Data Management 1 Proofs (ACSAI)

Gianmaria Romano

A.Y. 2024/2025 (First Semester)

Disclaimer

This document contains the proofs given by Professor Perelli for the "Data Management and Analysis Unit 1" course for the "Applied Computer Science and Artificial Intelligence" course at Sapienza Università di Roma.
Keep in mind, however, that these notes do **not** replace the course material as they are meant to be used for preparation to the oral exam, meaning that it is still suggested to check the Professor's resources as well.
These notes are free to use/share but please remember to credit me as the author and therefore do not hide/remove this page.

Contents

Chapter 1

Relational Theory

1.1 Armstrong's rules

1.1.1 Union rule

If $X \to Y \in F^A$ and $X \to Z \in F^A$, then $X \to YZ \in F^A$.
For proof, notice that, if $X \to Y \in F^A$, then $X \to XY \in F^A$ by augmentation with respect to X, and, similarly, if $Y \to Z \in F^A$, then $XY \to YZ \in F^A$ by augmentation with respect to Y.
Therefore, it is possible to conclude by transitivity that $X \to YZ \in F^A$.

1.1.2 Decomposition rule

If $X \to Y \in F^A$ and $Z \subseteq Y$, then $X \to Z \in F^A$.
For proof, if $Z \subseteq Y$, then $Y \to Z \in F^A$ by reflexivity, meaning that, since it is assumed that $X \to Y \in F^A$, it is possible to obtain $X \to Z \in F^A$ by transitivity.

1.1.3 Pseudotransitivity rule

If $X \to Y \in F^A$ and $WY \to Z \in F^A$, then $WX \to Z \in F^A$.
For proof, if $X \to Y \in F^A$, then $WX \to WY \in F^A$ by augmentation with respect to W, meaning that, since $WY \to Z \in F^A$ by assumption, it is possible to conclude that $WX \to Z \in F^A$ by transitivity.

1.2 Closure lemma

Given a relational schema R and a set F of functional dependencies, it is possible to state that $X \to Y \in F^A$ if and only if $Y \subseteq X_F^+$.
If $X \to Y \in F^A$, then, by decomposition rule, $X \to A \in F^A \ \forall \ A \in Y$, implying, by definition of X_F^+, $A \subseteq X_F^+$, meaning that $Y \subseteq X_F^+$ as this holds $\forall \ A \in Y$.

On the other hand, if $Y \subseteq X_F^+$, it is possible to obtain $A \subseteq X_F^+$, meaning that, by definition of X_F^+, $X \to A \in F^A$, allowing to conclude that, by union rule, $X \to Y \in F^A$.

1.3 $F^A = F^+$

For every relational schema R and a set F of functional dependencies, it holds that $F^A = F^+$.
Since the goal is to prove that two sets are equal, the theorem will be proved by double inclusion, meaning that both $F^A \subseteq F^+$ and $F^+ \subseteq F^A$ must hold true.

- To prove that $F^A \subseteq F^+$, it is possible to proceed by induction: if F_i^A denotes the set of functional dependencies obtained by applying Armstrong's axioms at most i times, then it must hold that $F^A = F_0^A \cup F_1^A \cup \cdots \cup F_n^A$, allowing to state the following:
  For the base case, given by $i = 0$, if $X \to Y \in F_0^A$, then $X \to Y$ was obtained by applying Armstrong's axioms at most zero times, meaning that, actually, $F_0^A = F$ and therefore $X \to Y \in F$, which implies that $X \to Y \in F^+$ as well since $F \subseteq F^+$ trivially.
  Supposing the statement is true for some i, the goal is now to prove that it is true for $i+1$ as well: if $X \to Y \in F_{i+1}^A$, then $X \to Y$ was obtained by applying Armstrong's axioms at most $i+1$ times from one of the following cases:

  1. **Reflexivity**
     Using reflexivity implies that $Y \subseteq X \Rightarrow X \to Y \in F^A$ and, since this is a trivial dependence, it is possible to state that, for every tuple t of every legal instance r of R, $t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y]$ and therefore $X \to Y \in F^+$.
  2. **Augmentation**
     By augmentation, $\exists\ V \to W \in F_i^A$ such that $VZ \to WZ \in F_{i+1}^A$ for every attribute Z of R.
     Let $X = VZ$ and $Y = WZ$ and notice that it is possible to obtain:

     $$t_1[X] = t_2[X] \Rightarrow t_1[VZ] = t_2[VZ] \Rightarrow (t_1[V] = t_2[V]) \wedge (t_1[Z] = t_2[Z])$$

     However, since $V \to W \in F_i^A$, it must hold by inductive hypothesis that $V \to W \in F^+$, allowing to state that, for every legal instance r or R, the following must be true:

     $$(t_1[V] = t_2[V]) \wedge (t_1[Z] = t_2[Z]) \Rightarrow (t_1[W] = t_2[W]) \wedge (t_1[Z] = t_2[Z])$$

     Most particularly, since $WZ = Y$, it is possible to apply union:

     $$(t_1[W] = t_2[W]) \wedge (t_1[Z] = t_2[Z]) \Rightarrow t_1[WZ] = t_2[WZ] \Rightarrow t_1[Y] = t_2[Y]$$

     Therefore, by applying transitivity, it is possible to conclude that $t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y]$, meaning that, indeed, $X \to Y \in F^+$.

3. **Transitivity**
By transitivity, $\exists\ X \to Z,\ Z \to Y \in F_i^A$ such that $X \to Y \in F_{i+1}^A$.
By inductive hypothesis, it is possible to use $X \to Z,\ Z \to Y \in F^+$ to state that:

$$t_1[X] = t_2[X] \Rightarrow t_1[Z] = t_2[Z] \Rightarrow t_1[Y] = t_2[Y]$$

Indeed, it is possible to recover $t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y]$ and conclude that $X \to Y \in F^+$.

Therefore, since the statement is true for $i+1$ as well, it must hold that $F^A \subseteq F^+$.

- To prove instead that $F^+ \subseteq F^A$, it is possible to proceed via closure lemma.
Given $X \to Y \in F^+$, consider a legal instance r of R formed by just two tuples, t_1 and t_2, which take the same value in X_F^+ and different values in $R - X_F^+$.
Since r is indeed a legal instance by construction, pick a generic functional dependence $V \to W \in F$, meaning that $V \to W \in F^A$ as well by base point axiom.
Most particularly, observe by construction that $t_1[V] = t_2[V]$ if and only if $V \subseteq X_F^+$, meaning that $X \to V \in F^A$ by closure lemma, allowing to obtain, via transitivity, that $X \to W \in F^A$ as well, also implying that $W \subseteq X_F^+$ by closure lemma, resulting in $t_1[W] = t_2[W]$ by construction.
Therefore, it is possible to conclude that $t_1[V] = t_2[V] \Rightarrow t_1[W] = t_2[W]$ for any generic functional dependence $V \to W \in F$, meaning that, indeed $V \to W \in F^A$.
Similarly, it is possible to show that, since r is a legal instance of R, then $X \to Y \in F^+$ must be satisfied as well, meaning that it must hold that $t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y]$: since it trivially holds, by definition, that $X \subseteq X_F^+$, it is also possible to recover $Y \subseteq X_F^+$, allowing to conclude that $X \to Y \in F^A$ via closure lemma.
This allows to show that $F^+ \subseteq F^A$.

Therefore, since both statements have been proved, it is possible to conclude that $F^A = F^+$.

1.4 Closure of X with respect to F (first version)

The closure algorithm always ends, meaning that there must be a finite index f such that $Z_f = Z_{f+1} = Z_f \cup S_f$, meaning that $S_f \subseteq Z_f$: if the statement is true, the algorithm is correct and $Z_f = X_F^+$.
Since the goal is to prove that two sets are equal, the theorem will be proved by double inclusion, meaning that both $Z_f \subseteq X_F^+$ and $X_F^+ \subseteq Z_f$ must hold true.

- To prove that $Z_f \subseteq X_F^+$, it is possible to proceed by induction over the number of iterations.

For the base case, given by $i = 0$, observe that, actually, $Z_0 = X$, meaning that $Z_0 \subseteq X_F^+$ is trivially true by definition of X_F^+.
Supposing the statement is true for some i, the goal is now to prove that it is true for $i+1$ as well: given $A \in Z_{i+1} = Z_i \cup S_i$, there are two possible cases:

1. $A \in Z_i$
   By inductive hypothesis, $A \in Z_i \subseteq X_F^+$, meaning that $A \in X_F^+$ as well.

2. $A \in S_i$
   By definition of S_i, $\exists\, Y \to V \in F$ such that $Y \in Z_i$ and $A \in V$.
   By inductive hypothesis, $Z_i \subseteq X_F^+$ implies that $Y \subseteq X_F^+$, meaning that, by closure lemma, it must hold that $X \to Y \in F^A$, and, since $F \subseteq F^A$ by base point axiom, it is possible to state that $Y \to V \in F \Rightarrow Y \to V \in F^A$, it is possible to retrieve by transitivity that $X \to V \in F^A$: this implies, by closure lemma, that $V \subseteq X_F^+$, but, since $A \in V$, it is possible to conclude that $A \in X_F^+$ as well.

Therefore, since the statement is true for $i+1$ as well, it must hold that $Z_f \subseteq X_F^+$.

- To prove instead that $X_F^+ \subseteq Z_f$, consider an instance r or R with two tuples t_1 and t_2 such that $t_1[Z_f] = t_2[Z_f]$ but $t_1[R - Z_f] \neq t_2[R - Z_f]$.
  Pick a generic $Y \to V \in F$ and assume $t_1[Y] = t_2[Y]$: this implies that, by construction of the instance, $Y \subseteq Z_f$, meaning that, by applying the closure of X algorithm, it must hold that $B \in S_f \subseteq Z_f\ \forall\, B \in V$, resulting in $B \in Z_f \Rightarrow V \in Z_f$, and therefore, by construction of the instance, $t_1[V] = t_2[V]$.
  Since it is possible to recover $t_1[Y] = t_2[Y] \Rightarrow t_1[V] = t_2[V]\ \forall\, Y \to V \in F$, the instance is indeed legal, meaning that, by closure lemma, it is possible to state that $A \in X_F^+ \Leftrightarrow X \to A \in F^A = F^+$, with $X = Z_0 \subseteq Z_f$ by algorithm design and $A \in Z_f$ by legality of the instance.
  This allows to show that $X_F^+ \subseteq Z_f$.

Therefore, since both statements have been proved, it is possible to conclude that $Z_f = X_F^+$.

1.5 Closure inclusion lemma

Let F and G be two sets of functional dependencies over the same relational schema R: if $F \subseteq G^+$, then $F^+ \subseteq G^+$.
For proof, let $F \to F'$ denote the fact that every functional dependence in F' can be obtained by applying Armstrong's axioms on functional dependencies in F (trivially, $F \to F^+$), which is true if and only if $F' \subseteq F^+$: this means that $F \subseteq G^+$ if and only if $G \to F \to F^+$, allowing to recover that $G \to F^+$, also implying that $F^+ \subseteq G^+$.

N.B.: This lemma is part of the proof for the equivalence between two sets of functional dependencies F and G ($F \equiv G \Leftrightarrow F^+ = G^+$), which is used to check whether a decomposition ρ of a relational schema R preserves F.

1.6 Closure of X with respect to G (second version)

Since this algorithm is an alternative template to find the closure of a set X with respect to a set G of functional dependencies, the algorithm is correct as well, with $Z_f = X_G^+$.
Again, since the goal is to prove that two sets are equal, the theorem will be proved by double inclusion, meaning that both $Z_f \subseteq X_G^+$ and $X_G^+ \subseteq Z_f$ must hold true.

- To prove that $Z_f \subseteq X_G^+$, it is possible to proceed by induction over the number of iterations.
  For the base case, given by $i = 0$, observe that, actually, $Z_0 = X \subseteq X_G^+$ is trivially true by definition of X_G^+.
  Supposing the statement is true for some i, the goal is now to prove that it is true for $i+1$ as well: given $A \in Z_{i+1} = Z_i \cup S_i$, there are two possible cases:

  1. $A \in Z_i$
     By inductive hypothesis, $A \in Z_i \subseteq Z_G^+$, meaning that $A \in X_G^+$ as well.
  2. $A \in S_i$
     By definition of $S = \bigcup_{j=1}^{k}((Z_i \cap R_j)_F^+ \cap R_j)$, there must exist a subschema R_j such that the following is true:

     $$A \in (Z_i \cap R_j)_F^+ \cap R_j \Leftrightarrow A \in (Z_i \cap R_j)_F^+ \wedge A \in R_j$$

     Therefore, by closure lemma, $A \in (Z_i \cap R_j)_F^+ \Leftrightarrow (Z_i \cap R_j) \to A \in F^+$, but, since $Z_i \cap R_j \subseteq R_j$ and $A \subseteq R_j$, it is possible to obtain:

     $$(Z_i \cap R_j) \to A \in \pi_{R_i}(F) \in G \in G^A \in G^+ \Rightarrow A \in (Z_i \cap R_j)_G^+$$

     Furthermore, by inductive hypothesis, $Z_i \cap R_j \subseteq Z_i \subseteq X_G^+$, meaning that, by closure lemma, $X \to (Z_i \cap R_j) \in G^+$, allowing to recover $X \to A \in G^+ \Leftrightarrow A \in X_G^+$ by transitivity.

  Therefore, since the statement is true for $i+1$ as well, it must hold that $Z_f \subseteq X_G^+$.

- To prove instead that $X_G^+ \subseteq Z_f$, observe that, for any set G of functional dependencies, it must hold that $X \subseteq Y \Rightarrow X_G^+ \subseteq Y_G^+$, meaning that it is sufficient to show that $X_G^+ \subseteq (Z_f)_G^+$ by proving that $Z_f \subseteq (Z_f)_G^+$.
  Most particularly, since $Z_f \subseteq (Z_f)_G^+$ is trivially true, the aim is to show

that $(Z_f)^+_G \subseteq Z_f$.
Consider the set $S' = \{A : Y \to V \in G,\ Y \subseteq Z_f \wedge A \in V\}$ that is obtained by the closure algorithm, meaning that, by correctness of the algorithm, it must hold that $S' \subseteq Z_f$: start by picking $A \in S'$, implying that $\exists\ Y \to V \in G$ such that $Y \subseteq Z_f \wedge A \in V$, meaning that, by definition of G, $\exists\ R_j \in \rho$ such that $YV \subseteq R_j$.
Most particularly, it is possible to recover $(Z_f \cap R_j) \to Y \in F^+$ by reflexivity, allowing to obtain by transitivity that $(Z_f \cap R_j) \to A \in F^+$, meaning that, by closure lemma, $A \subseteq (Z_f \cap R_j)^+_F$.
However, since $A \subseteq R_j$ as well, it is possible to conclude the proof by stating that:

$$A \in (Z_f \cap R_j)^+_F \cap R_j \subseteq S_f \subseteq Z_f \text{ by algorithm definition.}$$

This allows to show that $X^+_G \subseteq Z_f$.

Therefore, since both statements have been proved, it is possible to conclude that $Z_f = X^+_G$.

1.7 Merge properties

1.7.1 First property

It is always the case that $r \subseteq m_\rho(r)$.
For proof, pick $t \in r$ and let $t[R_i] \in \pi_{R_i}(r)\ \forall\ R_i \in \rho$, meaning that it is possible to rewrite t in the following way via natural join:

$$t = \{t[R_1]\} \bowtie \cdots \bowtie \{t[R_k]\} \subseteq \pi_{R_i}(r) \bowtie \cdots \bowtie \pi_{R_k}(r) = m_\rho(r) \Rightarrow t \subseteq m_\rho(r)$$

Since this applies $\forall\ t \in r$, it is possible to conclude that $r \subseteq m_\rho(r)$.

1.7.2 Second property

It is always the case that $\pi_{R_i}(m_\rho(r)) = \pi_{R_i}(r)$.
The statement is proved by double inclusion:

- To prove that $\pi_{R_i}(r) \subseteq \pi_{R_i}(m_\rho(r))$, pick $t_{R_i} \in \pi_{R_i}(r)$, implying that $\exists\ t \in r$ such that $t[R_i] = t_{R_i}$, resulting in $t \in m_\rho(r) \Rightarrow t[R_i] \in \pi_{R_i}(m_\rho(r))$. However, since $t[R_i] = t_{R_i}$, this is equivalent to saying $t_{R_i} \in \pi_{R_i}(m_\rho(r))$
- To prove that $\pi_{R_i}(m_\rho(r)) \subseteq \pi_{R_i}(r)$, pick $t_{R_i} \in \pi_{R_i}(m_\rho(r))$, implying that $\exists\ t \in m_\rho(r)$ such that $t[R_i] = t_{R_i}$, meaning that there must be a sequence $r_1 \in \pi_{R_1}(r), \ldots, r_k \in \pi_{R_k}(r)$ such that $t_{R_i} = t[R_i] = r_i \in \pi_{R_i}(r)\ \forall\ i$.

Since both inclusions have been proved, it is possible to conclude that, indeed, $\pi_{R_i}(m_\rho(r)) = \pi_{R_i}(r)$.

1.7.3 Third property

A merge is said to be idempotent because it is always the case that $m_\rho(m_\rho(r)) = m_\rho(r)$.
It follows from the same property that:

$$m_\rho(m_\rho(r)) = \pi_{R_1}(m_\rho(r)) \bowtie \cdots \bowtie \pi_{R_k}(m_\rho(r)) = \pi_{R_1}(r) \bowtie \cdots \bowtie \pi_{R_k}(r) = m_\rho(r)$$

The previous statement basically implies that, indeed, $m_\rho(m_\rho(r)) = m_\rho(r)$.

1.8 Correctness of the lossless join algorithm

Regardless of the outcome, the algorithm always terminates and returns the correct answer, meaning that a decomposition ρ has a lossless join if and only if $\exists\ t^a \in r$ such that $t^a = (a, \ldots, a)$.
N.B.: For the purpose of the course, only the first side of the equivalence is proved.
For proof, let r^0 and r^f denote the initial and final versions of the table r as the algorithm is executed.
Observe that, $\forall\ t_i^0 \in r^0$, it holds by construction that $t_i^0[R_i] = (a, \ldots, a)$: since the algorithm never modifies a values by definition, this implies that $t_i^0 = t_i^f = (a, \ldots, a)$.
Therefore, it is possible to reconstruct t^f by natural join and obtain that:

$$t^a \in \{t_1^f[R_1]\} \bowtie \cdots \bowtie \{t_k^f[R_k]\} \subseteq \pi_{R_i}(r^f) \bowtie \cdots \bowtie \pi_{R_k}(r^f) = m_\rho(r^f)$$

However, since r^f is obtained by satisfying every functional dependence in F, it holds by algorithm definition that r^f is a legal instance, meaning that, since ρ is assumed to have a lossless join, $m_\rho(r^f) = r^f$ and therefore, since $t^a \in r_f$, r_f contains a row $t^a = (a, \ldots, a)$.

1.9 Correctness of the decomposition algorithm

Given a relational schema R and a set F of functional dependencies that is also a minimal cover, the decomposition algorithm always manages to find in polynomial time a decomposition ρ such that F is preserved and every subschema is in third normal form.
For proof, let $G = \bigcup_{i=1}^{k} \pi_{R_i}(F)$, meaning that, $\forall\ X \to A \in F$, it is possible to state that, since $XA \in \rho$ by algorithm definition, it is possible to find $X \to A \in G$ as well, implying that $F \subseteq G$ and therefore $F^+ \subseteq G^+$ by closure inclusion lemma.
In addition, since $G \subseteq F^+$ by definition, the closure inclusion lemma allows to obtain $G^+ \subseteq F^+$, meaning that, actually, $F^+ = G^+ \Rightarrow F \equiv G$ by double inclusion.
Therefore, it is possible to conclude that ρ preserves F.

Now, the goal is to show that every subschema $R_1 \in \rho$ is in third normal form. Notice that the algorithm provides three cases:

1. $S \in \rho$
   By definition of third normal form, it holds that every attribute of S must trivially be in the key, meaning that S is in third normal form.

2. $R \in \rho$
   Since F is assumed to be a minimal cover, $\exists\ R - A \to A \in F$ and, since $\nexists\ X \subset R$ such that $X - A \to A \in F$, then $R - A$ is also a key for R.
   Therefore, given a generic functional dependence $Y \to B \in F$, either $B = A$, meaning that $Y = X - A$ is a superkey, or $B \neq A$, meaning that $B \in R - A$ is a prime attribute: this means that every functional dependence will satisfy third normal form and therefore R is also in third normal form.

3. $XA \in \rho$
   Since F is assumed to be a minimal cover, $\nexists\ X' \subset X$ such that $X \to A \in F$, meaning that X is a key for XA.
   Therefore, given a generic functional dependence $Y \to B \in F$, either $B = A$, meaning that $Y = X$ is a superkey, or $B \neq A$, meaning that $B \in X$ is a prime attribute: this means that every functional dependence will satisfy third normal form and therefore XA is also in third normal form.

N.B.: This proof does not cover the condition that ρ has a lossless join: it is proved authoritatively that ρ has a lossless join if (at least) one of its subschemas contains one of the keys of R and, if this is not the case, it is sufficient to add one of the keys as a subschema.

Chapter 2

Physical Organization

2.1 Search costs

2.1.1 Heap file organization

Assuming heap file organization with linear search, the worst-case search cost for a file containing B_T blocks will be equal to BI_T memory accesses.
N.B.: Generally speaking, these values apply for HASH organization as well.

2.1.2 Sequential file organization

Assuming sequential file organization, the worst-case search cost for a file containing B_T blcoks will actually depend on the search method:

- If linear search is used, the worst-case search cost will be equal to B_T memory accesses.
- If binary search is used, the worst-case search cost will be equal to $\log_2(B_T)$ memory accesses.

2.1.3 Indexed sequential file organization

Assuming ISAM organization, the worst-case search cost for a file containing BI_T blocks will be equal to $\lceil \log_2(BI_T) \rceil + 1$ memory accesses.

2.1.4 Binary tree organization

Assuming tree organization with N nodes, the worst-case cost is typically $\lceil \log_2(N) \rceil$. However, if the tree is complete, meaning that $N = 2^h - 1$, it is possible to rewrite this cost simply as h.

2.1.5 B-tree organization

Assuming B-tree organization with N nodes, which have at most m children and at least $d = \frac{m}{2}$ children, the average search-cost will tend to range between a best-case cost and a worst-case cost:

$$\lceil \log_m(N+1) - 1 \rceil \leq \text{ cost } \leq \lfloor \log_d(\frac{N+1}{2}) \rfloor$$

However, it is often the case that, if this tree has height h, the search cost can be expressed as h.