

CMPSCI 250: Introduction to Computation

Lecture #29: Proving Regular Language Identities
David Mix Barrington
8 November 2013

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

- For example, if we view the union operator $+$ as “addition” and the concatenation operator \cdot as “multiplication”, then the rule $S(T + U) = ST + SU$ is a statement about languages and (as we’ll prove today) is a regular language identity. In fact it’s a language identity as regularity doesn’t matter.
- 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 \times , 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 \times .
- 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 \cdot . That is, if R and S are regular, so are $R + S$ and $R \cdot S$.

The Semiring Axioms Again

- Both operations of a semiring must be associative and each must have an identity. For languages, \emptyset is the identity for union and $\{\lambda\} = \emptyset^*$ is the identity for concatenation, as $\emptyset + R = R + \emptyset = R$ and $R\emptyset^* = \emptyset^*R = R$. We also need the distributive law which we'll prove soon.
- Note that $+$ is commutative but \cdot 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 #1

- Consider the rule “ $(X + 1)^2 = X^2 + 2X + 1$ ”, where “1” is the identity of the semiring, and “2” is $1 + 1$. Which of these statements about this rule is true?
- (a) It is true only if the semiring obeys the rule “ $XY = YX$ ”.
- (b) It is not valid since squaring is not defined.
- (c) It is true in any semiring where $X+Y = Y+X$.
- (d) It is only true for one-element semirings.

Answer #1

- Consider the rule “ $(X + 1)^2 = X^2 + 2X + 1$ ”, where “1” is the identity of the semiring, and “2” is $1 + 1$. Which of these statements about this rule is true?
- (a) It is true only if the semiring obeys the rule “ $XY = YX$ ”.
- (b) It is not valid since squaring is not defined.
- (c) *It is true in any semiring where $X+Y = Y+X$.*
- (d) It is only true for one-element semirings.

Union and Concatenation

- We've already proved everything we need to know about identities that just use $+$ for languages, since they are **set identities** for the union operator.
- We know that $S + T = T + S$, that $S + (T + U) = (S + T) + U$, that $S + \emptyset = \emptyset + S = S$, that $S + S = S$, and that $S + \Sigma^* = \Sigma^*$.

Union and Concatenation

- We looked at concatenation of languages back in Chapter 2.
- Statements like $S(TU) = (ST)U$, $S\emptyset = \emptyset S = \emptyset$, and $S\emptyset^* = \emptyset^*S = S$ 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, $w \in (ST)U \leftrightarrow$
 $\exists u:\exists z:(w = uz) \wedge (u \in ST) \wedge (z \in U) \leftrightarrow$
 $\exists x:\exists y:\exists z:(w = xyz) \wedge (x \in S) \wedge (y \in T) \wedge (z \in U) \leftrightarrow$
 $\exists x:\exists v:(w = xv) \wedge (x \in S) \wedge (v \in TU) \leftrightarrow$
 $w \in S(TU).$
- 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.

Proving the Distributive Law

- The equational sequence method also works to prove $S(T + U) = ST + SU$, using our definitions and some logical rules.

$$\begin{aligned}w \in S(T + U) &\leftrightarrow \\ \exists u:\exists v:(w = uv) \wedge u \in S \wedge v \in (T + U) &\leftrightarrow \\ \exists u:\exists v: w = uv \wedge u \in S \wedge (v \in T \vee v \in U) &\leftrightarrow \\ \exists u:\exists v: w = uv \wedge [(u \in S \wedge v \in T) \vee (u \in S \wedge v \in U)] &\leftrightarrow \\ (\exists u:\exists v: w = uv \wedge u \in S \wedge v \in T) \vee & \\ (\exists u:\exists v: w = uv \wedge u \in S \wedge v \in U) &\leftrightarrow \\ w \in ST \vee w \in SU &\leftrightarrow \\ w \in ST + SU &\end{aligned}$$

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:
- (1) $\lambda \in A^*$,
- (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 $\Sigma = \{a, b, c\}$ and let $P(w)$, for $w \in \Sigma^*$, be “ w has twice as many a’s as it has b’s”. Let X be the language $(baac + ccaba + aababa)^*$. If I want to prove “ $\forall w: (w \in X) \rightarrow P(w)$ ”, what is my inductive step?
- (a) $P(\lambda) \rightarrow (P(baac) \wedge P(ccaba) \wedge P(aababa))$
- (b) $P(X) \rightarrow (P((baac)^*) \wedge P((ccaba)^*) \wedge P((aababa)^*))$
- (c) $P(v) \rightarrow (P(vaab) \wedge P(vaba) \wedge P(vbaa))$
- (d) $P(v) \rightarrow (P(vbaac) \wedge P(vccaba) \wedge P(vaababa))$

Answer #2

- Let $\Sigma = \{a, b, c\}$ and let $P(w)$, for $w \in \Sigma^*$, be “ w has twice as many a’s as it has b’s”. Let X be the language $(baac + ccaba + aababa)^*$. If I want to prove “ $\forall w: (w \in X) \rightarrow P(w)$ ”, what is my inductive step?
- (a) $P(\lambda) \rightarrow (P(baac) \wedge P(ccaba) \wedge P(aababa))$
- (b) $P(X) \rightarrow (P((baac)^*) \wedge P((ccaba)^*) \wedge P((aababa)^*))$
- (c) $P(v) \rightarrow (P(vaab) \wedge P(vaba) \wedge P(vbaa))$
- (d) $P(v) \rightarrow (P(vbaac) \wedge P(vccaba) \wedge P(vaababa))$

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 $Q(n) \rightarrow Q(n+1)$ would essentially be the same as the proof of $P(w) \rightarrow P(wa)$.

Identities for Kleene Star

- The statement “ $(u \in A^* \wedge v \in A^*) \rightarrow uv \in A^*$ ”, 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 “ $(u \in A^* \wedge v \in A) \rightarrow uv \in A^*$ ”, 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 $u \in A^*$, derive $uw \in A^*$ 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 than set equalities from the Kleene star definition.
- The equality identities that are true, like $(S^*)^* = S^*$, are most easily proved by showing both $(S^*)^* \subseteq S^*$ and $S^* \subseteq (S^*)^*$.
- 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 $S \subseteq T^*$, let $P(w)$ be " $w \in T^*$ ", 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) $S + T^* \subseteq (S + T)^*$
- (b) $(S^* + T^*)^* \subseteq S^* + T^*$
- (c) $((S + T)(S + T))^* = (S + T)^*(S + T)^*$
- (d) $(ST)^* \subseteq S(TS)^*T$

Clicker Question #3

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