Explanation and Model Solution for “Assignment 5: Dependency Theory”

Jelle Hellings

3DB3: Databases – Fall 2021

Department of Computing and Software
McMaster University

Foreword

Each solution in this document consists of two parts. First, there is the original problem description (from the assignment). Next, there is the solution. As the solutions should be *explained* (readable), I have not included a rationale or further explanation of the answer in all cases.

Model Solution

1. Prove that $\{AB \longrightarrow C, A \longrightarrow D, CD \longrightarrow EF\} \models AB \longrightarrow F$ holds using only the Armstrong Axioms.

   Solution:

   (a) We use the Augmentation rule with $AB \longrightarrow C$ to obtain $AB \longrightarrow AC$.
   (b) We use the Augmentation rule with $A \longrightarrow D$ to obtain $AC \longrightarrow CD$.
   (c) We use the Transitivity rule with $AB \longrightarrow AC$ and $AC \longrightarrow CD$ to obtain $AB \longrightarrow CD$.
   (d) We use the Transitivity rule with $AB \longrightarrow CD$ and $CD \longrightarrow EF$ to obtain $AB \longrightarrow EF$.
   (e) We use the Reflexivity rule to obtain $EF \longrightarrow F$.
   (f) We use the Transitivity rule with $AB \longrightarrow EF$ and $EF \longrightarrow F$ to obtain $AB \longrightarrow F$.

   Remark. Consider $\alpha \longrightarrow \beta$. Normally, both α and β are sets of attributes. Hence, adding an attribute to α that is already in α doesn't do anything. For example, if we add A with the Augmentation rule to $AB \longrightarrow C$, then the result would be (if we write out sets explicitly) $\{A\} \cup \{A, B\} \longrightarrow \{A\} \cup \{C\}$, which is equivalent to $AB \longrightarrow AC$.

2. Prove the soundness of the following inference rule directly from the definition of functional dependencies (without using any inference rules):

$$\text{if } X \longrightarrow Y \text{ and } YW \longrightarrow Z, \text{ then } XW \longrightarrow Z.$$

   Solution:

   Let $\mathbf{R}$ be any relational schema that satisfies both $X \longrightarrow Y$ and $YW \longrightarrow Z$. By definition, relational schema $\mathbf{R}$ satisfies $XW \longrightarrow Z$ if we have $r_1[XW] = r_2[XW] \implies r_1[Z] = r_2[Z]$ for every instance $\mathcal{I}$ of $\mathbf{R}$ and every pair of rows $r_1, r_2 \in \mathcal{I}$.

   Assume we have rows $r_1, r_2 \in \mathcal{I}$ of instance $\mathcal{I}$ of $\mathbf{R}$ with $r_1[XW] = r_2[XW]$.

- By $r_1[XW] = r_2[XW]$, we conclude $r_1[X] = r_2[X]$.
- By $r_1[X] = r_2[X]$ and $X \longrightarrow Y$, we conclude $r_1[Y] = r_2[Y]$.
- By $r_1[XW] = r_2[XW]$ and $r_1[Y] = r_2[Y]$, we conclude $r_1[YW] = r_2[YW]$.
- By $r_1[YW] = r_2[YW]$ and $YW \longrightarrow Z$, we conclude $r_1[Z] = r_2[Z]$.

3. Prove the soundness of the following inference rule for inclusion dependencies:

$$\text{if } R[X] \subseteq S[Y] \text{ and } S[Y] \subseteq T[Z], \text{ then } R[X] \subseteq T[Z].$$

Solution:

Assume that $R[X] \subseteq S[Y]$ and $S[Y] \subseteq T[Z]$ hold on all instances of R, S, and T. We must prove that for every instance $\mathcal{I}_1$ of R and every row $r_1 \in \mathcal{I}_1$, there exists a row $r_2 \in \mathcal{I}_2$ of T with $r_1[X] = r_2[Z]$.

Assume we have rows $r_1 \in \mathcal{I}_1$ of instance $\mathcal{I}_1$ of R

- By $R[X] \subseteq S[Y]$ and r_1, there exists a row $r' \in \mathcal{I}'$ of S such that $r_1[X] = r'[Y]$.
- By $S[Y] \subseteq T[Z]$ and r', there exists a row $r_2 \in \mathcal{I}_2$ of T such that $r'[Y] = r_2[Z]$.
- By $r_1[X] = r'[Y]$ and $r'[Y] = r_2[Z]$, we conclude $r_1[X] = r_2[Z]$.

4. Prove the soundness of the following inference rule

$$\text{if } X \twoheadrightarrow Y \text{ and } XY \longrightarrow Z, \text{ then } X \longrightarrow Z \setminus (X \cup Y).$$

Solution:

Alternative 1 (via definitions). Let $\mathbf{R}$ be any relational schema that satisfies both $X \twoheadrightarrow Y$ and $XY \longrightarrow Z$. By definition, relational schema $\mathbf{R}$ satisfies $X \longrightarrow Z \setminus (X \cup Y)$ if we have $r_1[X] = r_2[X] \implies r_1[Z \setminus (X \cup Y)] = r_2[Z \setminus (X \cup Y)]$ for every instance $\mathcal{I}$ of $\mathbf{R}$ and every pair of rows $r_1, r_2 \in \mathcal{I}$.

Assume we have rows $r_1, r_2 \in \mathcal{I}$ of instance $\mathcal{I}$ of $\mathbf{R}$ with $r_1[X] = r_2[X]$.

- By $r_1[X] = r_2[X]$ and $X \twoheadrightarrow Y$, there exists a row r_3 with $r_1[XY] = r_3[XY]$ and $r_2[XA] = r_3[XA]$ (with A all attributes of $\mathbf{R}$ not in X and Y).
- By $r_1[XY] = r_3[XY]$ and $XY \longrightarrow Z$, we conclude $r_1[Z] = r_3[Z]$.
- By $r_1[Z] = r_3[Z]$, we conclude $r_1[Z \setminus (X \cup Y)] = r_3[Z \setminus (X \cup Y)]$
- By $r_2[XA] = r_3[XA]$, and $(Z \setminus (X \cup Y)) \subseteq A$, we have $r_2[Z \setminus (X \cup Y)] = r_3[Z \setminus (X \cup Y)]$.
- By $r_1[Z \setminus (X \cup Y)] = r_3[Z \setminus (X \cup Y)]$ and $r_2[Z \setminus (X \cup Y)] = r_3[Z \setminus (X \cup Y)]$, we conclude $r_1[Z \setminus (X \cup Y)] = r_2[Z \setminus (X \cup Y)]$.

Alternative 2 (using inference rules).

(a) We use the Complementation rule with $X \twoheadrightarrow Y$ to obtain $X \twoheadrightarrow A$ with A all attributes not in $X \cup Y$.

(b) Note that, by construction of A, we have $A \cap (X \cup Y) = \emptyset$. Hence, $(Z \setminus (X \cup Y)) \subseteq A$.

(c) We use Reflexivity to obtain $Z \longrightarrow Z \setminus (X \cup Y)$.

(d) We use Transitivity on $XY \longrightarrow Z$ and $Z \longrightarrow Z \setminus (X \cup Y)$ to obtain $XY \longrightarrow Z \setminus (X \cup Y)$.

(e) We use the Coalescence rule with $X \twoheadrightarrow A$ and $XY \longrightarrow Z \setminus (X \cup Y)$ to obtain $X \longrightarrow Z \setminus (X \cup Y)$.

5. Prove that the following inference rule *is not sound*:

$$\text{if } XW \longrightarrow Y \text{ and } XY \longrightarrow Z, \text{ then } X \longrightarrow Z.$$

Hint: Look for a counterexample by constructing a table in which $XW \longrightarrow Y$ and $XY \longrightarrow Z$ hold, but $X \longrightarrow Z$ does not hold.

Solution:

We construct an instance in which $XW \longrightarrow Y$ and $XY \longrightarrow Z$ hold, but $X \longrightarrow Z$ does not hold:

X	W	Y	Z
0	1	0	0
0	0	1	1
0	1	0	0
0	0	1	1

.

D1 A completeness proof for Armstrong

Consider the following inference rules. Consider the following inference rules.

R1. *Reflexivity.* If $Y \subseteq X$, then $X \longrightarrow Y$.

R2. *Augmentation.* If $X \longrightarrow Y$ then $XZ \longrightarrow YZ$ for any Z.

R3. *Transitivity.* If $X \longrightarrow Y$ and $Y \longrightarrow Z$, then $X \longrightarrow Z$.

6. Prove that the inference rules R1, R2, and R3 are *complete*: prove that if $\mathfrak{S} \models X \longrightarrow Y$ holds for some set of functional dependencies $\mathfrak{S}$, then we can derive $X \longrightarrow Y$ from $\mathfrak{S}$ using only the inference rules R1, R2, and R3.

Hint: Use the fact that the Closure algorithm is complete. Can you prove that any *sound* derivation made by the Closure algorithm can also be derived using the inference rules R1–R3? You may use the Union rule and the Decomposition rule (as we have derived them using the inference rules R1–R3).

Solution:

As the Closure algorithm is sound and complete, we have $\mathfrak{S} \models X \longrightarrow Y$ if and only if $Y \subseteq \text{Closure}(\mathfrak{S}, X)$. Hence, we can prove that the Armstrong Axioms can derive $\mathfrak{S} \models X \longrightarrow Y$ by proving that the Armstrong Axioms can derive $X \longrightarrow Y$ whenever $Y \subseteq \text{Closure}(\mathfrak{S}, X)$. To do so, we are going to augment the soundness proof of the Closure algorithm by proving the following additional loop invariant:

Invariant: If $Y \subseteq$ *closure*, then $\mathfrak{S} \models X \longrightarrow Y$ can be derived using inference rules R1–R3.

First, we prove that the invariant holds initially (after Line 1 of the Closure algorithm). In that case, we have $Y \subseteq$ *closure* $= X$. Hence, $Y \subseteq X$ and by *reflexivity* (R1), we have $\mathfrak{S} \models X \longrightarrow Y$.

Next, we prove that the invariant holds after after every iteration of the **while** loop (after Line 3 of the Closure algorithm). Let C be the attributes in *closure* before Line 3. Hence, we must prove that, for every $Y \subseteq (C \cup B)$, we can derive $\mathfrak{S} \models X \longrightarrow Y$ using inference rules R1–R3. Let $Y_C = Y \cap C$ and $Y_B = Y \cap B$. As the invariant holds before Line 3, there is a derivation of $\mathfrak{S} \models X \longrightarrow Y_C$ using inference rules R1–R3. Next, we consider the derivation of $\mathfrak{S} \models X \longrightarrow Y_B$ using inference rules R1–R3. Consider the functional dependency $A \longrightarrow B$ picked at Line 2 for which $A \subseteq C$ and $B \nsubseteq C$ holds. As $A \subseteq C$, there is a derivation of $\mathfrak{S} \models X \longrightarrow A$ using inference rules R1–R3. Using *transitivity* (R3) on $X \longrightarrow A$ and $A \longrightarrow B$, we derive $X \longrightarrow B$. Then, as $Y_B \subseteq B$, we use the *Decomposition rule* on $X \longrightarrow B$, to derive $X \longrightarrow Y_B$. Finally, using the *Union rule* on $X \longrightarrow Y_C$ and $X \longrightarrow Y_B$, we derive $X \longrightarrow Y$.

When the algorithm terminates, we have $Y \subseteq$ *closure* if and only if $\mathfrak{S} \models X \longrightarrow Y$ (as the Closure algorithm is sound and complete) and $\mathfrak{S} \models X \longrightarrow Y$ can be derived using inference rules R1–R3 (by the invariant), completing the proof.

D2 The human computer

Consider the relational schema $\mathbf{r}(A, B, C, D, E)$ and the following set of functional dependencies:

$$\mathfrak{S} = \{AB \longrightarrow D, AC \longrightarrow E, BC \longrightarrow D, C \longrightarrow A, D \longrightarrow A, E \longrightarrow B\}.$$

Answer the following questions:

7. Provide the attribute closure of set of attributes C (hence, C^+) and of set of attributes EA (hence, $(EA)^+$) with respect to $\mathfrak{S}$. Explain your steps.

   Solution:
   We run the closure algorithm:

   - Closure($\mathfrak{S}, \{C\}$). We initialize *closure* = $\{C\}$. Then we can use $C \longrightarrow A$ and add A to *closure*, resulting in *closure* = $\{A, C\}$. Then we can use $AC \longrightarrow E$ and add E to *closure*, resulting in *closure* = $\{A, C, E\}$. Then we can use $E \longrightarrow B$ and add B to *closure*, resulting in *closure* = $\{A, B, C, E\}$. Then we can use $AB \longrightarrow D$ and add D to *closure*, resulting in *closure* = $\{A, B, C, D, E\}$. At this point, we cannot use the remaining functional dependencies to add new attributes, as we already have all attributes. Hence, the result is $\{A, B, C, D, E\}$.
   - Closure($\mathfrak{S}, \{A, E\}$). We initialize *closure* = $\{A, E\}$. Then we can use $E \longrightarrow B$ and add B to *closure*, resulting in *closure* = $\{A, B, E\}$. Then we can use $AB \longrightarrow D$ and add D to *closure*, resulting in *closure* = $\{A, B, D, E\}$. At this point, we cannot use the remaining functional dependencies to add new attributes: $D \longrightarrow A$ would not add new attributes, while all other remaining functional dependencies require that C is in *closure* (as they are of the form $X \longrightarrow Y$ with $C \in X$). Hence, the result is $\{A, B, D, E\}$.

8. Compute the closure $\mathfrak{S}^+$. Explain your steps. Based on the closure, indicate which attributes are *superkeys* and which attributes are *keys*.

   Hint: A *superkey* is a set of attributes that determines *all* attributes from $\mathbf{r}$. A *key* k is a superkey of minimal size (if we remove any attribute from k, it is no longer a key).

   Solution:
   Recall that $\mathfrak{S}^+ = \{X \longrightarrow Y \mid \mathfrak{S} \models X \longrightarrow Y\}$. To compute all $X \longrightarrow Y$ for a given $X \subseteq \{A, B, C, D, E\}$, we compute X^+ via the attribute closure algorithm (Closure($\mathfrak{S}, X$)). Then, for all $Y \subseteq X^+$, the functional dependency $X \longrightarrow Y$ holds. In the next table, we will denote all closures of attribute sets.

   Remark. You can compute the transitive closure of all cases independently, but you can also take safe shortcuts. As a first shortcut, to *start* the computation of the transitive closure $(\alpha \cup \beta)^+$, we can start with $\alpha^+ \cup \beta^+$ instead of $\alpha \cup \beta$. E.g., $A^+ \cup E^+ = \{A, B, E\} \subseteq AE^+$. As a second shortcut, we note that we have $(\alpha \cup \beta)^+ = \alpha^+$ whenever $\beta \subseteq \alpha^+$. E.g., $\{B\} \subseteq E^+$. Hence, $\{B, E\}^+ = E^+$.

Attribute(s)	Closure	Comment
A	$\{A\}$	Only $A \longrightarrow A$.
B	$\{B\}$	Only $B \longrightarrow B$.
C	$\{A, B, C, D, E\}$	This is a key! We have $C \longrightarrow Y$ for all $Y \subseteq \{A, B, C, D, E\}$.
D	$\{A, D\}$	We have $D \longrightarrow A$, $D \longrightarrow D$, and $D \longrightarrow AD$.
E	$\{B, E\}$	We have $E \longrightarrow B$, $E \longrightarrow E$, and $E \longrightarrow EB$.
AB	$\{A, B, D\}$	We have $AB \longrightarrow Y$ for all $Y \subseteq \{A, B, D\}$.
AC	$\{A, B, C, D, E\}$	A superkey. We have $AC \longrightarrow Y$ for all $Y \subseteq \{A, B, C, D, E\}$.
AD	$\{A, D\}$	We have $AD \longrightarrow A$, $AD \longrightarrow D$, and $AD \longrightarrow AD$.
AE	$\{A, B, D, E\}$	We have $AE \longrightarrow Y$ for all $Y \subseteq \{A, B, D, E\}$
BC	$\{A, B, C, D, E\}$	A superkey. We have $BC \longrightarrow Y$ for all $Y \subseteq \{A, B, C, D, E\}$.
BD	$\{A, B, D\}$	We have $BD \longrightarrow Y$ for all $Y \subseteq \{A, B, D\}$.
BE	$\{B, E\}$	We have $E \longrightarrow B$, $E \longrightarrow E$, and $E \longrightarrow EB$.
CD	$\{A, B, C, D, E\}$	A superkey. We have $CD \longrightarrow Y$ for all $Y \subseteq \{A, B, C, D, E\}$.
CE	$\{A, B, C, D, E\}$	A superkey. We have $CE \longrightarrow Y$ for all $Y \subseteq \{A, B, C, D, E\}$.
DE	$\{A, B, D, E\}$	We have $DE \longrightarrow Y$ for all $Y \subseteq \{A, B, D, E\}$
ABC	$\{A, B, C, D, E\}$	A superkey. We have $ABC \longrightarrow Y$ for all $Y \subseteq \{A, B, C, D, E\}$.
ABD	$\{A, B, D\}$	We have $ABD \longrightarrow Y$ for all $Y \subseteq \{A, B, D\}$.
ABE	$\{A, B, D, E\}$	We have $ABE \longrightarrow Y$ for all $Y \subseteq \{A, B, D, E\}$
ACD	$\{A, B, C, D, E\}$	A superkey. We have $ACD \longrightarrow Y$ for all $Y \subseteq \{A, B, C, D, E\}$.
ACE	$\{A, B, C, D, E\}$	A superkey. We have $ACE \longrightarrow Y$ for all $Y \subseteq \{A, B, C, D, E\}$.
ADE	$\{A, B, D, E\}$	We have $ADE \longrightarrow Y$ for all $Y \subseteq \{A, B, D, E\}$
BCD	$\{A, B, C, D, E\}$	A superkey. We have $BCD \longrightarrow Y$ for all $Y \subseteq \{A, B, C, D, E\}$.
BCE	$\{A, B, C, D, E\}$	A superkey. We have $BCE \longrightarrow Y$ for all $Y \subseteq \{A, B, C, D, E\}$.
BDE	$\{A, B, D, E\}$	We have $BDE \longrightarrow Y$ for all $Y \subseteq \{A, B, D, E\}$
CDE	$\{A, B, C, D, E\}$	A superkey. We have $CDE \longrightarrow Y$ for all $Y \subseteq \{A, B, C, D, E\}$.
ABCD	$\{A, B, C, D, E\}$	A superkey. We have $ABCD \longrightarrow Y$ for all $Y \subseteq \{A, B, C, D, E\}$.
ABCE	$\{A, B, C, D, E\}$	A superkey. We have $ABCE \longrightarrow Y$ for all $Y \subseteq \{A, B, C, D, E\}$.
ABDE	$\{A, B, D, E\}$	We have $ABDE \longrightarrow Y$ for all $Y \subseteq \{A, B, D, E\}$
ACDE	$\{A, B, C, D, E\}$	A superkey. We have $ACDE \longrightarrow Y$ for all $Y \subseteq \{A, B, C, D, E\}$.
BCDE	$\{A, B, C, D, E\}$	A superkey. We have $BCDE \longrightarrow Y$ for all $Y \subseteq \{A, B, C, D, E\}$.
ABCDE	$\{A, B, C, D, E\}$	A superkey. We have $ABCDE \longrightarrow Y$ for all $Y \subseteq \{A, B, C, D, E\}$.

9. Provide a minimal cover for $\mathfrak{S}$. Explain your steps.

Solution:

We start with $\mathfrak{S} = \{AB \longrightarrow D, AC \longrightarrow E, BC \longrightarrow D, C \longrightarrow A, D \longrightarrow A, E \longrightarrow B\}$.

As the first step, we make all functional dependencies *minimalistic* by breaking up any functional dependencies of the form $\alpha \longrightarrow \beta$ with α and β sets of functional dependencies and $\beta = \{\beta_1, \ldots, \beta_n\}$ into functional dependencies $\alpha \longrightarrow \beta_1, \ldots, \alpha \longrightarrow \beta_n$. As all functional dependencies are already of the latter form, this step does not change the set of functional dependencies.

Next, we determine whether we can remove functional dependencies.

- First, we try to remove $AB \longrightarrow D$. We can do so only if $\mathfrak{S}_{\neg AB \longrightarrow D} \models AB \longrightarrow D$ holds with $\mathfrak{S}_{\neg AB \longrightarrow D} = \{AC \longrightarrow E, BC \longrightarrow D, C \longrightarrow A, D \longrightarrow A, E \longrightarrow B\}$. To test whether $\mathfrak{S}_{\neg AB \longrightarrow D} \models AB \longrightarrow D$ holds, we compute the closure $\textsc{Closure}(\mathfrak{S}_{\neg AB \longrightarrow D}, AB) = \{A, B\}$. As $D \notin \textsc{Closure}(\mathfrak{S}_{\neg AB \longrightarrow D}, AB)$, we conclude that $\mathfrak{S}_{\neg AB \longrightarrow D} \models AB \longrightarrow D$ does not hold. Hence, we cannot remove $AB \longrightarrow D$.
- Next, we try to remove $AC \longrightarrow E$. We do so analogous to the previous step: as $E \notin \textsc{Closure}(\mathfrak{S}_{\neg AC \longrightarrow E}, AC) = \{A, C\}$ with $\mathfrak{S}_{\neg AC \longrightarrow E} = \{AB \longrightarrow D, BC \longrightarrow D, C \longrightarrow A, D \longrightarrow$

$A, E \longrightarrow B\}$, we conclude that we cannot remove $AC \longrightarrow E$.

Remark. We can *replace* $AC \longrightarrow E$, however, by a simpler functional dependency. Due to $C \longrightarrow A$ and $AC \longrightarrow E$, we also have $C \longrightarrow E$, and we have $\{C \longrightarrow A, C \longrightarrow E\} \models AC \longrightarrow E$. Hence, we can replace $AC \longrightarrow E$ by the simpler $C \longrightarrow E$. This simplification is *not required*, however!

- Next, we try to remove $BC \longrightarrow D$. We already have $\{AB \longrightarrow D, C \longrightarrow A\} \models BC \longrightarrow D$. Indeed, we have $D \in \textsc{Closure}(\mathfrak{S}_{\neg BC \longrightarrow D}, BC\} = \{A, B, C, D, E\}$ with $\mathfrak{S}_{\neg BC \longrightarrow D} = \{AB \longrightarrow D, AC \longrightarrow E, C \longrightarrow A, D \longrightarrow A, E \longrightarrow B\}$. Hence, we conclude that we *can* and *will* remove $BC \longrightarrow D$!
- Next, we try to remove $C \longrightarrow A$. We already removed $BC \longrightarrow D$. Hence, we can *additionally* remove $C \longrightarrow A$ only if $\mathfrak{S}_{\neg(BC \longrightarrow D, C \longrightarrow A)} \models C \longrightarrow A$ holds with $\mathfrak{S}_{\neg(BC \longrightarrow D, C \longrightarrow A)} = \{AB \longrightarrow D, AC \longrightarrow E, D \longrightarrow A, E \longrightarrow B\}$. Analogous to the first step, we compute the closure $\textsc{Closure}(\mathfrak{S}_{\neg(BC \longrightarrow D, C \longrightarrow A)}, C)$ to determine whether $\mathfrak{S}_{\neg(BC \longrightarrow D, C \longrightarrow A)} \models C \longrightarrow A$ holds. We have $A \notin \textsc{Closure}(\mathfrak{S}_{\neg(BC \longrightarrow D, C \longrightarrow A)}, C) = \{C\}$. Hence, we conclude that we cannot remove $C \longrightarrow A$.
- Next, we try to remove $D \longrightarrow A$. We do so analogous to the previous step: as $A \notin \textsc{Closure}(\mathfrak{S}_{\neg(BC \longrightarrow D, D \longrightarrow A)}, D) = \{D\}$ with $\mathfrak{S}_{\neg(BC \longrightarrow D, D \longrightarrow A)} = \{AB \longrightarrow D, AC \longrightarrow E, C \longrightarrow A, E \longrightarrow B\}$, we conclude that we cannot remove $D \longrightarrow A$.
- Finally, we try to remove $E \longrightarrow B$. We do so analogous to the previous two steps: as $B \notin \textsc{Closure}(\mathfrak{S}_{\neg(BC \longrightarrow D, E \longrightarrow B)}, E) = \{E\}$ with $\mathfrak{S}_{\neg(BC \longrightarrow D, E \longrightarrow B)} = \{AB \longrightarrow D, AC \longrightarrow E, C \longrightarrow A, D \longrightarrow A\}$, we conclude that we cannot remove $E \longrightarrow B$.

The above steps yield the minimal cover:

$$AB \longrightarrow D, AC \longrightarrow E, C \longrightarrow A, D \longrightarrow A, E \longrightarrow B.$$

Remark. With further simplifications, the minimal cover would be

$$AB \longrightarrow D, C \longrightarrow E, C \longrightarrow A, D \longrightarrow A, E \longrightarrow B.$$