Problem Statement and Goals
Drasil Matrix, Vector and Tensor Extension

Christopher W. Schankula

Table 1: Revision History

Date	Developer(s)	Change
January 19th, 2025	Christopher Schankula	Initial version of problem statement

1 Problem Statement

This section describes the problem that will be solved in this project. We start by presenting background information related to matrices, vectors and tensors. We then provide a high-level description of the problem, followed by an overview of the stakeholders and environment in which the resulting software will run.

1.1 Background

This section describes background on matrices, vectors, tensors, and the Drasil project.

1.1.1 Matrices, Vectors, and Tensors

A matrix is a rectangular table of numbers, symbols or expressions which are used to represent some mathematical object or concept. They are used in linear algebra, physics, and number theory, among other fields [4]. Meanwhile, a vector is a "quantities that cannot be expressed by a single number (a scalar)" [3]. A vector can be seen as a sort of one-dimensional matrix. Similarly to matrices, vectors are used in a wide variety of scientific fields.

A tensor is a type of mathematical objects that describe a relationship between matrices, vectors, or other tensors [5]. The relationship is a multilinear map, meaning it is a function of several linear inputs [2]. Tensors can be used to solve important physics problems, including in mechanics, electrodynamics, and general relativity [5].

1.2 Drasil Project

Drasil is a system that takes advantage of the inherent duplication in knowledge in a software engineering project (e.g. code, specification, testing, etc.). By capturing this knowledge once and generating these artifacts, they can be kept up-to-date and lead to more robust software [1].

1.3 Problem

Currently, vectors and matrices in Drasil are implemented in a way that does not promote the correctness of algorithms using them. For calculations on matrices, vectors and tensors to be correct, it is necessary to ensure certain properties about them, including the sizes of operands. Furthermore, current implementations of matrices and vectors in Drasil support only fixed sizes, and tensors are not yet implemented at all.

The problem being solved in this project will be to allow the specification of tensors, matrices, and vectors in the Drasil system and document and code generation associated with them.

1.4 Inputs and Outputs

The inputs to the system will be a representation of operations using tensors (matrices and vectors will be built on top of tensors). The outputs from the system will be documents describing the operations specified in the inputs and code in several supported languages to compute the operations described.

1.5 Stakeholders

This project has the following stakeholders:

- Dr. Carette and Dr. Smith: Project supervisors
- Researchers interested in software engineering
- Software engineers looking to develop safe and correct software

1.6 Environment

The software environment of this project is the Haskell language since Drasil is written in Haskell. Thus, it is a constraint that this project will be written in Haskell and will extend the existing Drasil project and software environment in which it is defined.

There is no particular hardware environment for this project. Any hardware able to support compiling Haskell code is considered in-scope.

2 Goals

This project consists of five main goals and two stretch goals. The main goals define what the project must achieve to be considered a success. Stretch goals represent additional future goals worth working towards if time/resources allow (and whose considerations are worth keeping in mind as the main system is built), but which are considered out of scope for the current project. The main goals are as follows:

G1. Expand Drasil's expression language with new primitives for generalized tensors.

- G1.1 Define smart constructors (built on top of the generalized tensors) for the simple cases of vectors and matrices.

G2. Expand Drasil's type system to cover tensors, vectors and matrices.

G3. Enable documentation and code generation using these new primitives.

G4. Include enough information in the types to support at least code runtime checks of sizes on tensor/vector/matrix operations.

G5. Demonstrate the use of this system to generate documents and code for basic scientific computing tasks using vectors, matrices and/or tensors.

- G5.1. Ensure that existing Drasil examples continue to work despite these new changes.

3 Stretch Goals

The following are stretch goals considered out of scope for the current project but are goals worth working towards in the future:

SG1. Demonstrate the use of this system for basic image processing tasks (e.g. mean/median filters).

SG2. Enable type safety of operations at specification-time.

4 Challenge Level and Extras

The challenge level for this project is **Research**. Exact deliverables will be determined in conjunction with the instructor. The extra being proposed is to write a research paper describing the work and showing the examples. The paper is to be published on arXiv.org at a minimum, with hopes to publish in a journal and/or conference.

References

[1] J. Carette, S. Smith, D. Szymczak, J. Balaci, B. MacLachlan, M. Niazi, S. Crawford, D. Scime, D. Chen, A. Hunt, T.-Y. Wu, M. Bilal, and B. Bosman. Drasil, Feb. 2021. URL https://github.com/JacquesCarette/Drasil/tree/v0.1-alpha.

[2] Wikipedia contributors. Multilinear map, Jan. 2024. URL https://en.wikipedia.org/wiki/Multilinear_map.

[3] Wikipedia contributors. Vector (mathematics and physics), Oct. 2024. URL https://en.wikipedia.org/wiki/Vector_(mathematics_and_physics).

[4] Wikipedia contributors. Matrix (mathematics), Jan. 2025. URL https://en.wikipedia.org/wiki/Matrix_(mathematics).

[5] Wikipedia contributors. Tensor, Jan. 2025. URL https://en.wikipedia.org/wiki/Tensor.