**Closure of a Set of Functional Dependencies**

1. We need to consider *all* functional dependencies that hold. Given a set *F* of functional dependencies, we can prove that certain other ones also hold. We say these ones are **logically implied** by *F*.
2. Suppose we are given a relation scheme *R*=(*A*,*B*,*C*,*G*,*H*,*I*), and the set of functional dependencies:
3. *A tex2html_wrap_inline1090 B*

*A tex2html_wrap_inline1090 C*

*CG tex2html_wrap_inline1090 H*

*CG tex2html_wrap_inline1090 I*

*B tex2html_wrap_inline1090 H*

Then the functional dependency tex2html_wrap_inline1194is logically implied.

1. To see why, let tex2html_wrap_inline940and tex2html_wrap_inline946be tuples such that

tex2html_wrap_inline1200

As we are given *A tex2html_wrap_inline1090B* , it follows that we must also have

tex2html_wrap_inline1204

Further, since we also have *B tex2html_wrap_inline1090H* , we must also have

tex2html_wrap_inline1208

Thus, whenever two tuples have the same value on *A*, they must also have the same value on *H*, and we can say that *A tex2html_wrap_inline1090H* .

The **closure** of a set *F* of functional dependencies is the set of all functional dependencies logically implied by *F*.

We denote the closure of *F* by tex2html_wrap_inline1222.

To compute tex2html_wrap_inline1222, we can use some rules of inference called **Armstrong's Axioms**:

* + **Reflexivity rule:** if tex2html_wrap_inline958is a set of attributes and tex2html_wrap_inline1158, then tex2html_wrap_inline1058holds.
  + **Augmentation rule:** if tex2html_wrap_inline1058holds, and tex2html_wrap_inline1234is a set of attributes, then tex2html_wrap_inline1236holds.
  + **Transitivity rule:** if tex2html_wrap_inline1058holds, and tex2html_wrap_inline1240holds, then tex2html_wrap_inline1242holds.

These rules are **sound** because they do not generate any incorrect functional dependencies. They are also **complete** as they generate all of tex2html_wrap_inline1222.

To make life easier we can use some additional rules, derivable from Armstrong's Axioms:

* + **Union rule:** if tex2html_wrap_inline1058and tex2html_wrap_inline1242, then tex2html_wrap_inline1250holds.
  + **Decomposition rule:** if tex2html_wrap_inline1250holds, then tex2html_wrap_inline1058and tex2html_wrap_inline1242both hold.
  + **Pseudotransitivity rule:** if tex2html_wrap_inline1058holds, and tex2html_wrap_inline1260holds, then tex2html_wrap_inline1262holds.

Applying these rules to the scheme and set *F* mentioned above, we can derive the following:

* + *A tex2html_wrap_inline1090H*, as we saw by the transitivity rule.
  + *CG tex2html_wrap_inline1090HI* by the union rule.
  + *AG tex2html_wrap_inline1090I* by several steps:
    - Note that *A tex2html_wrap_inline1090C* holds.
    - Then *AG tex2html_wrap_inline1090CG* , by the augmentation rule.
    - Now by transitivity, *AG tex2html_wrap_inline1090I* .

(You might notice that this is actually pseudotransivity if done in one step.)

**Canonical Cover**

To minimize the number of functional dependencies that need to be tested in case of an update we may restrict *F* to a **canonical cover** tex2html_wrap_inline1324.

A canonical cover for *F* is a set of dependencies such that *F* logically implies all dependencies in tex2html_wrap_inline1324, and vice versa.

tex2html_wrap_inline1324must also have the following properties:

* + Every functional dependency tex2html_wrap_inline1058in tex2html_wrap_inline1324contains no **extraneous** attributes in tex2html_wrap_inline958(ones that can be removed from tex2html_wrap_inline958without changing tex2html_wrap_inline1342). So *A* is extraneous in tex2html_wrap_inline958if tex2html_wrap_inline1348and

displaymath1318

logically implies tex2html_wrap_inline1324.

* + Every functional dependency tex2html_wrap_inline1058in tex2html_wrap_inline1324contains no **extraneous** attributes in tex2html_wrap_inline1356(ones that can be removed from tex2html_wrap_inline1356without changing tex2html_wrap_inline1342). So *A* is extraneous in tex2html_wrap_inline1356if tex2html_wrap_inline1366and

displaymath1319

logically implies tex2html_wrap_inline1324.

* + Each left side of a functional dependency in tex2html_wrap_inline1324is unique. That is there are no two dependencies tex2html_wrap_inline1372and tex2html_wrap_inline1374in tex2html_wrap_inline1324such that tex2html_wrap_inline1378.

To compute a canonical cover tex2html_wrap_inline1324for *F*,

**repeat**

Use the union rule to replace dependencies of the form

tex2html_wrap_inline1372 and

tex2html_wrap_inline1386 with tex2html_wrap_inline1388 .

Find a functional dependency tex2html_wrap_inline1058 with an extraneous attribute in tex2html_wrap_inline958 or in tex2html_wrap_inline1356 .

If an extraneous attribute is found, delete it from tex2html_wrap_inline1058

**until** *F does not change.*

An example: for the relational scheme *R*=(*A*,*B*,*C*), and the set *F* of functional dependencies

*A tex2html_wrap_inline1090 BC*

*B tex2html_wrap_inline1090 C*

*A tex2html_wrap_inline1090 B*

*AB tex2html_wrap_inline1090 C*

we will compute tex2html_wrap_inline1324.

* + We have two dependencies with the same left hand side:
  + *A tex2html_wrap_inline1090 BC*
  + *A tex2html_wrap_inline1090 B*

We can replace these two with just *A tex2html_wrap_inline1090BC* .

* + *A* is extraneous in *AB tex2html_wrap_inline1090C* because *B tex2html_wrap_inline1090C* logically implies *AB tex2html_wrap_inline1090C* .
  + Then our set is *A tex2html_wrap_inline1090 BC*

*B tex2html_wrap_inline1090 C*

We still have an extraneous attribute on the right-hand side of the first dependency.

*C* is extraneous in *A tex2html_wrap_inline1090BC* because *A tex2html_wrap_inline1090B* and *B tex2html_wrap_inline1090C* logically imply that *A tex2html_wrap_inline1090BC* .

So we end up with

*A tex2html_wrap_inline1090 B*  *B tex2html_wrap_inline1090 C*