Homework # 4
due February 22, 3:30PM

1 Programming

In this Homework, we continue our definition of vector spaces in ML:

1. First, we rewrite all our previous definitions to avoid using `if` by using pattern matching. This means that `scale`, `add` and `dot` need new definitions. (The function `magn` doesn't need to be changed since it is defined without `if` anyway.) When you do this, you will need a few explicit type qualifications to ensure that the functions operate on vectors, rather than integer lists.
2. Then we will define new functions: `project`, `projectAll` and `inSpan` (see below).

All the functions (the re-written and the new) should be placed in a single file `vector2.sml`.

2 Notation

Mathematically we write a vector using a boldface lowercase letter:

$$\mathbf{v} = (v_1, v_2, \ldots, v_n)$$

Then we can define our existing operations:

`scale(x,v)`

$$x\mathbf{v} = (xv_1, xv_2, \ldots, xv_n)$$

`add(v,w)`

$$\mathbf{v} + \mathbf{w} = (v_1 + w_1, v_2 + w_2, \ldots, v_n + w_n, w_{n+1}, \ldots, w_m)$$

if m, the length of $\mathbf{w}$ is greater than n, the length of $\mathbf{v}$. If $m = n$, we end at the last common sum. If $m < n$, then $\mathbf{v}$'s elements continue past the end of the common sum.

`dot(v,w)`

$$\mathbf{v} \cdot \mathbf{w} = v_1 w_1 + v_2 w_2 + \ldots + v_p w_p$$

where $p = \min\{m, n\}$.

`magn(v)`

$$|\mathbf{v}| = \sqrt{\mathbf{v} \cdot \mathbf{v}}$$

3 New Operations

Our new operations are defined as follows:

`project(w,v)`

$$P_{\mathbf{w}}\mathbf{v} = \frac{\mathbf{v}\cdot\mathbf{w}}{\mathbf{w}\cdot\mathbf{w}}\mathbf{w}$$

`projectAll(S,v)`

$$P_{\{\mathbf{w}_1,\ldots,\mathbf{w}_p\}}\mathbf{v} = P_{\mathbf{w}_1}\mathbf{v} + \ldots + P_{\mathbf{w}_n}\mathbf{v}$$

`inSpan(S,v)`

$$\mathbf{v} \in \text{span}(S) \text{ iff } |\mathbf{v} - P_S\mathbf{v}| < \epsilon$$

where ϵ is a suitably small number, and vector subtraction is defined using addition:

$$\mathbf{v} - \mathbf{w} = \mathbf{v} + (-1)\mathbf{w}$$

In your program, you should define `FUDGE` as a small number and use it for ϵ:

```
val FUDGE = 0.000001;
```

4 Example

```
- val w1:vector = [1.0];
val w1 = [1.0] : vector
- val w2:vector = [0.0,1.0,1.0];
val w2 = [0.0,1.0,1.0] : vector
- val w3:vector = [0.0,0.0,0.0,2.0];
val w3 = [0.0,0.0,0.0,2.0] : vector
- val S = [w1,w2,w3];
val S = [[1.0],[0.0,1.0,1.0],[0.0,0.0,0.0,2.0]] : vector list
- projectAll(S,[1.0,2.0,3.0,4.0]);
val it = [1.0,2.5,2.5,4.0] : real list
- inSpan(S,[1.0,2.0,3.0,4.0]);
val it = false : bool
- projectAll(S,[1.0,2.0,2.0,3.0]);
val it = [1.0,2.0,2.0,3.0] : real list
- inSpan(S,[1.0,2.0,2.0,3.0]);
val it = true : bool
```

5 Overloading and Polymorphism

Do Exercises 3 and 4 of Chapter 8 (page 131).

6 Submitting Your Work

Leave `vector2.sml` in your AFS folder. Turn in the book exercises on paper at the beginning of lecture.