Functional Dependencies

Stéphane Bressan

number	name	department	position	salary
1XU3	Dewi Srijaya	sales	clerk	2000
5CT4	Axel Bayer	marketing	trainee	1200
4XR2	John Smith	accounting	clerk	2000
7HG5	Eric Wei	sales	assistant manager	2200
4DE3	Winnie Lee	accounting	manager	3000
8HG5	Sylvia Tok	marketing	manager	3000

The table above records the salaries of the different employees in our organisation. An agreement with the trade unions imposes that salaries are determined by the position. The actual value has been negotiated and fixed. The salary of a clerk is 2000$ per month, the salary of a manager is 3000$ per month, etc.

Salaries are determined by the position.

This kind of business rule can be translated into an integrity constraint called a functional dependency. It is an integrity constraint. We write:

$$\{position\} \rightarrow \{salary\}$$

$$\{position\} \rightarrow \{salary\}$$

This means that we should not encounter a table in which two employees have the same position but different salaries.

number	name	department	position	salary
1XU3	Dewi Srijaya	sales	clerk	2000
5CT4	Axel Bayer	marketing	trainee	1200
4XR2	John Smith	accounting	clerk	2000
7HG5	Eric Wei	sales	assistant manager	2200
4DE3	Winnie Lee	accounting	manager	3000
8HG5	Sylvia Tok	marketing	manager	4000

Definition

An instance r (a table) of a relation schema R satisfies the functional dependency σ: $X \rightarrow Y$ with $X \subset R$ and $Y \subset R$, if and only if if two tuples of r agree on their X-values, then they agree on their Y-values.

$X \rightarrow Y$ reads: X functionally determines Y, X determines Y, Y is functionally dependent on X, or, more casually, X implies Y.

Definition

An instance r of a relation schema R is a valid instance of R with Σ if and only if it satisfies Σ.

Definition

An instance r of a relation schema R violates a set of functional dependencies Σ if and only if it does not satisfies Σ.

Definition

An instance r of a relation schema R violates a functional dependency σ if and only if it does not satisfies σ.

A relation R with a set of functional dependencies Σ, R with Σ, refers to the set of valid instances of R with respect to the functional dependencies in Σ.

When we say that a set of functional dependencies Σ holds on a relation R, we only consider the valid instances of R with Σ.

$R = \{A, B, C, D\}$

The following instance of R is valid for the functional dependency $\{A, B\} \rightarrow \{D\}$.

A	B	C	D
1	2	a	4
1	2	b	4
1	3	c	4

The following instance of R is violates the functional dependency $\{A, B\} \rightarrow \{D\}$.

A	B	C	D
1	2	a	4
1	2	b	3
1	3	c	4

$R = \{A, B, C, D\}$

The following (empty) instance of R is valid for the functional dependency $\{A, B\} \rightarrow \{D\}$. It is the smallest instance that does not violate the functional dependency.

A	B	C	D

The following instance of R is violates the functional dependency $\{A, B\} \rightarrow \{D\}$. It is the smallest instance that violates the functional dependency.

A	B	C	D
1	2	a	4
1	2	b	3

There are many functional dependencies in the case.

The following functional dependencies (among many others) hold in the case.

$$\{position\} \rightarrow \{salary\}$$
$$\{number\} \rightarrow \{name\}$$
$$\{number\} \rightarrow \{number, name, department, position\}$$
$$\{number\} \rightarrow \{number, name, department, position, salary\}$$
$$\{number\} \rightarrow \{number\}$$
$$\{name, department, salary\} \rightarrow \{name, salary\}$$

The following functional dependencies (among many others) do not hold in the case. Note that they may accidentally hold on a given instance (in particular, they always hold on the empty instance).

$$\{salary\} \rightarrow \{position\}$$
$$\{name\} \rightarrow \{number\}$$
$$\{department, name\} \rightarrow \{number, name, department, position\}$$
$$\{department, salary\} \rightarrow \{name, salary\}$$

Definition

A functional dependency $X \to Y$ is trivial if and only if $Y \subset X$.

$R = \{A, B, C\}$

$\{A\} \to \{A\}$ is trivial.

$\{A, B\} \to \{A\}$ is trivial.

$\{A, B\} \to \emptyset$ is trivial.

$$R = \{number, name, department, position, salary\}$$

$\{number\} \rightarrow \{number\}$ is trivial.

$\{name, department, salary\} \rightarrow \{name, salary\}$ is trivial.

Definition

A functional dependency $X \to Y$ is non-trivial if and only if $Y \not\subset X$.

$R = \{A, B, C\}$

$\{A\} \to \{B\}$ is non-trivial.

$\{A, C\} \to \{B, C\}$ is non-trivial.

$R = \{number, name, department, position, salary\}$

$\{position\} \rightarrow \{salary\}$ is non-trivial.

$\{number\} \rightarrow \{name\}$ is non-trivial.

$\{number\} \rightarrow \{number, name, department, position\}$ is non-trivial.

$\{number\} \rightarrow \{number, name, department, position, salary\}$ is non-trivial.

Definition

A functional dependency $X \to Y$ is completely non-trivial if and only if $Y \neq \emptyset$ and $Y \cap X = \emptyset$.

$R = \{A, B, C\}$

$\{A\} \to \{B\}$ is completely non-trivial.

$\{A, C\} \to \{B, C\}$ is not completely non-trivial.

$R = \{number, name, department, position, salary\}$

$\{position\} \rightarrow \{salary\}$ is completely non-trivial.

$\{number\} \rightarrow \{name\}$ is completely non-trivial.

$\{number\} \rightarrow \{name, department, position\}$ is completely non-trivial.

$\{number\} \rightarrow \{name, department, position, salary\}$ is completely non-trivial.

A superkey is a set of attributes of a relation whose knowledge determines the value of the entire t-uple.

Definition

Let R be a relation. Let $S \subset R$ be a set of attributes of R. S is a superkey of R if and only if $S \to R$.

A candidate key is a minimal superkey (for inclusion).

Definition

Let R be a relation. Let $S \subset R$ be a set of attributes of R. S is a candidate of R if and only if $S \to R$ and for all $T \subset S$, $T \neq S$, T is not a superkey of R.

The primary key is the candidate key that the designer prefers or the candidate key if there is only one.

$\{number\}$ is a superkey of the table because
$\{number\} \rightarrow \{number, name, department, position, salary, position, salary\}$ holds.

$\{number\}$ is a candidate key of the table because there is no subset S of the set $\{number\}$ such that
$S \rightarrow \{number, name, department, position, salary, position, salary\}$ holds.

$\{number, name\}$ is a superkey of the table because
$\{number, name\} \rightarrow \{number, name, department, position, salary\}$ holds.

$\{number, name\}$ is not a candidate key of the table because
$\{number\} \rightarrow \{number, name, department, position, salary\}$ holds.

Definition

Let Σ be a set of functional dependencies on a relation schema R. A prime attribute is an attribute that is appears in some candidate key of R with Σ (otherwise it is called a non-prime attribute).

$R = \{A, B, C, D\}$
$\Sigma = \{\{A, B\} \rightarrow \{C, D\}, \{C\} \rightarrow \{A, B\}\}$

The candidate keys of R with Σ are $\{A, B\}$ and $\{C\}$.
A is a prime attribute of R with Σ.
B is a prime attribute of R with Σ.
C is a prime attribute of R with Σ.
D is a non-prime attribute of R with Σ.

Definition

Let Σ be a set of functional dependencies of a relation schema R. The closure of Σ, noted Σ^+, is the set of all functional dependencies logically entailed by the functional dependencies in Σ.

$R = \{A, B, C, D\}$
$\Sigma = \{\{A\} \rightarrow \{B\}, \{C\} \rightarrow \{A\}\}$

$\Sigma^+ = \{\{A\} \rightarrow \{B\}, \{C\} \rightarrow \{A\}, \{A\} \rightarrow \{A\}, \{D\} \rightarrow \{D\}, \{A, B\} \rightarrow \{A\}, \{A, C\} \rightarrow \{B, C\}, \{A, D\} \rightarrow \{B\}, \{C\} \rightarrow \{B\}, \cdots\}$

Find

- a trivial functional dependency in Σ^+.
- a non-trivial but not completely non-trivial functional dependency in Σ^+.
- a completely non-trivial functional dependency in Σ^+.

$R = \{A, B, C, D\}$
$\Sigma = \{\{A\} \to \{B\}, \{C\} \to \{A\}\}$

$\{A, D\} \to \{B, C\} \in \Sigma^+?$

$\{C, D\} \to \{B, A\} \in \Sigma^+?$

Armed with only the definition of functional dependency, the problems of computing Σ^+ and of testing membership to Σ^+ are daunting tasks.

Definition

Two sets of of functional dependencies Σ and Σ' are equivalent if and only if they have the same closure.

$$\Sigma \equiv \Sigma'$$

$$\Sigma^{+} = \Sigma'^{+}$$

Definition

Σ' is a cover of Σ (and Σ is a cover of Σ') if and only if $\Sigma \equiv \Sigma$.

Are $\Sigma = \{\{A\} \to \{B\}, \{B\} \to \{C\}, \{C\} \to \{A\}\}$ and $\Sigma' = \{\{C\} \to \{A, B\}, \{A\} \to \{B, C\}, \{B\} \to \{A\}, \{A, B\} \to \{C\}\}$ equivalent?

The answer is yes, but we need more tools to check that efficiently (without computing Σ^+ and Σ'^+).

Definition

Let Σ be a set of functional dependencies of a relation schema R. The closure of a set of attributes $S \subset R$, noted S^+, is the set of all attributes that are functionally dependent on S.

$$S^+ = \{A \in R \mid \exists (S \rightarrow \{A\}) \in \Sigma^+\}$$

The closure of a set of attributes can be computed by the fix-point iterative application of the functional dependencies as production rules.

Algorithm 1: Attribute Closure Algorithm

input : S, Σ
output: S^+

begin
  $\Omega := \Sigma$; // Ω `stands for‘‘unused’’`
  $\Gamma := S$; // Γ `stands for ‘‘closure’’`
  while $X \to Y \in \Omega$ *and* $X \subset \Gamma$ **do**
    $\Omega := \Omega - \{X \to Y\}$;
    $\Gamma := \Gamma \cup Y$;
  return Γ;

$R = \{A, B, C, D\}$
$\Sigma = \{\{A\} \rightarrow \{B\}, \{C\} \rightarrow \{A\}\}$

Compute $\{C\}^+$ using Algorithm 1.

$R = \{A, B, C, D\}$
$\Sigma = \{\{A\} \rightarrow \{B\}, \{C\} \rightarrow \{A\}\}$

1. $\Omega = \{\{A\} \rightarrow \{B\}, \{C\} \rightarrow \{A\}\}$
   $\Gamma = \{C\}^+$
2. use $\{C\} \rightarrow \{A\}$ $(\{C\} \subset \Gamma)$
   $\Omega = \{\{A\} \rightarrow \{B\}\}$
   $\Gamma = \{C\} \cup \{A\} = \{C, A\}$
3. use $\{A\} \rightarrow \{B\}$ $(\{A\} \subset \Gamma)$
   $\Omega = \emptyset$
   $\Gamma = \{C, A\} \cup \{B\} = \{C, A, B\}$
4. return $\Gamma = \{C, A, B\}$

$$R = \{A, B, C, D\}$$
$$\Sigma = \{\{A\} \to \{B\}, \{C\} \to \{A\}\}$$
$$\{C\}^+ = \{A, B, C\}$$

$R = \{A, B, C, D\}$
$\Sigma = \{\{A\} \rightarrow \{B\}, \{C\} \rightarrow \{A\}\}$

We compute $\{C, D\}^+$.

We have $\{C, D\}$, therefore $C \in \{C, D\}^+$ and $D \in \{C, D\}^+$.
We know that $\{C\} \rightarrow \{A\}$ and $\{C\} \subset \{C, D\}^+$, therefore $A \in \{C, D\}^+$.
We know that $\{A\} \rightarrow \{B\}$ and $\{A\} \subset \{C, D\}^+$, therefore $B \in \{C, D\}^+$.
Therefore $\{C, D\}^+ = \{A, B, C, D\}$

Definition

Let R be a set of attributes. The following inference rules are the Armstrong Axioms.

- Reflexivity
  $\forall X \subset R\ \forall Y \subset R\ ((Y \subset X) \Rightarrow (X \to Y))$
- Augmentation
  $\forall X \subset R\ \forall Y \subset R\ \forall Z \subset R\ ((X \to Y) \Rightarrow (X \cup Z \to Y \cup Z))$
- Transitivity
  $\forall X \subset R\ \forall Y \subset R\ \forall Z \subset R\ ((X \to Y \wedge Y \to Z) \Rightarrow (X \to Z))$

Technically, the Armstrong Axioms are not axioms but inference rules.

Theorem

The Reflexivity inference rule is sound (correct, valid).

Theorem

The Augmentation inference rule is sound.

Theorem

The Transitivity inference rule is sound.

Theorem

The Armstrong Axioms are complete.

$R = \{A, B, C, D\}$
$\Sigma = \{\{A\} \rightarrow \{B\}, \{C\} \rightarrow \{A\}\}$

Can we prove that $\{C, D\} \rightarrow \{B, A\} \in \Sigma^+$?

Proof.

1. Let $R = \{A, B, C, D\}$.
2. Let $\Sigma = \{\{A\} \to \{B\}, \{C\} \to \{A\}\}$.
3. We know that $\{A\} \to \{B\}$.
4. We know that $\{C\} \to \{A\}$.
5. Therefore $\{A\} \to \{A, B\}$ by Augmentation of (3) with $\{A\}$.
6. Therefore $\{C\} \to \{A, B\}$ by Transitivity of (4) and (5).
7. Therefore $\{C, D\} \to \{A, B, D\}$ by Augmentation of (6) with $\{D\}$.
8. Therefore $\{A, B, D\} \to \{A, B\}$ by Reflexivity with $\{A, B\} \subset \{A, B, D\}$.
9. Therefore $\{C, D\} \to \{A, B\}$ by Transitivity of (7) and (8).
10. Q.E.D

$\square$

Theorem

Let R be a relation with the set of functional dependencies Σ. We can compute Σ^+ by applying the Armstrong Axioms until no new functional dependency is produced.

There are other axioms (theorems)

Theorem

Weak Reflexivity is *sound*.
$\forall X \subset R \ (X \rightarrow \emptyset)$

Proof.

1. Let R be a relation schema.
2. Let $X \subset R$.
3. We know that $\emptyset \subset X$.
4. Therefore $X \rightarrow \emptyset$ by Reflexivity.
5. Q.E.D

$\square$

Theorem

Weak Augmentation is *sound*.

$$\forall X \subset R \; \forall Y \subset R \; \forall Z \subset R \; ((X \rightarrow Y) \Rightarrow (X \cup Z \rightarrow Y))$$

Proof.

1. Let R be a relation schema.
2. Let $X \subset R$.
3. Let $Y \subset R$.
4. Let $Z \subset R$.
5. Let $X \rightarrow Y$.
6. We know that $X \subset X \cup Z$.
7. Therefore $X \cup Z \rightarrow X$ by Reflexivity.
8. Therefore $X \cup Z \rightarrow Y$ by Transitivity of (7) and (5).
9. Q.E.D

$\square$

Definition

A set Σ of functional dependencies is minimal if and only if:

- The right hand-side of every functional dependency in Σ is minimal. Namely, every functional dependency is of the form $X \rightarrow \{A\}$.
- The left hand-side of every functional dependency is minimal. Namely, for every functional dependency in Σ of the form $X \rightarrow \{A\}$ there is no functional dependency $Y \rightarrow \{A\}$ in Σ^+ such that Y is a proper subset of X.
- The set itself is minimal. Namely, non of the functional dependency in Σ can derived from the other functional dependencies in Σ.

Consequently, a minimal cover of a set of functional dependencies Σ is set of functional dependencies Σ' that is both minimal and equivalent to Σ.

Theorem

Every set of functional dependencies has a minimal cover (or minimal basis).

An algorithm for the computation of the minimal cover Σ''' of a set of functional dependencies Σ has the following three steps.

1. Simplify (minimise) the right hand-side of every functional dependency in Σ to get Σ'.
2. Simplify (minimise) the left hand-side of every functional dependency in Σ' to get Σ''.
3. Simplify (minimise) the set Σ'' to get Σ'''.

The three steps have to be done in this order.

Definition

A set Σ of functional dependencies is compact if and only if there is no different functional dependencies with the same left-hand side.

$$\forall X \in R \; \forall Y \in R \; \forall Z \in R \; ((X \rightarrow Y \in \Sigma \; \wedge X \rightarrow Z \in \Sigma) \Rightarrow Y = Z))$$

The set of functional dependencies $\Sigma = \{\{A\} \rightarrow \{B\}, \{A\} \rightarrow \{C\}\}$ is not compact.

The set of functional dependencies $\Sigma = \{\{A\} \rightarrow \{B, C\}\}$ is compact.

Consequently, a compact cover of a set of functional dependencies Σ is set of functional dependencies Σ' that is both compact and equivalent to Σ.

Theorem

Every set of functional dependencies has a compact cover.

Consequently, a compact minimal cover (or canonical cover) of a set of functional dependencies Σ is set of functional dependencies Σ' that is both compact, minimal, and equivalent to Σ.

Theorem

Every set of functional dependencies has a compact minimal cover.

An algorithm for the computation of the compact minimal cover Σ'''' of a set of functional dependencies Σ has the following four steps:

1. Simplify (minimise) the right hand-side of every functional dependency in Σ to get Σ'.
2. Simplify (minimise) the left hand-side of every functional dependency in Σ' to get Σ''.
3. Simplify (minimise) the set Σ'' to get Σ'''.
4. Regroup all the functional dependencies with the same left-hand side in Σ''' to get Σ'''' (reverse of Step 1).

The four steps have to be done in this order.

$R = \{A, B, C, D, E\}$
$\Sigma = \{\{A, B\} \rightarrow \{C, D, E\}, \{A, C\} \rightarrow \{B, D, E\}, \{B\} \rightarrow \{C\}, \{C\} \rightarrow \{B\}, \{C\} \rightarrow \{D\}, \{B\} \rightarrow \{E\}, \{C\} \rightarrow \{E\}\}$

Compute the attribute closures.

Find all the candidate keys.

Find a minimal cover.

Find a compact minimal cover.

Compute all the singletons closures.

$$\{A\}^+ = \{A\}$$
$$\{B\}^+ = \{B, C, D, E\}$$
$$\{C\}^+ = \{B, C, D, E\}$$
$$\{D\}^+ = \{D\}$$
$$\{E\}^+ = \{E\}$$

Compute all the pairs closures.

$$\{A, B\}^+ = \{A, B, C, D, E\}$$
$$\{A, C\}^+ = \{A, B, C, D, E\}$$
$$\{A, D\}^+ = \{A, D\}$$
$$\{A, E\}^+ = \{A, E\}$$
$$\{B, C\}^+ = \{B, C, D, E\}$$
$$\{B, D\}^+ = \{B, C, D, E\}$$
$$\{B, E\}^+ = \{B, C, D, E\}$$
$$\{C, D\}^+ = \{B, C, D, E\}$$
$$\{C, E\}^+ = \{B, C, D, E\}$$
$$\{D, E\}^+ = \{D, E\}$$

Any set of attributes containing $\{A, B\}$ or $\{A, C\}$ is a superkey. $\{A, B\}$ and $\{A, C\}$ are candidate keys.

Compute all the remaining triplet closures.

$$\{A, D, E\}^+ = \{A, D, E\}$$
$$\{B, D, E\}^+ = \{B, C, D, E\}$$
$$\{C, D, E\}^+ = \{B, C, D, E\}$$
$$\{B, C, E\}^+ = \{B, C, D, E\}$$
$$\{B, C, D\}^+ = \{B, C, D, E\}$$

Compute all the remaining quadruplet closures.

$$\{B, C, D, E\}^+ = \{B, C, D, E\}$$

We know that all quintuplet covers are superkeys.

The two candidate keys are $\{A, B\}$ and $\{A, C\}$.

We compute a minimal cover.

$$
\begin{aligned}
&\Sigma = \{\\
&\{A, B\} \rightarrow \{C, D, E\},\\
&\{A, C\} \rightarrow \{B, D, E\},\\
&\{B\} \rightarrow \{C\},\\
&\{C\} \rightarrow \{B\},\\
&\{C\} \rightarrow \{D\},\\
&\{B\} \rightarrow \{E\},\\
&\{C\} \rightarrow \{E\}\}
\end{aligned}
$$

We simplify the right-hand sides (easy).

$$
\begin{aligned}
\Sigma' = \{ \\
&\{A, B\} \rightarrow \{C\}, \\
&\{A, B\} \rightarrow \{D\}, \\
&\{A, B\} \rightarrow \{E\}, \\
&\{A, C\} \rightarrow \{B\}, \\
&\{A, C\} \rightarrow \{D\}, \\
&\{A, C\} \rightarrow \{E\}, \\
&\{B\} \rightarrow \{C\}, \\
&\{C\} \rightarrow \{B\}, \\
&\{C\} \rightarrow \{D\}, \\
&\{B\} \rightarrow \{E\}, \\
&\{C\} \rightarrow \{E\}\}
\end{aligned}
$$

We simplify the left-hand sides (very difficult).

$\Sigma'' = \{$
$\cancel{\{A, B\} \rightarrow \{C\}}$, (is replaced with $\{B\} \rightarrow \{C\}$)
$\cancel{\{A, B\} \rightarrow \{D\}}$, (is replaced with) $\{B\} \rightarrow \{D\}$,
$\cancel{\{A, B\} \rightarrow \{E\}}$, (is replaced with $\{B\} \rightarrow \{E\}$)
$\cancel{\{A, C\} \rightarrow \{B\}}$, (is replaced with $\{C\} \rightarrow \{B\}$),
$\cancel{\{A, C\} \rightarrow \{D\}}$, (is replaced with $\{C\} \rightarrow \{D\}$),
$\cancel{\{A, C\} \rightarrow \{E\}}$, (is replaced with $\{C\} \rightarrow \{E\}$),
$\{B\} \rightarrow \{C\}$,
$\{C\} \rightarrow \{B\}$,
$\{C\} \rightarrow \{D\}$,
$\{B\} \rightarrow \{E\}$,
$\{C\} \rightarrow \{E\}\}$

We simplify the left-hand sides (very difficult).

$$\Sigma'' = \{$$
$$\{B\} \rightarrow \{D\},$$
$$\{B\} \rightarrow \{C\},$$
$$\{C\} \rightarrow \{B\},$$
$$\{C\} \rightarrow \{D\},$$
$$\{B\} \rightarrow \{E\},$$
$$\{C\} \rightarrow \{E\}\}$$

We simplify the set itself by removing functional dependencies that can be derived from the others. (difficult).

$\Sigma''' = \{$
$\cancel{\{B\} \rightarrow \{D\}}$, (it can be obtained from $\{B\} \rightarrow \{C\}$ and $\{C\} \rightarrow \{D\}$)
$\{B\} \rightarrow \{C\}$,
$\{C\} \rightarrow \{B\}$,
$\{C\} \rightarrow \{D\}$,
$\cancel{\{B\} \rightarrow \{E\}}$, (it can be obtained from $\{B\} \rightarrow \{C\}$ and $\{C\} \rightarrow \{E\}$)
$\{C\} \rightarrow \{E\}\}$

Σ''' is a minimal cover of Σ.

$$\Sigma''' = \{$$
$$\{B\} \rightarrow \{C\},$$
$$\{C\} \rightarrow \{B\},$$
$$\{C\} \rightarrow \{D\},$$
$$\{C\} \rightarrow \{E\}\}$$

We could reach different minimal covers by considering the constraints in a different order.

$$\Sigma''' = \{ \\ \{C\} \rightarrow \{B\}, \\ \{B\} \rightarrow \{C\}, \\ \{B\} \rightarrow \{D\}, \\ \{B\} \rightarrow \{E\}\}$$

$$\Sigma''' = \{ \\ \{C\} \rightarrow \{B\}, \\ \{B\} \rightarrow \{C\}, \\ \{B\} \rightarrow \{D\}, \\ \{C\} \rightarrow \{E\}\}$$

The algorithm always finds a minimal cover but some minimal covers may be unreachable with the algorithm.
For instance, if Σ is already a minimal cover, the algorithm cannot reach a different minimal cover even if it exists.
To be guaranteed to reach all minimal covers with the algorithm one needs to start from Σ^+.

We compute a compact minimal cover by regrouping the constraints with the same left-hand side (easy).

$$\Sigma''' = \{\{B\} \rightarrow \{C\}, \{C\} \rightarrow \{B, D, E\},$$

The other compact minimal covers are as follows.

$$\Sigma'''' = \{\{C\} \rightarrow \{B\}, \{B\} \rightarrow \{C, D, E\},$$

$$\Sigma'''' = \{\{B\} \rightarrow \{C, D\}, \{C\} \rightarrow \{B, E\},$$

$$\Sigma'''' = \{\{B\} \rightarrow \{C, E\}, \{C\} \rightarrow \{B, D\},$$