Domain-Specific Languages of Mathematics
Course codes: DAT326 / DIT982

Patrik Jansson

2022-06-08

Contact	Patrik Jansson, 0729852033.
Results	Announced within 15 workdays
Exam check	2022-06-17, 12.15-13.00 in EDIT 6452
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** $SubAB = Empty \mid A \mid B \mid AB$ 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. [25p] **Laplace**

   Consider the following differential equation:

$$2 * f''\ t + 5 * f'\ t + 2 * f\ t = e^{-t}, \quad f\ 0 = \frac{1}{2}, \quad f'\ 0 = -\frac{1}{2}$$

   (a) [10p] 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′*, *fs″*, and *rhs*, where the *rhs* is the power series representation of the right-hand side. (Hint: $rhs = integ\ 1\ (scale\ (-1)\ rhs)$). What are the first three coefficients of *fs*? Explain how you compute them.

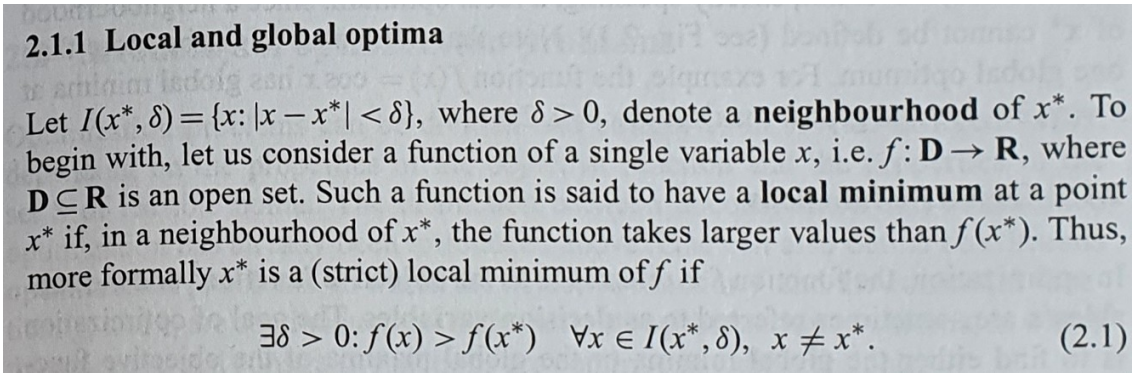
   (b) [15p] Solve the equation using the Laplace transform. You should need this formula (note that α can be a complex number) and the rules for linearity + derivative:

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

   Show that your solution does indeed satisfy the three requirements.

3. [25p] **Typing maths: Optima**

Consider the following mathematical text from M. Wahde [2008]:



2.1.1 Local and global optima

Let $I(x^*, \delta) = \{x: |x - x^*| < \delta\}$, where $\delta > 0$, denote a **neighbourhood** of x^*. To begin with, let us consider a function of a single variable x, i.e. $f: \mathbf{D} \to \mathbf{R}$, where $\mathbf{D} \subseteq \mathbf{R}$ is an open set. Such a function is said to have a **local minimum** at a point x^* if, in a neighbourhood of x^*, the function takes larger values than $f(x^*)$. Thus, more formally x^* is a (strict) local minimum of f if

$$\exists \delta > 0: f(x) > f(x^*) \quad \forall x \in I(x^*, \delta), \; x \neq x^*. \tag{2.1}$$

On the following page of Wahde [2008] the definition is generalised by letting $\mathbf{D}$ be an open set in an n-dimensional vectorspace and replacing the absolute value function used in I by the Euclidean norm (the length) of a vector:

$$norm(v) = \|v\| = \sqrt{\sum_{i=1}^{n} v_i^2}$$

(a) [7p] In the generalised case, give, and explain, the types of I, x, x^*, δ, f, and $norm$.

(b) [8p] Define a first-order logic predicate $LocMin$, with f and x^* as parameters, which captures Equation 2.1 (in the generalised case). Explain the scope of the different symbols (make sure no symbol is used before it is bound).

(c) [5p] Prove $LocMin(sin \circ norm, zero)$ where $zero = (0, 0)$.

(d) [5p] Prove $\neg LocMin(g, zero)$ where $g(x, y) = x * y$.

4. [25p] **Notation and equational reasoning**

Consider the following implementation of 2-by-2 matrix algebra + the determinant:

```
data M a = M a a a a deriving (Eq, Show)

diagM :: Additive a => a -> M a; diagM c = M c zero zero c
zeroM :: Additive a => M a;      zeroM = diagM zero
oneM  :: Ring a => M a;          oneM  = diagM one

addM :: Additive a => M a -> M a -> M a
addM (M a00 a01 a10 a11) (M b00 b01 b10 b11) = error "TODO (a)"

mulM :: Ring a => M a -> M a -> M a
mulM (M a00 a01 a10 a11) (M b00 b01 b10 b11) = error "TODO (b)"

instance Additive a => Additive (M a)  where zero = zeroM; (+) = addM
instance Ring a => Multiplicative (M a) where one  = oneM;  (*) = mulM

det :: Ring a => M a -> a
det (M a00 a01 a10 a11) = a00 * a11 - a01 * a10

detHomo :: (Eq a, Ring a) => M a -> M a -> Bool
detHomo m1 m2 = det (m1 * m2) == det m1 * det m2
```

(a) [5p] Implement $addM$.

(b) [8p] Implement $mulM$.

(c) [12p] Prove the homomorphism $\forall\ a, b.\ \ detHomo\ a\ b$ by equational reasoning.