31 Tuesday

IIT2018010 DBMS MAKEUP

25/04/20

Solution Explanation for calculating minimal canonical form.

Sol^n: Step 1: Take input of number of elements present in domain.

Step 2: Keep a track of all those present (possibly in map structure).

Step 3: Take the input of fuctional dependency
Eg: Say if Relation is AB → CD,
Take input as AB - CD

Step : 4 While taking input, check if all parameters are valid,
if yes → proceed
if no → terminate.

Step : 5 While taking input perform union structure (function) on input for simplicity.

Eg: If input is
A → CD
and already if A → B
then store input as A → BCD

1 Wednesday

Step 6: Then apply decomposition rule to remove any redundancy (extraneous attribute on Right-hand-Side).

→ After doing this either the rule is extraneous as whole

→ Or there is redundant (extraneous attribute) on Left hand Side only.

Ex: $A \to BC$ is decomposed as $A \to B$, $A \to C$.

Step 7:

Check if extraneous rules. (fd)

for doing so,

$\forall i \in 1, 2, \ldots n$ where n is number of rules

..

we have → deleted array [to check if rule is extraneous / deleted.

To check if any rule $AB \to CD$ needs to be deleted or not,

get closure using rule / Not using above rule

→ If both are same, delete the rule.

Step 8: Getting enclosure for deleting rule.

→ Start with states on L.H.S.
→ for including rule → unlock states on R.H.S.

For each rule make a visited array to see if it is visited

→ if i in 1... n;
vis[n] = 0
→ ~~string~~ set enclosure? = empty
set.push (LHS states)
set.push (R.H.S. states). ← i
vis[i] = true.

→ Now repeat a loop
→ In loop do not visit deleted or already visited rules.
→ If a rule is unvisited,
check if we have

3 Friday

all of its attributes on (L.H.S.) in enclosure set,

If so, then this rule can be unlocked/ visited

→ visit it.

→ If new attribute is added, from R.H.S of above rule,

↙ Yes

There is a possibility that now if a rule is not - visited yet,

It can be visited now → REPEAT LOOP AGAIN.

Else → If new attribute is added → TERMINATE LOOP.

→ This gives enclosure using rule.

For calculating enclosure without rule all above steps remain same but (i)

(do not perform it)

→ Now if above two enclosures are same, ~~not perform~~ rule can be deleted.

→ delete it.

4 Saturday Step 9: Now extra rules have been removed

Extra attributes for L.H.S need to be deleted for each Rule. (that is not deleted).

For each part on L.H.S

While there can be an attribute that can be deleted:

- Get the enclosure using remaining attributes
- If the rule was valid using remaining attributes, this rule can be unlocked to its R.H.S. side,

→ Go Ahead.

Perform enclosure as in Step 8.

- Now perform enclosure without this attribute.
- Now we do not have all the valid required attributes of L.H.S,

we DO NOT INCLUDE R.H.S here.

5 Sunday

6 Monday

→ Perform another enclosure without the attribute that we are testing if it can be deleted.

→ In case in above process we found the attribute somewhere else in same another rule.
(transitivity applies here)

→ then we can unlock / visit Rest of account rule.
→ visit it.

→ Example: A → B
A B → C

B can be found from A → B.

→ If above two enclosures are equal, this attribute is extragenous
→ delete it.

→ Perform the same for Remaining rules.
→ Finally perform the Union to minimize number of Rules (using map and set).

→ Finally, print the results

7 Tuesday Flow chart:-

TAKE DOMAIN INPUTS

↓

TAKE FUNCTIONAL DEPENDENCY INPUT

↓

CHECK VALIDITY ← NO → TERMINATE

↓ YES

PERFORM UNION ON ALL FD'S.

↓

APPLY DECOMPOSITION

↓

REMOVE INVALID RULES

↓

CHECK FOR EXTRANEOUS ATTRIBUTES ON LHS IN REMAINING RULES

↓

PERFORM UNION

↓

PRINT RESULT.