CMPSCI 250: Introduction to Computation

Lecture #29: Proving Regular Language Identities David Mix Barrington 4 April 2014

Regular Language Identities

- Regular Language Identities
- The Semiring Axioms Again
- Identities Involving Union and Concatenation
- Proving the Distributive Law
- The Inductive Definition of Kleene Star
- Identities Involving Kleene Star
- (ST)*, S*T*, and (S + T)*

Regular Expression Identities

- In this lecture and the next we'll use our new formal definition of the regular languages to prove things about them.
- In particular, in this lecture we'll prove a number of **regular language identities**, which are statements about languages where the types of the free variables are "regular expression" and which are true for all possible values of those free variables.

Regular Expression Identities

- We can use the inductive definition of regular expressions to prove statements about the whole family of them -- this will be the subject of the next lecture.

The Semiring Axioms Again

- The set of natural numbers, with the ordinary operations + and ×, forms an algebraic structure called a **semiring**.
- Earlier we proved the semiring axioms for the naturals from the Peano axioms and our inductive definitions of + and ×.
- It turns out that the languages form a semiring under union and concatenation, and the regular languages are a **subsemiring** because they are **closed** under + and ·. That is, if R and S are regular, so are R + S and R·S.

The Semiring Axioms Again

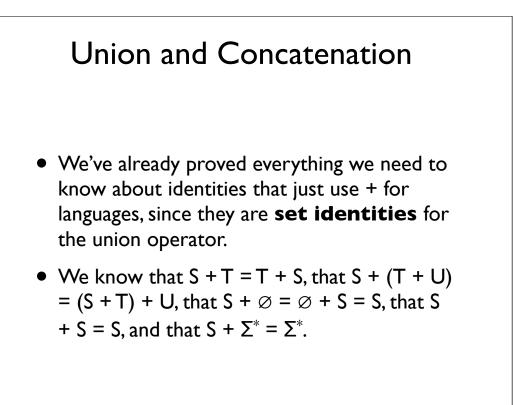
- Both operations of a semiring must be associative and each must have an identity. For languages, Ø is the identity for union and {λ} = Ø* is the identity for concatenation, as Ø + R = R + Ø = R and RØ* = Ø*R = R. We also need the distributive law which we'll prove soon.
- Note that + is commutative but · is not as in general XY and YX are different languages. There are other identities like X + X = X that are not true for the natural numbers.

Clicker Question #I

- Consider the rule "(X + 1)³ = X³ + 3X² + 3X + 1", where "1" is the identity of the semiring S, and "3" is the element 1 + 1 + 1. Which of these statements about this rule is true?
- (a) It is false unless S = {I}.
- (b) It is true only if the semiring obeys the rule "XY = YX".
- (c) It is not valid since cubing is not defined.
- (d) It is true if addition in S is commutative.

Answer #I

- Consider the rule "(X + 1)³ = X³ + 3X² + 3X + 1", where "1" is the identity of the semiring S, and "3" is the element 1 + 1 + 1. Which of these statements about this rule is true?
- (a) It is false unless S = {I}.
- (b) It is true only if the semiring obeys the rule "XY = YX".
- (c) It is not valid since cubing is not defined.
- (d) It is true if addition in S is commutative.



Union and Concatenation

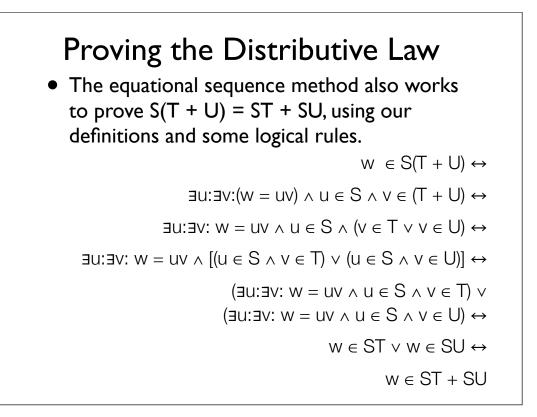
- We looked at concatenation of languages back in Chapter 2.
- Statements like S(TU) = (ST)U, SØ = ØS = Ø, and SØ^{*} = Ø^{*}S = S may be proved by the equational sequence method.
- To prove "X = Y", for example, we let w be an arbitrary string and prove $w \in X \leftrightarrow w \in Y$.

Union and Concatenation

• For example,
$$w \in (ST)U \leftrightarrow$$

$$\begin{split} \exists u: \exists z: (w = uz) \land (u \in ST) \land (z \in U) \leftrightarrow \\ \exists x: \exists y: \exists z: (w = xyz) \land (x \in S) \land (y \in T) \land (z \in U) \leftrightarrow \\ \exists x: \exists v: (w = xv) \land (x \in S) \land (v \in TU) \leftrightarrow \\ w \in S(TU). \end{split}$$

 At each stage we use the definition of concatenation of languages or the associativity of concatenation of strings, "x(yz) = (xy)z", which we've already proved.



The Inductive Definition of Star

- To prove identities about the Kleene star operation, we use its inductive definition.
- If A is any language, we define A* by three rules:
- (I) $\lambda \in A^*$,
- (2) if $u \in A^*$ and $v \in A$, then $uv \in A^*$, and
- (3) a string is only in A* if it can be proved to be so by rules (1) and (2).

The Inductive Definition of Star

- The definition we gave earlier, " $w \in A^*$ if and only if w is the concatenation of zero or more strings, each of which is in A" is equivalent.
- By induction on naturals n, we can prove that any concatenation of n strings from A is in A* according to the second definition.
- And we can prove by induction on all strings w in A* (according to the second definition) that there exists an n such that w is the concatenation of n strings from A.

Clicker Question #2

- Let Σ = {a, b, c} and let P(w), for w ∈ Σ*, be "w has an equal number of a's and b's". Let X be the language (bcac + ccabc + acbabc)*. If I want to prove that "∀w: (w ∈ X) → P(w)", what is my inductive step?
- (a) $P(X) \rightarrow (P((bcac)^*) \land P((ccabc)^*) \land P((acbabc)^*))$
- (b) $P(v) \rightarrow (P(vab) \land P(vba) \land P(vc))$
- (c) $P(\lambda) \rightarrow (P(bcac) \land P(ccabc) \land P(acbabc))$
- (d) $P(v) \rightarrow (P(vbcac) \land P(vccabc) \land P(vacbabc))$

Answer #2

- Let Σ = {a, b, c} and let P(w), for w ∈ Σ*, be "w has an equal number of a's and b's". Let X be the language (bcac + ccabc + acbabc)*. If I want to prove that "∀w: (w ∈ X) → P(w)", what is my inductive step?
- (a) $P(X) \rightarrow (P((bcac)^*) \land P((ccabc)^*) \land P((acbabc)^*))$
- (b) $P(v) \rightarrow (P(vab) \land P(vba) \land P(vc))$
- (c) $P(\lambda) \rightarrow (P(bcac) \land P(ccabc) \land P(acbabc))$
- (d) $P(v) \rightarrow (P(vbcac) \land P(vccabc) \land P(vacbabc))$

Structural Induction

- This is an example of a general phenomenon -- any of our **structural inductions** on the definition of a class could be rephrased as inductions on the naturals.
- Rather than proving P(w) for all strings w, for example, we could let Q(n) mean "P(w) for all w of length n" and then prove Q(n) for all naturals n. The proof of Q(n) → Q(n+1) would essentially be the same as the proof of P(w) → P(wa).

Identities for Kleene Star

- The statement " $(u \in A^* \land v \in A^*) \rightarrow uv \in A^*$ ", or "A* is closed under concatenation", is *not* part of the definition of Kleene star.
- It looks very much like our rule (2) which says "(u ∈ A* ∧ v ∈ A) → uv ∈ A*", but it requires a proof.
- Let's prove this closure rule by induction on all strings v in A*.

A* Closed Under Concatenation

- Our statement P(v) is " $u \in A^* \rightarrow uv \in A^*$ ", where we have let u be arbitrary.
- The base case is $v = \lambda$, and it is clear that if $u \in A^*$ and $v = \lambda$, then $uv \in A^*$ since uv = u.
- For the induction, assume that v = wx, that $w \in A^*$, that $x \in A$, and that we already know P(w), which says that $u \in A^* \rightarrow uw \in A^*$.

A* Closed Under Concatenation

- Now to prove P(v), we assume u ∈ A*, derive uw ∈ A* from the IH, and derive that uv = uwx is in A*.
- This follows from rule (2), because $uw \in A^*$ and $x \in A$.
- This should remind you of the proof that the path relation on graphs is transitive, using the inductive definition of paths.

$(ST)^*$, S^*T^* , and $(S + T)^*$

- It is generally much easier to prove subset relationships that set equalities from the Kleene star definition.
- The equality identities that are true, like (S^{*})^{*}
 = S^{*}, are most easily proved by showing both directions, here (S^{*})^{*} ⊆ S^{*} and S^{*} ⊆ (S^{*})^{*}.
- These in turn follow from the identities $T \subseteq T^*$ and $(S \subseteq T) \rightarrow (S^* \subseteq T^*)$. Both of these in turn follow from $(S \subseteq T^*) \rightarrow (S^* \subseteq T^*)$.

$(ST)^*$, S^*T^* , and $(S + T)^*$

- How shall we prove that $S \subseteq T^* \rightarrow S^* \subseteq T^*$?
- We'll assume S ⊆ T*, let P(w) be "w ∈ T*", and prove P(w) for all w in S*.
- For the base case, $w = \lambda$ and we know $\lambda \in T^*$.
- For the induction, assume w = xy with P(x)true and $y \in S$. So $X \in T^*$ by the IH, $y \in T^*$ because $S \subseteq T^*$, and then w = xy is in T^* by the closure of T^* under concatenation.

$(ST)^*$, S^*T^* , and $(S + T)^*$

- We have seen that parentheses matter, so that (ST)^{*} and S^{*}T^{*} are two different languages for most choices of S and T.
- (We saw that $(ab)^* \neq a^*b^*$, for example.)
- But we can prove that both (ST)* and S*T* are contained in (S + T)*, using the identities above.

Clicker Question #3

- Let S and T be any regular expressions. Which of these statements is guaranteed to be true?
- (a) T + S^{*} + (STS)^{*} \subseteq (S + T)^{*}
- (b) (TS)^{*} ⊆ T(ST)^{*}S
- (c) $(T^* + S^*)^* \subseteq S^* + T^*$
- (d) $((T + S)(S + T))^* = (S + T)^*(T + S)^*$

Answer #3

- Let S and T be any regular expressions.
 Which of these statements is guaranteed to be true?
- (a) $T + S^* + (STS)^* \subseteq (S + T)^*$
- (b) $(TS)^* \subseteq T(ST)^*S$ (\emptyset^* is on left, not on right)
- (c) $(T^* + S^*)^* \subseteq S^* + T^* (ST is counterexample)$
- (d) ((T + S)(S + T))* = (S + T)*(T + S)* (things in the left set must use an even number of S,T)