applying the Boolean algebra operations to the appropriate number of arguments, e.g., all the left hand sides and right hand sides of the above equations.)

   (c) The datatype *Bool* is the simplest non-trivial Boolean algebra. The type **data** *SubAB* = *Empty* | *A* | *B* | *AB* can be used to represent all subsets of the two-element set $\{A, B\}$. Implement an instance declaration *BoolAlg SubAB* where meet is intersection and join is union.

   (d) Give a type signature for, and define, a general evaluator for *BA v* expressions on the basis of an assignment function.

   (e) Specialise the evaluator to the *SubAB* instance. Take three Boolean algebra expressions of type *BA String*, give the appropriate assignments and compute the results of evaluating the three expressions.

   Each question carries 5pts.

2. [25pts] Consider the following differential equation:

$$f'(t) = 2f(t) - f''(t), \quad f(0) = a, \quad f'(0) = b$$

   (a) [10pts] Solve the equation assuming that f can be expressed by a power series *fs*, that is, use *integ* and the differential equation to express the relation between *fs*, *fs*$'$, and *fs*$''$. What are the first three coefficients of *fs* (expressed in terms of a and b)?

   (b) [15pts] Solve the equation using the Laplace transform. You should need this formula (and the rules for linearity + derivative):

$$\mathcal{L}\,(\lambda t.\, e^{\alpha * t})\, s = 1/(s - \alpha)$$

   Show that your solution does indeed satisfy the three requirements.

3. [25p] **Proofs and homomorphisms:**

   That a function $h : A \to B$ is a homomorphism from $Op : A \to A \to A$ to $op : B \to B \to B$ can be abbreviated as:

   $$H_2(h, Op, op) = \forall\ x.\ \ \forall\ y.\ \ h\ (Op\ x\ y) == op\ (h\ x)\ (h\ y)$$

   (a) [5p] Prove $\exists\ op.\ \ H_2(odd, (+), op)$ where $odd :: \mathbb{Z} \to Bool$ checks if a number is odd.

   (b) [10p] Prove or disprove $\exists\ add.\ \ H_2(degree, (+), add)$ where $degree :: Poly\ \mathbb{R} \to Maybe\ \mathbb{N}$ computes *Just* the degree of a polynomial (or *Nothing* for the zero polyomial).

   (c) [10p] With $D : Fun \to Fun$ being the usual derivative operator for functions of one argument ($Fun = \mathbb{R} \to \mathbb{R}$), prove $\neg\ (\exists\ mul.\ \ H_2(D, (*), mul))$, where $(*)$ is pointwise multplication of functions.

   Motivate your steps and make sure to keep track of types, scope, etc. of your expressions.

4. [25p] **Typing maths:** Derivative of Inverse

   Consider the following (slightly edited) quote from [Adams, p167]:

   > **Derivatives of Inverse Functions**
   >
   > Suppose that the function f is differentiable on an interval (a, b) and that either $f'(x) > 0$ for $a < x < b$, so that f is increasing on (a, b), or $f'(x) < 0$ for $a < x < b$, so that f is decreasing on (a, b). In either case f is one-to-one on (a, b) and has an inverse, f^{-1} , defined by
   >
   > $$y = f^{-1}(x) \quad \iff \quad x = f(y), \quad (a < y < b).$$
   >
   > [... example graph skipped ...]
   >
   > Therefore, [... calculation skipped ...] and
   >
   > $$\frac{d}{dx} f^{-1}(x) = \frac{1}{f'(f^{-1}(x))}$$

   (a) [5p] Give types for f, a, b, x, y, f^{-1}, and f'. Make sure to account for the possibility that the types of x and y could be different.

   (b) [7p] Give (short, textual) names and types to d/dx and to the function $\cdot^{-1}$ that takes a function to its inverse.

   (c) [8p] Restate the final equation twice using your new names, first as a direct translation, then in point-free style (with no mention of x). You may use $recip\ x = 1\ /\ x$.

   (d) [5p] Consider the function $f(x) = 2 - x^2$ on the interval $(0, 1)$. Prove or disprove that the conditions from the quote are applicable to f.