Robert Hergenroder II
Dr. Graham
Lin 177 Final Project

Latin is an extremely concise language due to its inflection. Being heavily inflected, semantic precision in Latin is given via extensive suffixation. Nouns in English, for example, are differentiated in grammatical case by word positioning and helpers such as prepositions; whereas Latin differentiates grammatical case via suffixation. This reduces the number of, and makes null the importance of ordering, words in Latin.

Latin nouns have six primary cases they must account for-- nominative, vocative, accusative, genitive, dative, and ablative. These refer to subject, directly addressing a noun, direct objects, possession of another noun, indirect objects, and objects of prepositional phrases in English respectively. Each of these Latin nouns have a specific gender which must be accounted for, as well as for their number. The process of forming the suffixes to account for case, number, and gender is declension.

Latin has five declensions-- each of which is based upon gestalts concerning their case, number and gender, e.g. first declension nouns are mostly feminine, second mostly masculine etc.. Declension is known by the genitive singular form of a word. Declining a Latin noun, then, is the process of marking a particular stem with a particular suffix in a strict, may I even say computational, manner.

"Latin_declension.pl" is an attempt at automatization of declining Latin nouns. The primary predicate of "latin_declension.pl" is as follows:

decline_word(INPUT_WORD, CASE, NUMBER, {GENDER}, OUTPUT_WORD).

Where

INPUT_WORD is the word desired to be declined in it's singular genitive form
CASE is one of the following: [nom], [voc], [acc], [gen], [dat], [abl]
NUMBER is either [sing] or [pl]
{GENDER} is optional, and is [m], [f], or [n]
OUTPUT_WORD is the word properly declined

(Note: I have attached a set of Latin nouns for testing purposes as pages 3 and 4.)

By allowing gender to be optional, I hope to allow occurrences where acute gender is not known, and the major paradigm of the declension can be followed.

The program mostly works as intended. Each declension has subsets of word-forms which dictate the manner in which they are declined, the formation of said subsets in the third declension is somewhat tricky to model. While the majority of third declension nouns can be properly declined after looking at how their stem ends-- one requires implementation of the

so-called Parasyllabic rule. This rule states that when the nominative and genitive singular forms have the same number of syllables-- the word can be, but isn't always, declined in a specific, unique manner. My program fails at differentiating third declension nouns with this form-- I simply didn't have the time to code in syllable recognition. Other than that-- the program should be able to decline any other Latin noun properly.

The algorithm utilized in "latin_declension.pl" seems mostly principled in that it follows the same process I remember utilizing to decline nouns by hand; e.g. figure out which declension the word is, figure out its stem form, attach the appropriate ending. The only manner in which "latin_declension.pl" is not principled is that it doesn't differentiate long and short vowels. (Many Roman authors did differentiate the vowels, even if they didn't differentiate words with spaces. Although there is no loss in meaning, pronounceability is completely lost without distinguishing long and short vowels.)

Improvement of "latin_declension.pl" would include differentiating long and short vowels, as well as a method for determining when to apply the Parasyllabic rule. Irregular noun forms, there are at least 6 that I am acutely aware of, would also be hard coded in as special cases. In the same way that supplying gender is optional, other heuristics for determining declension with suboptimal input could be coded as well.

(Note: when testing sometimes the gender must be given for proper declension as the heuristics fail.)

First Declension:

Puella, *[f] girl*

	[sing]	[pl]
[nom]	puella	puellae
[voc]	puella	puellae
[acc]	puellam	puellas
[gen]	puellae	puellarum
[dat]	puellae	puellis
[abl]	puella	puellis

Second Declension:

Murus, *[m] wall*

	[sing]	[pl]
[nom]	murus	muri
[voc]	mure	muri
[acc]	murum	muros
[gen]	muri	murorum
[dat]	muro	muris
[abl]	muro	muris

Bellum, *[n] war*

	[sing]	[pl]
[nom]	bellum	bella
[voc]	bellum	bella
[acc]	bellum	bella
[gen]	belli	bellorum
[dat]	bello	bellis
[abl]	bello	bellis

Puer, *[m] boy*

	[sing]	[pl]
[nom]	puer	pueri
[voc]	puer	pueri
[acc]	puerum	pueros
[gen]	pueri	puerorum
[dat]	puero	pueris
[abl]	puero	pueris

Vir, *[m] man*

	[sing]	[pl]
[nom]	vir	viri
[voc]	vir	viri
[acc]	virum	viros
[gen]	viri	virorum
[dat]	viro	viris
[abl]	viro	viris

Third Declension:

Dux, *[m] leader*

	[sing]	[pl]
[nom]	dux	duces
[voc]	dux	duces
[acc]	ducem	duces
[gen]	ducis	ducum
[dat]	duci	ducibus
[abl]	duce	ducibus

Virtus, *[f] virtue*

	[sing]	[pl]
[nom]	virtus	virtutes
[voc]	virtus	virtutes
[acc]	virtutem	virtutes
[gen]	virtutis	virtutum
[dat]	virtuti	virtutibus
[abl]	virtute	virtutibus

Nomen, *[n] name*

	[sing]	[pl]
[nom]	nomen	nomina
[voc]	nomen	nomina
[acc]	nomen	nomina
[gen]	nominis	nominum
[dat]	nomini	nominibus
[abl]	nomine	nominibus

Fourth Declension:

Portus, *[m] port*

	[sing]	[pl]
[nom]	portus	portus
[voc]	portus	portus
[acc]	portum	portus
[gen]	portus	portuum
[dat]	portui	portibus
[abl]	portu	portibus

Cornu, *[n] horn*

	[sing]	[pl]
[nom]	cornu	cornua
[voc]	cornu	cornua
[acc]	cornu	cornua
[gen]	cornus	cornuum
[dat]	cornui	cornibus
[abl]	cornu	cornibus

Fifth Declension:

Dies, *[m] day*

	[sing]	[pl]
[nom]	dies	dies
[voc]	dies	dies
[acc]	diem	dies
[gen]	diei	dierum
[dat]	diei	diebus
[abl]	die	diebus

Res, *[f] thing*

	[sing]	[pl]
[nom]	res	res
[voc]	res	res
[acc]	rem	res
[gen]	rei	rerum
[dat]	rei	rebus
[abl]	re	rebus