Does column entry subsume the row entries?

	Block	Statement	Condition	Decision	Condition-Decision	Multiple-condition	c-use	p-use	All-use
Block		Y	N	Y	Y	Y	N	Y	Y
Statement	Y		N	Y	Y	Y	N	Y	Y
Condition	N	N		N	Y	Y	N	N	N
Decision	N	N	N		Y	Y	N	Y	Y
Condition-decision	N	N	N	N		Y	N	N	N
Multiple-condition	N	N	N	N	N		N	N	N
c-use	N	N	N	N	N	N		N	Y
p-use	N	N	N	N	N	N	N		Y
All-use	N	N	N	N	N	N	N	N	

(1) Block (B) vs. Statement (S)

Every statement must belong to some block, thus covering every block is equivalent to covering every statement. So B subsumes S.

Every block must contain at least one statement, thus covering every statement is equivalent to covering every block. So S subsumes B.

B ≡ S

(2) Block (B) vs. Condition (C)

Take the following code as example:

```
Stmt1;
Stmt2;
If ((a>b) && (c >d))
{stmt3;}
Stmt4
```

(here for ease of reference, we use C1 denote for (a>b) and C2 denote for (c>d))

A: Case: If C1 is True and C2 is True, then all blocks and statements are covered, but conditions are not covered. So B does not subsume C and S does not subsumes C.

B: suppose we have 2 cases: Case1 -> C1 is True and C2 is False; and Case2 -> C1 is false and C2 is True. Then Case1 and Case2 together satisfy condition coverage while block coverage is not satisfied (since stmt3 wont be touched). So C does not subsume B and C does not subsume S.

(3) Block (B) vs Decision (D)

Using part A of the example above, all statements are covered, but the decision only took the 'True' branch. So S does not subsume D.

Every statement must be part of some branch; or by default it will be covered. So covering both 'True' or 'False' branch of a decision, leads to covering every statement. So D subsumes S and D subsumes B.

(4) Condition (C) vs. Decision (D)

Using part B of the example above, condition can be satisfied without satisfying decision. So C does not subsume D.

C: If we change the example to: if (C1 || C2), then consider:

C1 is True and C2 is False -> true branch

C1 is False and C2 is False -> false branch

We can see from this example that the decision is covered without condition covered. So D does not subsume C.

(5) Condition-Decision (CD) vs. Condition (C)

By definition, CD subsumes C.

But C does not subsume CD as C does not subsume D.

(6) Condition-Decision (CD) vs. Decision (D)

By defination CD subsumes D.

But D does not subsume CD as D does not subsume C.

(7) Block (B) vs. Condition-Decision (CD)

B does not subsume C or D and therefore, B does not subsume CD.

CD subsumes B as CD subsumes D which subsumes B.

(8) multiple condition (MC) vs. Condition (C)

Every possible combination of conditions -> both true|false outcomes of every condition has been exercised. Therefore MC subsumes C.

Recall Part B of (2), C can be satisfied without satisfying MC. So C does not subsume MC.

(9) Multiple Condition (MC) vs. Decision (D)

The true/false branch of each decision must be coverable by some combination of its constituent conditions. So covering all combinations ensures covering both true/false branches. Thus MC susumes D.

Part C in (4) gives an example to show how D can be stisfied without sarisfying MC. So D does not subsume MC.

(10) Block (B)/ Statement (S) vs. Multiple Condition (MC)

Part A in (2) shows how B/S can be sarisfied withouth satisfying MC. So B/S does not subsume MC

MC subsumes D and D subsumes B/S. So MC subsumes B/S.

(11) Multiple Condition (MC) vs. Condition-Decision (CD)

Since MC susumes C and D, MC subsumes CD.

Taking the example of part B in (2), if we add one more test case C1 is true and C2 is true then CD is satisfied. However we still missing case C1 is false and C2 is false. In that case MC is not satisfied. So CD does not subsume MC.

(12) c-use vs. p-use

No set relationship between c-use and p-use.

So c-use does not subsume p-use.

p-use does not subsume c-use.

(13) c-use vs. Decision (D)

D cannot have every def-use pair to be covered.

For example, a loop where def at 'x' is used at 'x'.

So D does not subsume c-use.

c-use cannot ensure every decision takes a true/false branch.

So c-use does not subsume D. And for the same reason, c-use does not subsumes C, B/S. or MC.

(14) MC vs. c-use

No set relationship between MC and c-use.

Covering each def-use pair does not ensure each multiple condition combination is covered and vice versa.

So MC does not subsume c-use and vice versa.

(15) p-use vs. Decision

For p-use coverage, every p-use must independently contribute to the decision taking both a ‘true’ + ‘false’ branch. The focus is on the fact that both branches are taken. So p-use subsumes Decision. Plus, p-use subsumes B/S.

Just because each branch was taken, does not mean each p-use contributed. So D does not subsume p-use.

(16) p-use vs. Condition (C)

Similar arguments for D vs C. p-use doesn not subsume C.

Similar arguments for C vs D. C doesn not subsume p-use.

Also can infer that p-use does not subsume MC.

And MC does not subsume p-use since MC cannot guarantee each variable independently made the decision take ‘T’ + ‘F’.

(17) c-use vs. all-uses

By definition all-uses subsumes c-use/p-use.

But c-use/p-use does not subsume all-uses.

p-use does not subsume c-use.

c-use does not subsume p-use.

(18) all-uses vs. Decision (D)

D cannot subsume c-use or p-use. So D does not subsume all-uses.

All-uses subsumes p-use which subsumes D. So all-uses subsumes D.

No set relationship between all-uses and ther others.