

HOMEWORK ASSIGNMENT: CHAPTER 13**Question 13.1:**

a) Provide a precise definitions of the statements:

- all p-use some c-use
- all c-use some p-use

I interpreted (a) to mean by 'predicate' that the DU pair was located within an if() branch for control-flow and "computational" uses referred to DU pairs located inside the body or similar to aliases, I defined the two criteria as follows:

- **All p-use some c-use:**

A test suite T for a program P satisfies the all p-use, some c-use criteria iff, for at least one definition def of P and for each DU path dp of P , there exists at least one test case in T that exercises a DU pair that includes def and another that includes dp .

- **All c-use some p-use:**

A test suite T for a program P satisfies the all p-use, some c-use criteria iff, for each definition def of P and for at least one DU path dp of P , there exists at least one test case in T that exercises a DU pair that includes def and another that includes dp .

b) Describe the differences in the test suites derived applying the different criteria to function `cgi_decode` in Figure 13.1.

I interpreted that at least one predicate DU execution sequence would always attempt to execute the DU inside of the first control-flow loop encountered for the purposes of my description of satisfying the criteria in the context of 13.1, knowing this would not necessarily be the case. Without external knowledge of the state of the program by which to confine the paths, this rule could not be assumed and thus the possible pairs/paths executable could be infinite, so it is determinedly limited in my description below:

- **All p-use some c-use:**

Because there is only

- **All c-use some p-use:**

Question: 13.2:

Demonstrate the subsumes relation between all p-use some c-use, all c-use some p-use, all DU pairs, all DU paths and all definitions.

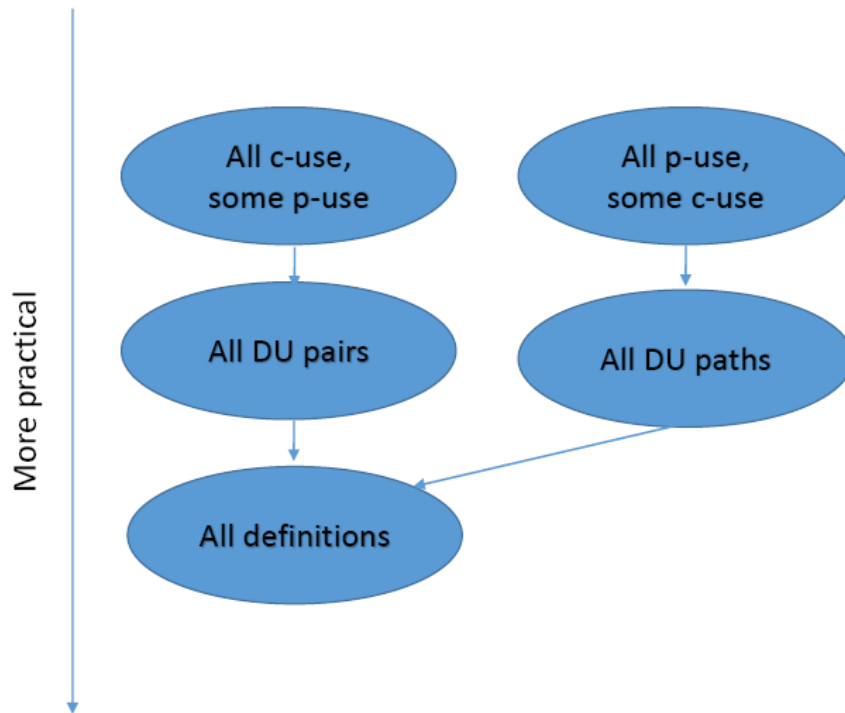
Response:

Figure: Subsumes relations for DU

Question 13.3:

How would you treat the figure 16.1 (transduce() procedure)?

Response:

It is a bit unclear to me exactly what they mean to ask in this question, but I'd say:

-not knowing the details of the emit() procedure's implementation, we would not know whether 'pos' is redefined outside the scope of the transduce() procedure, so we would have to limit the scope of our analysis to the procedure we can see, firstly, to be practical. Then, although there could be many cases where the 'default' case is reached in the switch-control, there are only two cases for the non-default conditions to be true, all of which are exclusive to the other, in terms of any DU pairs/paths.

Admittedly, maybe I'm not understanding the material in this chapter/question, but it seems, the treatment of the procedure is a matter of determining an adequately sized test suite, depending on the DU test adequacy criteria-level determined appropriate. In any case, a minimally -thorough/-sized test suite *T* might be comprised of at least 3 TCases (where the test suite represents standard input's contents) to reach all loops consisted of:

- {'\0', '#LF', '#CR'}