CS447 Lab #3 (Week 4)

Introduction

For this lab, you will practice using branching and bitwise logical instructions in MIPS.

Note: This Lab must be submitted online by 11:59 pm tonight. Please see the instructions on the final page.

Objective: Use a binary representation of yes/no answers to determine the correct path through a decision tree.

In many domains, hierarchies of categories and subcategories can be represented by decision trees. A very simple form of tree involves no more than a single 'yes' or 'no' decision on each level, with each level of the tree providing a finer-grained level of categorization. For example, in Figure 1, the first yes/no decision leads to a classification of matter as either "pure" or a "mixture", and the second decision leads to finer-grained classifications within those categories.

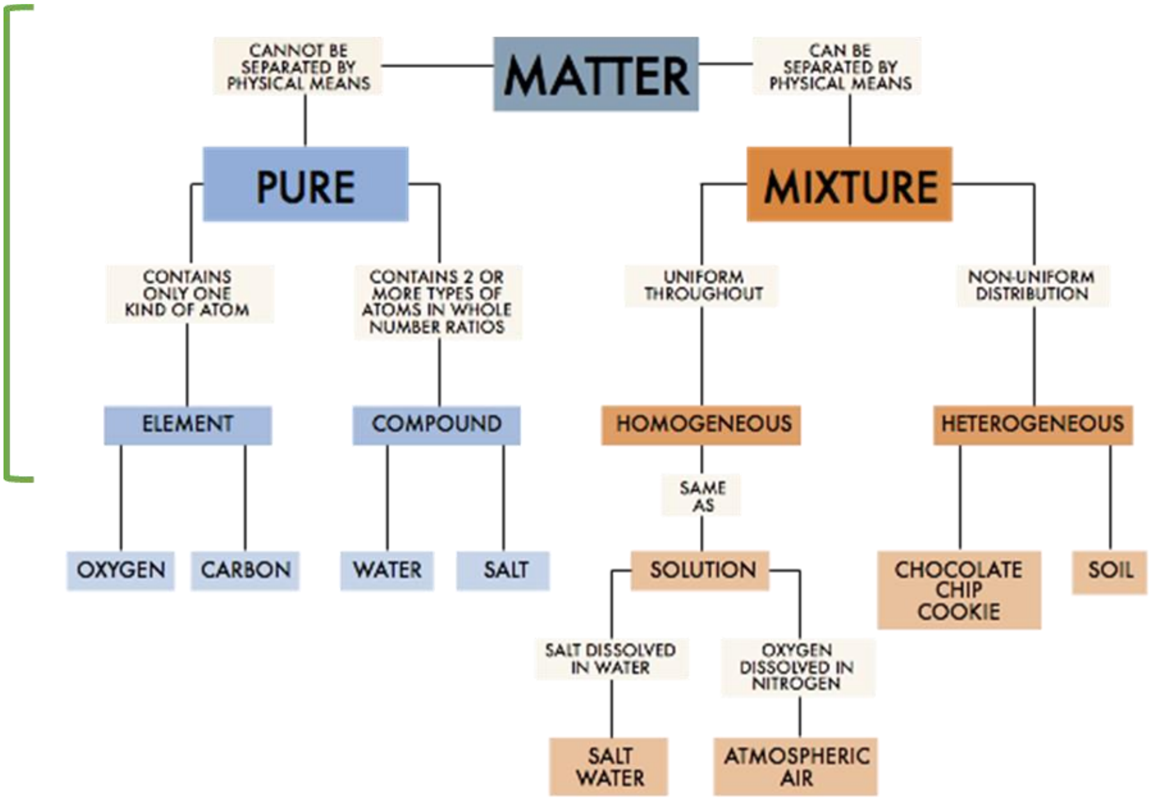

Figure 1. Source: https://students.ga.desire2learn.com/d2l/lor/viewer/viewFile.d2lfile/1798/12889/matter4.html. Additional reference: http://www.kentchemistry.com/links/Matter/ClassifiyingMatter.htm.

If you've worked with trees before, you may be familiar with the convention of labeling paths through a tree with a series of 0's and 1's. To follow a path encoded according to this convention, you start at the root of the tree (here, the node labeled **Matter**), and then read the binary string from left to right. You go down a **left** branch each time you encounter a **0**, and go down a **right** branch each time you encounter a **1**. For example, for Figure 1, the binary string **01** would tell you to travel through **Matter → Pure → Compound**.

In the Lab #3 exercise, you will decode binary encoded yes/no responses to classify matter samples as either a pure substance or mixture. If the sample is pure, you will further determine whether it is an element or a compound, and if it is a mixture, you will determine whether it is a homogeneous mixture or a heterogeneous mixture.

Prepared code:

For Lab #3, some starter code has been prepared for you. This can be downloaded from the Github repository: https://github.com/kc13/CS447. The file name is **lab3.asm**.

If you assemble and run **lab3.asm**, you'll notice that it sets up a basic interactive program, which asks the user the questions needed to descend through the Figure 1 decision tree. An example run is pasted below:

```
Welcome to the matter classification program!
Can substances within the sample be separated by physical means? (no = 0, yes = 1)
0
Does the sample contain more than one kind of atom? (no = 0, yes = 1)
1
```

From the answers, we know that the sample is a compound.

It is important to make note of how the user's response is stored. The idea is to encode the response as a binary string that can be followed as a path through the decision tree. For the purpose of these instructions, assume that the bit positions in the binary string are labeled in ascending order from right to left (for example, in a byte, the least significant bit is at position **0**, and the most significant bit is at position **7**).

The program will encode the user's response in the 2 least-significant bits of $t0 (bit 1 and bit 0). The current instructions perform this encoding in two steps. First, on lines **27-29**, we store the first step of the path:

```
# store result in bit 1 of $t0 (where bit 0 = Least Significant Bit)

add $t0, $zero, $v0

sll $t0, $t0, 1
```

As is typical for the “read integer” **syscall**, the user’s response is initially stored in **$v0.** These instructions place the contents of **$v0** in **$t0** with an **add** instruction (**move** would also be fine)**,** and then the digit is shifted one position to the left (to bit position 1) with a **Shift left logical** (**sll**) instruction.

The second response is added on line 41:

# store result in bit 0 of $t0 (the LSB)

or $t0, $t0, $v0

This **or** instruction will ensure that the first response will remain encoded in bit position 1 of **$t0** (since the corresponding bit will be **0** in **$v0**, and we know that **or(x,0) = x**). Bit 0 will be updated to the user’s second response (since the **sll** operation cleared out the 0th bit of **$t0**).

These operations imply that the final contents of **$t0** may contain 1 of 4 values: **00**, **01**, **10**, or **11**. Remember that they will appear differently in the register file, which uses hexadecimal formatting.

Lab exercise:

To complete the lab, you should **modify lab3.asm** so that it does the following:

(1) Uses the bit in position 1 of **$t0** to determine whether the matter sample is a pure substance or a mixture.

→ Print the output of this classification to the screen. To save time, some messages have already been prepared in the **.data** section (although you are welcome to change them if you prefer something different).

(2) Use the bit in position 0 of **$t0** to determine whether the sample is an element or a compound (for a pure substance) or a homogeneous or heterogeneous mixture.

→ Print the output of this classification

Here is a sample output:

```
Welcome to the matter classification program!
Can substances within the sample be separated by physical means? (no = 0, yes = 1)
0
Does the sample contain more than one kind of atom? (no = 0, yes = 1)
1
This is a pure substance.
It is a compound.
```

Note: There are multiple ways to complete this assignment. Some instructions that you might find helpful include branches (for example, **beq/bne/j**), and the bitwise **and** (or **andi**) instructions for masking out individual bits.

Submission:

If you finish during recitation, please notify the TA, so that your credit for this assignment can be confirmed right away.

Regardless of whether you finish during recitation or not, you must submit your work online. Save the file as **YourPittUserName_lab3.asm** and submit the file via the appropriate link in Courseweb (see Course Documents/Week 4 Lab). The lab must be submitted **today, September 23rd**, by **11:59 pm**.