CS235 Languages and Automata
Prof. Lyn Turbak
Wellesley College

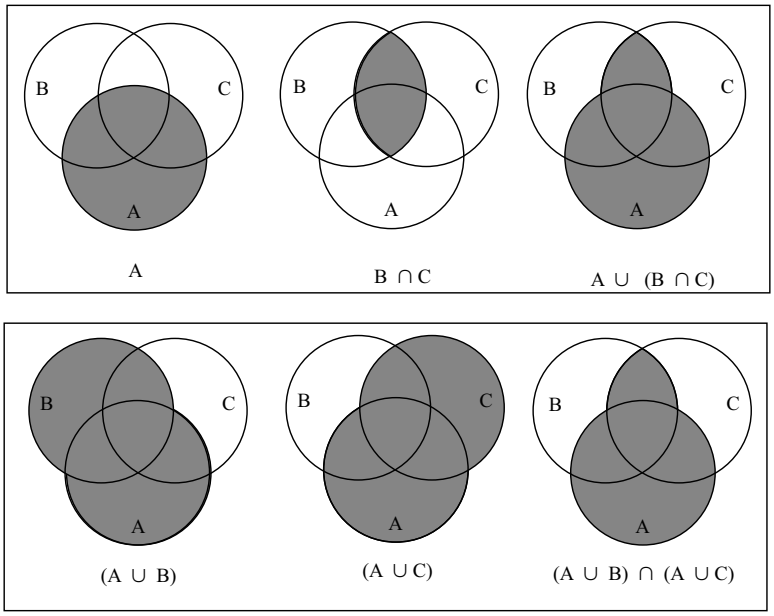
Handout # 5
September 27, 2007

Problem Set 1 Solutions

Problem 1 [15]

You were asked to give two proofs that $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$:

a. **[5]** Below is a "proof by picture", using Venn diagrams:



b. **[10]** Here is a symbolic (i.e., text-based as opposed to picture-based) proof that does not use any pictures. We want to show $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$. To prove equality between two sets, we show that each is a subset of the other.

$A \cup (B \cap C) \subseteq (A \cup B) \cap (A \cup C)$**:**

To show one set is a subset of another, take an arbitrary element of the first set and show it is in the second. Here, suppose $x \in A \cup (B \cap C)$. By the definition of $\cup$, either $x \in A$ or $x \in (B \cap C)$. We consider both cases in turn:

$x \in A$**:** If $x \in A$, then $x \in (A \cup B)$ and $x \in (A \cup C)$ (by the definition on $\cup$), so $x \in (A \cup B) \cap (A \cup C)$ (by the definition of $\cap$)[1]

$x \in (B \cap C)$ If $x \in (B \cap C)$, then $x \in B$ and $x \in C$ (by the definition of $\cap$). $x \in B$ implies $x \in (A \cup B)$ and $x \in C$ implies $x \in (A \cup C)$, so $x \in (A \cup B) \cap (A \cup C)$.

$(A \cup B) \cap (A \cup C) \subseteq A \cup (B \cap C)$**:**

Suppose $y \in (A \cup B) \cap (A \cup C)$ Then $y \in (A \cup B)$ and $y \in (A \cup C)$. We consider two cases, based on whether y is in A:

$y \in A$**:** If $y \in A$, then $y \in (A \cup S)$ for any set S. So $y \in (A \cup B)$ and $y \in (A \cup C)$, making it in $(A \cup B) \cap (A \cup C)$. And it is also in $A \cup (B \cap C)$.

$y \notin A$**:** Since $y \in (A \cup B) \cap (A \cup C)$, it must be that $y \in (A \cup B)$ and $y \in (A \cup C)$. But $y \notin A$, so it must be that $y \in B$ and $y \in C$. So $y \in (B \cap C)$, implying $y \in A \cup (B \cap C)$.

[1] Normally, we don't need to justify set operations using phrases like "by the definition of $\cup$" or "by the definition of $\cap$". I'm doing it here because it's the first set problem we're doing. I will omit such justifications later.

Problem 2 [10]

You were given the following two properties:

Prop 1: $\overline{\overline{A}} = A$

Prop 2: $\overline{A \cup B} = \overline{A} \cap \overline{B}$ (DeMorgan's law (1) from Slide 2-12).

Using the above properties, here is an algebraic proof that $\overline{C \cap D} = \overline{C} \cup \overline{D}$ (DeMorgan's law (2) from Slide 2-12).

$$\begin{aligned}
\overline{C \cap D} &= \overline{\overline{\overline{C}} \cap \overline{\overline{D}}} && \text{by two applications of Prop 1, one with } A = C \text{ and one with } A = D. \\
&= \overline{\overline{\overline{C} \cup \overline{D}}} && \text{by Prop 2, where } A = \overline{C} \text{ and } B = \overline{D}. \text{ I.e., } \boxed{\overline{\overline{C}}} \cap \boxed{\overline{\overline{D}}} = \overline{\boxed{\overline{C}} \cup \boxed{\overline{D}}} \\
&= \overline{C} \cup \overline{D} && \text{by Prop 1, where } A = \overline{\overline{C} \cup \overline{D}}
\end{aligned}$$

Above, we have renamed A and B to C and D in what we're trying to prove to avoid any confusion with the As and Bs mentioned in the properties. Many students did, in fact, confuse the As and Bs from different parts of the proof. If you find the names confusing, use consistent renaming to make the names different!

An algebraic proof is organized as a series of equalities, where each equality step is justified by some property or definition. It is not always possible to write proofs in an algebraic style, but proofs written in such a style tend to be very clear.

Problem 3 [20] You were given the following relations, where F is assumed to be any finite subset of *Nat* and G is assumed to be any undirected graph with a finite number of vertices:

$sameMod5 = \{(a, b) \mid a, b \in Int \text{ and } (a - b) \bmod 5 = 0\}$

$differsByOneChar = \{\ (r, s) \mid\ r, s \in String, \text{length}(r) = \text{length}(s) > 0,$ and r and s have the same characters at every index but one.$\}$

$connected = \{(v, w) \mid v, w \text{ are vertices in } G \text{ and } (v, w) \text{ is an edge in } G\}$

$pairLEQ = \{(p, q) \mid p, q \in Nat \times Nat, \text{first(p)} \leq \text{first(q)}, \text{ and second(p)} \leq \text{ second(q)}\}$

$hasOneLessElt = \{(A, B) \mid A, B \in \mathcal{P}(F) \text{ and } B = A \cup \{n\} \text{ for some } n \in F \text{ s.t } n \notin A\}$

Here are English descriptions of these relations:

- a *sameMod5* b iff a and b have the same value mod 5.
- r *differsByOneChar* s iff r and s are strings of the same length that have different characters at exactly one index.
- v *connected* w iff v and w are connected by an edge in G.
- $pair_1$ *pairLEQ* $pair_2$ iff the components of $pair_1$ are less than or equal to the corresponding components of $pair_2$.
- A *hasOneLessElt* B iff A and B are subsets of F and A has all the elements of B but one.

a. **[7]** Here is a table indicating the properties of these relations:

Relation	Reflexive	Symmetric	Transitive	Equivalence Relation?
sameMod5	Yes	Yes	Yes	Yes
differsByOneChar	No	Yes	No	No
connected	No	Yes	No	No
pairLEQ	Yes	No	Yes	No
hasOneLessElt	No	No	No	No

Note: In an undirected graph, self-edges (edges from a vertex to itself) are usually disallowed. Even if they are allowed, vertices are not automatically considered connected to themselves, so the *connected* relation is not reflexive.

b. **[13]** For each of the above relations, describe (in English) its (1) reflexive closure; (2) transitive closure; and (3) reflexive, transitive closure.

- *sameMod5* is already reflexive and transitive. So all three closures yield the original relation, *sameMod5*.
- *differsByOneChar*:

  Reflexive Closure relates two strings of the same length that are exactly the same or differ in exactly one character posiiton. (Using an English name like *differsByOneCharOrSame* is too vague, since it doesn't specify that the strings must have the same length.)

  Transitive Closure relates any two strings of the same length.

  Reflexive Transitive Closure relates any two strings of the same length (same as the transitive closure).

- *connected*:

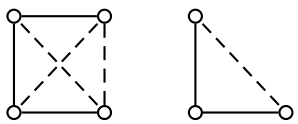
  Reflexive Closure relates two vertices if they are the same or connected by a single edge. Alternatively, it relates all vertices connected by a path of length 0 or 1.

  Transitive Closure relates two vertices iff they are connected by a nonzero-length path.

  Reflexive Transitive Closure relates two vertice iff they are connected by any path (including one of zero length).

  Notes:

  - The closure operations do *not* actually add any edges to the graph G, which remains unchanged. However, the closure operations may relate vertices which were unrelated before, which is like adding "virtual" edges.
  - Many students said the result of the transitive closure is a **complete** graph — i.e., a graph in which every vertex is connected to every other vertex (we'll assume by a virtual edge rather than a real one). But this isn't true in general. The transitive closure will not relate vertices that weren't originally connected by some path. For example, consider the following graph, in which solid lines are the original edges, and dashed lines are the virtual ones added by transitive closure. The virtual edges make each connected component complete, but do not make the whole graph complete.



- *pairLEQ* is already reflexive and transitive. So all three closures yield the original relation, *pairLEQ*.
- *hasOneLessElt*:

Reflexive Closure relates A and B if they are the same set or A has all the elements of B but one.

Transitive Closure is the proper subset relation. I.e., it relates A and B iff $A \subset B$.

Reflexive Transitive Closure is the subset relation. I.e., it relates A and B iff $A \subseteq B$.

Many students invented tortured descriptions of the relations resulting from the closures of *hasOneLessElt*. Strive for simple descriptions!

Problem 4 [10] You were asked to find the error in the following proof that $2 = 1$:

> Consider the equation $a = b$. Multiply both sides by a to obtain $a^2 = ab$. Subtract b^2 from both sides to get $a^2 - b^2 = ab - b^2$. Now factor each side, $(a+b)(a-b) = b(a-b)$, and divide each side by $(a-b)$, to get $a + b = b$. Finally, let a and b equal 1, which shows that $2 = 1$.

The proof begins with the assumption that $a = b$, so $(a - b) = 0$. Since dividing by 0 is undefined, the step that divides each side by $(a - b)$ is invalid.

Problem 5 [15]

How did one of Shelby Fuddled students know she had a chalk mark on her head? She reasoned as follows:

> I either have a chalk mark on my head or I do not. Suppose I do not. Then my classmates have their hands up because they see each others' chalk mark. But seeing my unchalked head, each classmate (who is very smart, after all) would quickly realize that her head must be chalked (else the other classmate would not have raised her hand) and would then lower her hand quickly. But they did not do this. Therefore, my head must be chalked.

This proof involves contradiction at two levels.

1. The winning student determines her answer by assuming one case (her head is unchalked) and showing that this leads to a contradiction.
2. In reasoning about the case where her head is unchalked, the winning student W imagines how her classmates would reason. Each classmate C would assume a case that *her own* head is unchalked and arrive at a contradiction (the other classmate's hand would not be raised).

Problem 6 has been moved to PS2.

Problem 7 [15] (from Sipser) Show that every graph with 2 or more nodes contains two nodes that have equal degrees. *(Hint: this is a counting problem. Use the pigeon-hole principle.)* Here's one way to phrase this proof, using the pigeon-hole principle:

> In a graph G with k nodes, nodes can have degree between 0 and $k - 1$. G cannot have both a node a with degree 0 and a node b with degree $k - 1$, because b would have to be connected to a (which, because it has degree 0, cannot be connected to any other node). So the possible degrees of nodes in G can range over $[0..k - 2]$ or $[1..k - 1]$, both of which have $k - 1$ elements. Because there are k nodes and only $k - 1$ possible degrees, by the pigeon-hole principle, two nodes in G must have the same degree. (Here, nodes in G are the pigeons and possible degrees of nodes in G are the holes.)

Here's another way to phrase this proof (due Randy) as a proof by contradiction:

In a graph with k nodes, nodes can have degree between 0 and $k-1$. If G is a graph with k nodes, no two of which have the same degree, there must be a one-to-one correspondence between the degrees of nodes of G and the integers $[0..k-1]$. This implies that some node has degree 0 and hence is connected to no other node in the graph, while another has degree $k-1$ and is connected to every node in the graph. This is a contradiction.

Problem 8 [15] For any set S, you were asked to define a bijection between $\mathcal{P}(S)$ (the power set of S) and $S \to Bool$.

This was the toughest problem on the assignment, and gave many students a lot of trouble. It is worth everyone's time to study the solution to this problem carefully.

Every proof begins with an intuition. (If you don't understand *why* something is true, how can you possibly prove it?) Intuitively, why is there a bijection between $\mathcal{P}(S)$ (the power set of S) and $S \to Bool$?

Well, first off, let's remind ourselves of what a bijection is. Informally, there is a bijection between sets A and B if every element of A is "matched" with one of B, and vice versa. In the case where S is a finite set, we know that $|\mathcal{P}(S)| = 2^{|S|}$ and $|S \to Bool| = |Bool|^{|S|} = 2^{|S|}$. Since there is a bijection between any two finite sets of the same size, the desired result clearly holds when S is finite.

But how about when S is infinite? In this case, we must actually describe the correspondence between $\mathcal{P}(S)$ and $S \to Bool$. To figure this out, we must understand what kinds of elements each set has:

- Elements of $\mathcal{P}(S)$ are just subsets of S. Any particular subset $A \subseteq S$ is characterized by those elements that are in A (contrasted with those elements in S that are not in A). For example, sample subsets of Nat are (1) the empty set; (2) the set $\{2, 3, 5\}$; (3) the set of even numbers; and (4) the set Nat itself.

- Elements of $S \to Bool$ are *predicate functions* on S — i.e., functions that map every element of S to *true* or *false*. Any particular predicate $p \in S \to Bool$ is characterized by those elements for which it returns true (contrasted with those elements for which it returns false). For example, sample predicates in $Nat \to Bool$ are (1) the constant *false* function, which returns *false* for every number; (2) the function that returns *true* for the numbers 2,3,and 5 and *false* for all other numbers; (3) the *isEven?* function that returns *true* for even numbers and false for odd numbers; and (4) the constant *true* function, which returns *true* for every number.

Notice the correspondence between the sample elements of $\mathcal{P}(Nat)$ and the sample elements of $Nat \to Bool$. The predicate returns *true* for exactly the elements that are in the corresponding set. Or, equivalently, the set contains exactly the elements for which the corresponding predicate returns *true*. So subsets of Nat and predicate functions on Nat are really just two different notations for exactly the same thing!

Now we need to formalize the above intuition. Formally, there is a bijection between A and B if there is a function $f \in A \to B$ and an inverse function $f^{-1} \in B \to A$ such that:

1. $(f^{-1} \circ f) = id_A$. That is, $f^{-1}(f(x)) = x$ for all $x \in A$. In other words, if we start at an element x in A, take the map f to B, and take the map f^{-1} back, we end up exactly where we started!

2. $(f \circ f^{-1}) = id_B$. That is, $f(f^{-1}(y)) = y$ for all $y \in B$. In other words, going from B via f^{-1} to A and back to B via f doesn't lose any information.

Here, f and f^{-1} define the correspondences that match the elements of A and B.

We can capture this correspondence by formally defining a bijection f from $\mathcal{P}(S)$ and $S \to Bool$:

$$f = (\mathcal{P}(S),\ S \to Bool,\ \{(A, (S,\ Bool,\ \{(x, x \in A) \mid x \in S\})) \mid A \subseteq S\})$$

In English: f is a function that takes a subset A of S and returns a predicate p in $S \to Bool$ such that $p(x) = x \in A$. The inverse function for f is:

$$f^{-1} = (S \to Bool,\ \mathcal{P}(S),\ \{(p, \{y \mid (y \in S) \text{ and } p(y)\}) \mid p \in (S \to Bool)\})$$

In English: f^{-1} is a function that takes a predicate p in $S \to Bool$ and returns the subset of S containing exactly those elements in S for which p returns true.

To complete the proof of the bijection, we must show that no information is lost in a round-trip between A abd B:

- First we'll show that for all C in $\mathcal{P}(S)$, $f^{-1}(f(C)) = C$.

$$\begin{array}{rcll} f^{-1}(f(C)) & = & f^{-1}(q) \text{ where } q = (S,\ Bool,\ \{(x, x \in C) \mid x \in S\}) & \text{by defn. of } f \\ & = & \{y \mid (y \in S) \text{ and } q(y)\} & \text{by defn. of } f^{-1} \\ & = & C & \text{by defn. of } q \end{array}$$

- Now we'll show that for all r in $S \to Bool$, $f(f^{-1}(r)) = r$.

$$\begin{array}{rcll} f(f^{-1}(r)) & = & f(D) \text{ where } D = \{y \mid (y \in S) \text{ and } r(y)\} & \text{by defn. of } f^{-1} \\ & = & (S,\ Bool,\ \{(x, x \in D) \mid x \in S\}) & \text{by defn. of } f \\ & = & (S,\ Bool,\ \{(x, r(x)) \mid x \in S\}) & \text{by defn. of } D \\ & = & r & \text{by defn. of } r \end{array}$$

The last step holds because any function $g \in E \to F$ is equal to $(E, F, \{(z, g(z)) \mid z \in E\})$

In the above reasoning, I introduced new names (C and D for sets, q and r for predicates) to avoid any confusion with the names A and p in the definitions of f and f^{-1}. Such renaming isn't necessary, but it can help to clarify things.

Problems 9–11 have been moved to PS2.