Introduction to Logic for Computer Science

S. Arun-Kumar

Department of Computer Science and Engineering
Indian Institute of Technology
New Delhi, 110016
email: sak@cse.iitd.ernet.in

August 18, 1999

Contents

Chapter 1

Introduction and Mathematical Preliminaries

"Contrariwise", continued Tweedledee, "if it was so, it might be, and if it were so, it would be; but as it isn't, it ain't. That's logic!"

Lewis Carroll, "Through the Looking Glass"

1.1 Motivation for the Study of Logic

In the early years of this century symbolic or formal logic became quite popular with philosophers and mathematicians because they were interested in the concept of what constitutes a correct proof in mathematics. Over the centuries mathematicians had pronounced various mathematical proofs as correct which were later disproved by other mathematicians. The whole concept of logic then hinged upon what is a correct argument as opposed to a wrong (or faulty) one. This has been amply illustrated by the number of so-called proofs that have come up for Euclid's parallel postulate and for Fermat's last theorem. There have invariably been "bugs" (a term popularized by computer scientists for the faults in a program) which were often very hard to detect and it was necessary therefore to find infallible methods of proof. For centuries (dating back at least to Plato and Aristotle) no rigorous formulation was attempted to capture the notion of a correct argument which would guide the development of all mathematics.

The early logicians of the nineteenth and twentieth centuries hoped to establish formal logic as a foundation for mathematics, though that never really happened. But mathematics does rest on one firm foundation, namely set theory. But Set theory itself has been expressed in first order logic. needed to be answered were questions relating to the automation or mechanizability of proofs. These questions are very relevant and important the development of present-day computer science and form the basis of many developments in automatic theorem proving David Hilbert asked the important question, as to whether all mathematics, if reduced to statements of symbolic logic, can be derived by a machine., be reduced to the manipulation of statements in symbolic logic., it enabled mathematicians to point out why an alleged proof is wrong, or where in the proof the reasoning has been faulty. A large part of the credit for this achievement must go to the fact that by symbolizing arguments rather than writing them out in some natural language (which is fraught with ambiguity), checking the correctness of a proof becomes a much more viable task. Of course, trying to symbolize the whole of mathematics could be disastrous as then it would become quite impossible to even read and understand mathematics, since what is presented usually as a one page proof could run into several pages. But at least in principle it can be done.

Since the latter half of the twentieth century logic has been used in computer science for various purposes ranging from program specification and verification to theorem-proving. Initially its use was restricted to merely specifying programs and reasoning about their implementations. This is exemplified in the some fairly elegant research on the development of correct programs using first-order logic in such calculi such as the weakest-precondition calculus of Dijkstra. A method called Hoare Logic which combines first-order logic sentences and program phrases into a specification and reasoning mechanism is also quite useful in the development of small programs. Logic in this form has also been used to specify the meanings of some programming languages, notably Pascal.

The close link between logic as a formal system and computer-based theorem proving is proving to be very useful especially where there are a large number of cases (following certain patterns) to be analyzed and where quite often there are routine proof techniques available which are more easily and accurately performed by theorem-provers than by humans. The case of the four-colour theorem which until fairly recently remained a unproved conjecture is an instance of how human ingenuity and creativity may be used to divide up proof into a few thousand cases and where machines may be used to perform routine checks on the individual cases. Another use of computers in theorem-proving or model-checking is the verification of the design of large circuits before a chip is fabricated. Analyzing circuits with a billion transistors in them is at best error-prone and at worst a drudgery that few humans would like to do. Such analysis and results are best performed by machines using theorem proving techniques or model-checking techniques.

A powerful programming paradigm called declarative programming has evolved since the late seventies and has found several applications in computer science and artificial intelligence. Most programmers using this logical paradigm use a language called Prolog which is an implemented form of logic[1]. More recently computer scientists are working on a form of logic called constraint logic programming.

In the rest of this chapter we will discuss sets, relations, functions. Though most of these topics are covered in the high school curriculum this section also establishes the notational conventions that will be used throughout. Even a confident reader may wish to browse this section to get familiar with the notation.

1.2 Sets

A *set* is a collection of *distinct* objects. The class of CS253 is a set. So is the group of all first year students at the IITD. We will use the notation $\{a, b, c\}$ to denote the collection of the objects a, b and c. The elements in a set are not ordered in any fashion. Thus the set $\{a, b, c\}$ is the same as the set $\{b, a, c\}$. Two sets are *equal* if they contain exactly the same elements.

We can describe a set either by enumerating all the elements of the set or by stating the properties that uniquely characterize the elements of the set. Thus, the set of all even positive integers not larger than 10 can be described either as $S = \{2, 4, 6, 8, 10\}$ or, equivalently, as $S = \{x \mid x \text{ is an even positive integer not larger than } 10\}$

A set can have another set as one of its elements. For example, the set $A = \{\{a, b, c\}, d\}$ contains two elements $\{a, b, c\}$ and d; and the first element is itself a set. We will use the

[1] actually a subset of logic called Horn-clause logic

notation $x \in S$ to denote that x is an *element of* (or *belongs to*) the set S.

A set A is a *subset* of another set B, denoted as $A \subseteq B$, if $x \in B$ whenever $x \in A$.

An *empty set* is one which contains no elements and we will denote it with the symbol $\emptyset$. For example, let S be the set of all students who fail this course. S might turn out to be empty (hopefully; if everybody studies hard). By definition, the empty set $\emptyset$ is a subset of all sets. We will also assume an *Universe of discourse* $\mathbb{U}$, and every set that we will consider is a subset of $\mathbb{U}$. Thus we have

1. $\emptyset \subseteq A$ for any set A
2. $A \subseteq \mathbb{U}$ for any set A

The *union* of two sets A and B, denoted $A \cup B$, is the set whose elements are exactly the elements of either A or B (or both). The *intersection* of two sets A and B, denoted $A \cap B$, is the set whose elements are exactly the elements that belong to *both* A and B. The *difference* of B from the A, denoted $A - B$, is the set of all elements of A that do not belong to B. The *complement* of A, denoted $\sim A$ is the difference of A from the universe $\mathbb{U}$. Thus, we have

1. $A \cup B = \{x \mid (x \in A) \text{ or } (x \in B)\}$
2. $A \cap B = \{x \mid (x \in A) \text{ and } (x \in B)\}$
3. $A - B = \{x \mid (x \in A) \text{ and } (x \notin B)\}$
4. $\sim A = \mathbb{U} - A$

We also have the following named identities that hold for all sets A, B and C.

Basic properties of set union.

1. $(A \cup B) \cup C = A \cup (B \cup C)$ *Associativity*
2. $A \cup \phi = A$ *Identity*
3. $A \cup \mathbb{U} = \mathbb{U}$ *Zero*
4. $A \cup B = B \cup A$ *Commutativity*
5. $A \cup A = A$ *Idempotence*

Basic properties of set intersection

1. $(A \cap B) \cap C = A \cap (B \cap C)$ *Associativity*
2. $A \cap \mathbb{U} = A$ *Identity*
3. $A \cap \phi = \phi$ *Zero*
4. $A \cap B = B \cap A$ *Commutativity*
5. $A \cap A = A$ *Idempotence*

Other properties

1. $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ *Distributivity of $\cap$ over $\cup$*

2. $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ *Distributivity of* $\cup$ *over* $\cap$

3. $\sim (A \cup B) = \sim A \cap \sim B$ *De Morgan's law* $\sim \cup$

4. $\sim (A \cap B) = \sim A \cup \sim B$ *De Morgan's law* $\sim \cap$

5. $A \cap (\sim A \cup B) = A \cap B$ *Absorption* $\cup$

6. $A \cup (\sim A \cap B) = A \cup B$ *Absorption* $\cap$

The reader is encouraged to come up with properties of set difference and the complementation operations.

We will use the following notation to denote some standard sets:

The empty set: $\emptyset$

The Universe: $\mathbb{U}$

The Power set of a set A: $\mathbf{2}^A$ is the set of all subsets of the set A.

The set of Natural Numbers: [2] $\mathbb{N} = \{0, 1, 2, \dots\}$

The set of positive integers: $\mathbb{P} = \{1, 2, 3, \dots\}$

The set of integers: $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$

The set of real numbers: $\mathbb{R}$

The Boolean set: $\mathbb{B} = \{false, true\}$

1.3 Relations and Functions

The *Cartesian product* of two sets A and B, denoted by $A \times B$, is the set of all ordered pairs (a, b) such that $a \in A$ and $b \in B$. Thus,

$$A \times B = \{(a, b) \mid a \in A \text{ and } b \in B\}$$

Given another set C we may form the following different kinds of cartesian products (which are not at all the same!).

$$(A \times B) \times C = \{((a, b), c) \mid a \in A, b \in B \text{ and } c \in C\}$$

$$A \times (B \times C) = \{(a, (b, c)) \mid a \in A, b \in B \text{ and } c \in C\}$$

$$A \times B \times C = \{(a, b, c) \mid a \in A, b \in B \text{ and } c \in C\}$$

[2] We will include 0 in the set of Natural numbers. After all, it is quite natural to score a 0 in an examination

The last cartesian product gives the construction of tuples. Elements of the set $A_1 \times A_2 \times \cdots \times A_n$ for given sets $A_1, A_2, \ldots, A_n$ are called *ordered n-tuples.*

A^n is the set of all ordered n-tuples $(a_1, a_2, \ldots, a_n)$ such that $a_i \in A$ for all i. i.e.,

$$A^n = \underbrace{A \times A \times \cdots \times A}_{n \text{ times}}$$

A *binary relation* $\mathcal{R}$ from A to B is a subset of $A \times B$. It is a characterization of the intuitive notion that some of the elements of A are related to some of the elements of B. We also use the notation $a\mathcal{R}b$ to mean $(a, b) \in \mathcal{R}$. When A and B are the same set, we say $\mathcal{R}$ is a binary relation *on* A. Familiar binary relations from $\mathbb{N}$ to $\mathbb{N}$ are $=$, $\neq$, $<$, $\leq$, $>$, $\geq$. Thus the elements of the set $\{(0,0), (0,1), (0,2), \ldots, (1,1), (1,2), \ldots\}$ are all members of the relation $\leq$ which is a subset of $\mathbb{N} \times \mathbb{N}$.

In general, an *n-ary relation* among the sets $A_1, A_2, \ldots, A_n$ is a subset of the set $A_1 \times A_2 \times \cdots \times A_n$.

Definition 1.1 *Let $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{B}$ be a binary relation from A to B. Then*

1. *For any set $A' \subseteq A$ the* image *of A' under $\mathcal{R}$ is $\mathcal{R}(A') = \{b \in B \mid a\mathcal{R}b \; for \; some \; a \in A'\}$;*

2. *For every subset $B' \subseteq B$ the* pre-image *under $\mathcal{R}$ is $\mathcal{R}^{-1}(B') = \{a \in A \mid a\mathcal{R}b \; for \; some \; b \in B'\}$;*

3. *$\mathcal{R}$ is* onto *(or* surjective*) with respect to A and B if $\mathcal{R}(A) = B$;*

4. *$\mathcal{R}$ is* total *with respect to A and B if $\mathcal{R}^{-1}(B) = A$;*

5. *$\mathcal{R}$ is* one-to-one *(or* injective*) with respect to A and B if for every $b \in B$ there is at most one $a \in A$ such that $(a, b) \in \mathcal{R}$.*

6. *$\mathcal{R}$ is a* partial function *from A to B, usually denoted $\mathcal{R} : \mathcal{A} \hookrightarrow \mathcal{B}$, if for every $a \in A$ there is at most one $b \in B$ such that $(a, b) \in \mathcal{R}$;*

7. *$\mathcal{R}$ is a* total function *from A to B, usually denoted $\mathcal{R} : \mathcal{A} \rightarrow \mathcal{B}$ if $\mathcal{R}$ is a partial function from A to B and is total.*

8. *$\mathcal{R}$ is a* one-to-one correspondence *(or* bijection*) if it is an injective and surjective total function.*

Example 1.1 *The following are some examples of familiar binary relations along with their properties.*

1. *The $\leq$ relation on $\mathbb{N}$ is a relation from $\mathbb{N}$ to $\mathbb{N}$ which is total and onto. That is, both the image and pre-image of $\leq$ under $\mathbb{N}$ are $\mathbb{N}$ itself. What are image and the pre-image respectively of the relation $<$?*

2. *The binary relation which associates key sequences from a computer keyboard with their respective 8-bit ASCII codes is an example of a relation which is total and injective.*

3. *The binary relation which associates 7-bit ASCII codes with the corresponding ASCII character set is an example of a bijection.*

We may equivalently define partial and total functions as follows.

Definition 1.2 *A* function *(or a* total function*) f from A to B is a binary relation $f \subseteq A \times B$ such that for* every *element $a \in A$ there is a* unique *element $b \in B$ so that $(a, b) \in f$ (usually denoted $f(a) = b$ and sometimes $f : a \mapsto b$). We will use the notation $R : A \to B$ to denote a function R from A to B. The set A is called the* domain *of the function R and the set B is called the* co-domain *of the function R. The* range *of a function $R : A \to B$ is the set $\{b \in B \mid$ for some $a \in A$, $R(a) = b\}$. A* partial function *f from A to B, denoted $f : A \hookrightarrow B$ is a total function from some subset of A to the set B. Clearly every total function is also a partial function.*

The word "function" unless otherwise specified is taken to mean a "total function". Some familiar examples of partial and total functions are

1. $+$ and $*$ (addition and multiplication) are total functions of the type $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$
2. $-$ (subtraction) is a partial function of the type $f : \mathbb{N} \times \mathbb{N} \hookrightarrow \mathbb{N}$.
3. div and mod are total functions of the type $f : \mathbb{N} \times \mathbb{P} \to \mathbb{N}$. If $a = q * b + r$ such that $0 \leq r < b$ and $a, b, q, r \in \mathbb{N}$ then the functions div and mod are defined as $div(a, b) = q$ and $mod(a, b) = r$. We will often write these binary functions as $a * b$, $a\ div\ b$, $a\ mod\ b$ etc. Note that div and mod are also partial functions of the type $f : \mathbb{N} \times \mathbb{N} \hookrightarrow \mathbb{N}$.
4. The binary relations $=$, $\neq$, $<$, $\leq$, $>$, $\geq$ may also be thought of as functions of the type $f : \mathbb{N} \times \mathbb{N} \to \mathbb{B}$ where $\mathbb{B} = \{false, true\}$.

Definition 1.3 *Given a set A, a* list *(or* finite sequence*) of length $n \geq 0$ of elements from A, denoted $\vec{a}$, is a (total) function of the type $\vec{a} : \{1, 2, \ldots, n\} \to A$. We normally denote a list of length n by $[a_1, a_2, \ldots, a_n]$ Note that the empty list, denoted $[]$, is also such a function $[] : \emptyset \to A$ and denotes a sequence of length 0.*

It is quite clear that there exists a simple bijection from the set A^n (which is the set of all n-tuples of elements from the set A) and the set of all lists of length n of elements from A. We will often identify the two as being the same set even though they are actually different by definition[3]. The set of all lists of elements from A is denoted A^*, where

$$A^* = \bigcup_{n \geq 0} A^n$$

The set of all *non-empty* lists of elements from A is denoted A^+ and is defined as

$$A^+ = \bigcup_{n > 0} A^n$$

An *infinite* sequence of elements from A is a total function from $\mathbb{N}$ to A. The set of all such infinite sequences is denoted A^ω.

[3] In a programming language like ML, the difference is evident from the notation and the constructor operations for tuples and lists

1.4 Operations on Binary Relations

In this section we will consider various operations on binary relations.

Definition 1.4 *1. Given a set A, the* identity *relation over A, denoted $\mathcal{I}_A$, is the set $\{\langle a, a\rangle \mid a \in A\}$.*

2. *Given a binary relation $\mathcal{R}$ from A to B, the* converse *of $\mathcal{R}$, denoted $\mathcal{R}^{-1}$ is the relation from B to A defined as $\mathcal{R}^{-1} = \{(b, a) \mid (a, b) \in \mathcal{R}\}$.*

3. *Given binary relations $\mathcal{R} \subseteq A \times B$ and $\mathcal{S} \subseteq B \times C$, the* composition *of $\mathcal{R}$ with $\mathcal{S}$ is denoted $\mathcal{R} \circ \mathcal{S}$ and defined as $\mathcal{R} \circ \mathcal{S} = \{(a, c) \mid a\mathcal{R}b \text{ and } b\mathcal{S}c, \ for\ some\ b \in B\}$.*

Note that unlike in the case of functions (where for any function $f : A \to B$ its inverse $f^{-1} : B \to A$ may not always be defined), the converse of a relation is always defined. Given functions (whether partial or total) $f : A \hookrightarrow B$ and $g : B \hookrightarrow C$, their composition is the function $f \circ g : A \hookrightarrow C$ defined simply as the relational composition of the two functions regarded as binary relations. Hence $(f \circ g)(a) = g(f(a))$.

1.5 Ordering Relations

We may define the n-fold composition of a relation $\mathcal{R}$ on a set A by induction as follows

$$\mathcal{R}^0 = \mathcal{I}_A$$

$$\mathcal{R}^{n+1} = \mathcal{R}^n \circ R$$

We may combine these n-fold compositions to yield the *reflexive-transitive closure* of $\mathcal{R}$, denoted $\mathcal{R}$, as the relation

$$\mathcal{R}^* = \bigcup_{n \geq 0} \mathcal{R}^n$$

Sometimes it is also useful to consider merely the *transitive closure* $\mathcal{R}^+$ of $\mathcal{R}$ which is defined as

$$\mathcal{R}^+ = \bigcup_{n > 0} \mathcal{R}^n$$

Definition 1.5 *A binary relation $\mathcal{R}$ on a set A is*

1. reflexive *if and only if for every $a \in A$, $a\mathcal{R}a$;*

2. irreflexive *if and only if for no $a \in A$, $a\mathcal{R}a$;*

3. symmetric *if and only if for all $a, b \in A$, $a\mathcal{R}b$ implies $b\mathcal{R}a$;*

4. asymmetric *if and only if $a\mathcal{R}b$ implies $(b, a) \notin \mathcal{R}$;*

5. antisymmetric *if and only if $a\mathcal{R}b$ and $b\mathcal{R}a$ implies $a = b$;*

6. transitive *if and only if for all $a, b, c \in A$, $a\mathcal{R}b$ and $b\mathcal{R}c$ implies $a\mathcal{R}c$;*

7. connected *if and only if for all* $a, b \in A$*, if* $a \neq b$ *then* $a\mathcal{R}b$ *or* $b\mathcal{R}a$*.*

Given any relation $\mathcal{R}$ on a set A, it is easy to see that $\mathcal{R}^*$ is both reflexive and transitive.

Example 1.2

1. *The edge relation on an undirected graph is an example of a symmetric relation.*
2. *In any directed acyclic graph the edge relation is asymmetric.*
3. *Consider the reachability relation on a directed graph defined as: A pair of vertices* (A, B) *is in the reachability relation, if either* $A = B$ *or there exists a vertex* C *such that both* (A, C) *and* (C, B) *are in the reachability relation. The reachability relation is the reflexive transitive closure of the edge relation.*
4. *The reachability relation on directed graphs is also an example of a relation that need not be either symmetric or asymmetric. The relation need not be antisymmetric either.*

1.6 Partial Orders and Trees

Definition 1.6 *A binary relation* $\mathcal{R}$ *on a set* A *is*

1. *a* preorder *if it is reflexive and transitive;*
2. *a* strict preorder *if it is irreflexive and transitive;*
3. *a* partial order *if it is an antisymmetric preorder;*
4. *a* strict partial order *if it is irreflexive, asymmetric and transitive;*
5. *a* linear order[4] *if it is a connected partial order.*
6. *a* strict linear order *if it is an irreflexive linear order.*
7. *an* equivalence *if it is reflexive, symmetric and transitive.*

1.7 Exercises

1. Prove that for any binary relations $\mathcal{R}$ and $\mathcal{S}$ on a set A,
   (a) $(\mathcal{R}^{-1})^{-1} = \mathcal{R}$
   (b) $(\mathcal{R} \cap \mathcal{S})^{-1} = \mathcal{R}^{-1} \cap \mathcal{S}^{-1}$
   (c) $(\mathcal{R} \cup \mathcal{S})^{-1} = \mathcal{R}^{-1} \cup \mathcal{S}^{-1}$
   (d) $(\mathcal{R} - \mathcal{S})^{-1} = \mathcal{R}^{-1} - \mathcal{S}^{-1}$
2. Prove that the composition operation on relations is associative. Give an example of the composition of relations to show that relational composition is not commutative.
3. Prove that for any binary relations $\mathcal{R}$, $\mathcal{R}'$ from A to B and $\mathcal{S}$, $\mathcal{S}'$ from B to C, if $\mathcal{R} \subseteq \mathcal{R}'$ and $\mathcal{S} \subseteq \mathcal{S}'$ then $\mathcal{R} \circ \mathcal{S} \subseteq \mathcal{R}' \circ \mathcal{S}'$

[4]also called *total order*

4. Prove that relational composition satisfies the following distributive laws for relations, where $\mathcal{R} \subseteq A \times B$ and $\mathcal{S}, \mathcal{T} \subseteq B \times C$.

   (a) $\mathcal{R} \circ (\mathcal{S} \cup \mathcal{T}) = (\mathcal{R} \circ \mathcal{S}) \cup (\mathcal{R} \circ \mathcal{T})$

   (b) $\mathcal{R} \circ (\mathcal{S} \cap \mathcal{T}) = (\mathcal{R} \circ \mathcal{S}) \cap (\mathcal{R} \circ \mathcal{T})$

   (c) $\mathcal{R} \circ (\mathcal{S} - \mathcal{T}) = (\mathcal{R} \circ \mathcal{S}) - (\mathcal{R} \circ \mathcal{T})$

5. Prove that for $\mathcal{R} \subseteq A \times B$ and $\mathcal{S} \subseteq B \times C$, $(\mathcal{R} \circ \mathcal{S})^{-1} = (\mathcal{S}^{-1}) \circ (\mathcal{R}^{-1})$.

6. Show that a relation $\mathcal{R}$ on a set A is

   (a) reflexive if and only if $\mathcal{I}_A \subseteq \mathcal{R}$;

   (b) irreflexive if and only if $\mathcal{I}_A \cap \mathcal{R} = \emptyset$;

   (c) symmetric if and only if $\mathcal{R} = \mathcal{R}^{-1}$;

   (d) asymmetric if and only if $\mathcal{R} \cap \mathcal{R}^{-1} = \emptyset$;

   (e) antisymmetric if and only if $\mathcal{R} \cap \mathcal{R}^{-1} \subseteq \mathcal{I}_A$;

   (f) transitive if and only if $\mathcal{R} \circ \mathcal{R} \subseteq \mathcal{R}$;

   (g) connected if and only if $(A \times A) - \mathcal{I}_A \subseteq \mathcal{R} \cup \mathcal{R}^{-1}$

7. Consider any reflexive relation $\mathcal{R}$ on a set A. Does it necessarily follow that A is not asymmetric? If $\mathcal{R}$ is asymmetric does it necessarily follow that it is irreflexive?

8. Find the fallacy in the proof of the following alleged theorem.

   Theorem 1.1 *A symmetric and transitive binary relation is an equivalence.*

   Proof: Let $\mathcal{R}$ be a symmetric and transitive binary relation on a set A. For any pair of elements $(a, b) \in \mathcal{R}$, it follows from symmetry that $(b, a) \in \mathcal{R}$. Further by transitivity, it follows that if both $(a, b), (b, a) \in \mathcal{R}$ then $(b, b) \in \mathcal{R}$. Hence $\mathcal{R}$ is reflexive. Since it is also symmetric and transitive, $\mathcal{R}$ is an equivalence. □

9. Prove that there exists a bijection

   (a) from $\mathbb{N}^2$ to $\mathbb{N}$,

   (b) from $\mathbb{N}^n$ to $\mathbb{N}$, for any $n > 0$,

   (c) from $\mathbb{N}*$ to $\mathbb{N}$.

10. Can you prove that there exists no bijection between $\mathbb{N}^\omega$ and $\mathbb{N}$?

11. Show that for any relation $\mathcal{R}$, its reflexive-transitive closure $\mathcal{R}^*$ is a preorder.

12. Prove that for any relation $\mathcal{R}$ on a set A,

    (a) $\mathcal{S} = \mathcal{R}^* \cup (\mathcal{R}^*)^{-1}$ and $\mathcal{T} = (\mathcal{R} \cup \mathcal{R}^{-1})^*$ are both equivalence relations.

    (b) Prove or disprove[5]: $\mathcal{S} = \mathcal{T}$.

[5] that is, give an example of a relation $\mathcal{R}$ such that for $\mathcal{S}$ and $\mathcal{T}$ constructed as above, the given statement is contradicted

13. Given any preorder $\mathcal{R}$ on a set A, prove that the *kernel* of the preorder defined as $\mathcal{R} \cap \mathcal{R}^{-1}$ is an equivalence relation.

14. Consider any preorder $\mathcal{R}$ on a set A. We give a construction of another relation as follows. For each $a \in A$, let $[a]_{\mathcal{R}}$ be the set defined as $[a]_{\mathcal{R}} = \{b \in A \mid a\mathcal{R}b \textit{ and } b\mathcal{R}a\}$. Now consider the set $B = \{[a]_{\mathcal{R}} \mid a \in A\}$. Let $\mathcal{S}$ be a relation on B such that for every $a, b \in A$, $[a]_{\mathcal{R}}\mathcal{S}[b]_{\mathcal{R}}$ if and only if $a\mathcal{R}b$. Prove that $\mathcal{S}$ is a partial order on the set B.