10-25

CS 53, Fall 2017

Due October 27 at 2:59 pm

Problems 6.7.2, 6.7.5 in the textbook (using the usual stencil `Dimension_problems.py`, and the following problems not in the textbook (using `Dimension_other_problems.py`).

Problem 1: Write and test a procedure `exchange(S, A, z)` with the following spec:

- *input:* A set S of vectors, a set A of vectors that are all in S (such that len$(A) <$ len(S)), and a vector $\boldsymbol{z}$ such that $A \cup \{z\}$ is linearly independent
- *output:* a vector $\boldsymbol{w}$ in S but not in A such that

$$\text{Span } S = \text{Span } (\{\boldsymbol{z}\} \cup S - \{\boldsymbol{w}\})$$

Your procedure should follow the proof of the Exchange Lemma. You should use the `solver` module or the procedure `vec2rep(veclist, u)` from a previous problem. You can test whether a vector is in a list or set `C` using the expression `v in C`. Note that S need not be linearly independent.

Problem 2: Write and test a procedure `subset_basis(T)` with the following spec:

- *input:* a set T of vectors
- *output:* a set S consisting of vectors of T such that S is a basis for the span of T.

Your procedure should be based on either a version of the Grow algorithm or a version of the Shrink algorithm. Think about each one to see which is easier for you. You will need a loop or comprehension for this procedure. You can use as a subroutine any one of the following:

- the procedure `is_superfluous(L, b)` from `The_Basis_problems`, or
- the procedure `is_independent(L)` from `The_Basis_problems` or from the module `independence` we provide, or
- the procedure `solve(A,b)` from the `solver` module.

Disclaimer: The algorithm you are intended to use here is pedagogically motivated. It is intended to illustrate and illuminate our proof of the Subset-Basis Lemma. Ideas discussed later give rise to better algorithms for this computational problem.