Formal Languages and Automata

7 lectures for

University of Cambridge

2015 Computer Science Tripos

Part IA Discrete Mathematics

by Prof. Andrew Pitts

Syllabus for this part of the course

- Inductive definitions using rules and proofs by rule induction.
- Abstract syntax trees.
- Regular expressions and pattern matching.
- Finite automata and regular languages: Kleene's theorem.
- The Pumping Lemma.

mathematics needed for <u>computer science</u>

Common theme: mathematical techniques for defining formal languages and reasoning about their properties.

Key concepts: inductive definitions, automata

Relevant to:

- Part IB: Compiler Construction, Computation Theory, Complexity Theory, Semantics of Programming Languages
- Part II: Natural Language Processing, Optimising Compilers, Denotational Semantics, Temporal Logic and Model Checking

N.B. we do not cover the important topic of context-free grammars, which prior to 2013/14 was part of the CST IA course *Regular Languages and Finite Automata* that has been subsumed into this course.

see course web page for relevant Tripos questions

Formal Languages

Alphabets

An **alphabet** is specified by giving a finite set, Σ, whose elements are called **symbols**. For us, any set qualifies as a possible alphabet, so long as it is finite.

Examples:

- $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, $\mathbf{10}$-element set of decimal digits.
- $\{a, b, c, \ldots, x, y, z\}$, $\mathbf{26}$-element set of lower-case characters of the English language.
- $\{S \mid S \subseteq \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}\}$, $\mathbf{2^{10}}$-element set of all subsets of the alphabet of decimal digits.

Non-example:

- $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$, set of all non-negative whole numbers is not an alphabet, because it is infinite.

Strings over an alphabet

A **string of length** n (for $n = 0, 1, 2, \ldots$) over an alphabet Σ is just an ordered n-tuple of elements of Σ, written without punctuation.

Σ^* denotes set of all strings over Σ of any finite length.

Examples:

notation for the string of length 0

- If $\Sigma = \{a, b, c\}$, then ε, a, ab, aac, and $bbac$ are strings over Σ of lengths zero, one, two, three and four respectively.
- If $\Sigma = \{a\}$, then Σ^* contains ε, a, aa, aaa, $aaaa$, etc.

In general, a^n denotes the string of length n just containing a symbols

Strings over an alphabet

A **string of length** n (for $n = 0, 1, 2, \ldots$) over an alphabet Σ is just an ordered n-tuple of elements of Σ, written without punctuation.

Σ^* denotes set of all strings over Σ of any finite length.

Examples:

- If $\Sigma = \{a, b, c\}$, then ε, a, ab, aac, and $bbac$ are strings over Σ of lengths zero, one, two, three and four respectively.
- If $\Sigma = \{a\}$, then Σ^* contains ε, a, aa, aaa, $aaaa$, etc.
- If $\Sigma = \emptyset$ (the empty set), then what is Σ^*?

Strings over an alphabet

A **string of length** n (for $n = 0, 1, 2, \ldots$) over an alphabet Σ is just an ordered n-tuple of elements of Σ, written without punctuation.

Σ^* denotes set of all strings over Σ of any finite length.

Examples:

- If $\Sigma = \{a, b, c\}$, then ε, a, ab, aac, and $bbac$ are strings over Σ of lengths zero, one, two, three and four respectively.
- If $\Sigma = \{a\}$, then Σ^* contains ε, a, aa, aaa, $aaaa$, etc.
- If $\Sigma = \emptyset$ (the empty set), then $\Sigma^* = \{\varepsilon\}$.

Concatenation of strings

The **concatenation** of two strings u and v is the string uv obtained by joining the strings end-to-end. This generalises to the concatenation of three or more strings.

Examples:

If $\Sigma = \{a, b, c, \ldots, z\}$ and $u, v, w \in \Sigma^*$ are $u = ab$, $v = ra$ and $w = cad$, then

$$\begin{aligned} vu &= raab \\ uu &= abab \\ wv &= cadra \\ uvwuv &= abracadabra \end{aligned}$$

N.B.

Concatenation of strings

The **concatenation** of two strings u and v is the string uv obtained by joining the strings end-to-end. This generalises to the concatenation of three or more strings.

Examples:

If $\Sigma = \{a, b, c, \ldots, z\}$ and $u, v, w \in \Sigma^*$ are $u = ab$, $v = ra$ and $w = cad$, then

$$vu = raab$$
$$uu = abab$$
$$wv = cadra$$
$$uvwuv = abracadabra$$

N.B. $(uv)w = uvw = u(vw)$ (any u, v, w)

$u\varepsilon = u = \varepsilon u$

Formal languages

An extensional view of what constitutes a formal language is that it is completely determined by the set of ‘words in the dictionary’:

Given an alphabet Σ, we call any subset of Σ^* a (formal) **language** over the alphabet Σ.

We will use inductive definitions to describe languages in terms of grammatical rules for generating subsets of Σ^*.

Inductive Definitions

Axioms and rules

for inductively defining a subset of a given set U

- **axioms** $\dfrac{}{a}$ are specified by giving an element a of U
- **rules** $\dfrac{h_1 \; h_2 \; \cdots \; h_n}{c}$

  are specified by giving a finite subset $\{h_1, h_2, \ldots, h_n\}$ of U (the **hypotheses** of the rule) and an element c of U (the **conclusion** of the rule)

Derivations

Given a set of axioms and rules for inductively defining a subset of a given set U, a **derivation** (or proof) that a particular element $u \in U$ is in the subset is by definition

> a finite rooted tree with vertexes labelled by elements of U and such that:
>
> - the root of the tree is u (the conclusion of the whole derivation),
> - each vertex of the tree is the conclusion of a rule whose hypotheses are the children of the node,
> - each leaf of the tree is an axiom.

Example

$U = \{a, b\}^*$

axiom: $\dfrac{}{\varepsilon}$

rules: $\dfrac{u}{aub} \qquad \dfrac{u}{bua} \qquad \dfrac{u \quad v}{uv}$ (for all $u, v \in U$)

Example derivations:

$$\dfrac{\dfrac{\varepsilon}{ab} \qquad \dfrac{\dfrac{\varepsilon}{ab}}{aabb}}{abaabb} \qquad\qquad \dfrac{\dfrac{\dfrac{\varepsilon}{ba} \qquad \dfrac{\varepsilon}{ab}}{baab}}{abaabb}$$

Inductively defined subsets

Given a set of axioms and rules over a set U, the subset of U **inductively defined** by the axioms and rules consists of all and only the elements $u \in U$ for which there is a derivation with conclusion u.

For example, for the axioms and rules on Slide 15

- $abaabb$ is in the subset they inductively define (as witnessed by either derivation on that slide)
- $abaab$ is not in that subset (there is no derivation with that conclusion – why?)

(In fact $u \in \{a, b\}^*$ is in the subset iff it contains the same number of a and b symbols.)

Example: transitive closure

Given a binary relation $R \subseteq X \times X$ on a set X, its **transitive closure** R^+ is the smallest (for subset inclusion) binary relation on X which contains R and which is **transitive** ($\forall x, y, z \in X.\ (x, y) \in R^+ \ \&\ (y, z) \in R^+ \Rightarrow (x, z) \in R^+$).

R^+ is equal to the subset of $X \times X$ inductively defined by

axioms $\dfrac{}{(x, y)}$ (for all $(x, y) \in R$)

rules $\dfrac{(x, y) \qquad (y, z)}{(x, z)}$ (for all $x, y, z \in X$)

Example: reflexive-transitive closure

Given a binary relation $R \subseteq X \times X$ on a set X, its **reflexive-transitive closure** R^* is defined to be the smallest binary relation on X which contains R, is both transitive and **reflexive** $(\forall x \in X.\ (x, x) \in R^*)$.

R^* is equal to the subset of $X \times X$ inductively defined by

axioms $\dfrac{}{(x, y)}$ (for all $(x, y) \in R$) $\dfrac{}{(x, x)}$ (for all $x \in X$)

rules $\dfrac{(x, y) \qquad (y, z)}{(x, z)}$ (for all $x, y, z \in X$)

Example: reflexive-transitive closure

Given a binary relation $R \subseteq X \times X$ on a set X, its **reflexive-transitive closure** R^* is defined to be the smallest binary relation on X which contains R, is both transitive and **reflexive** ($\forall x \in X.\ (x, x) \in R^*$).

R^* is equal to the subset of $X \times X$ inductively defined by

axioms $\dfrac{}{(x, y)}$ (for all $(x, y) \in R$) $\qquad \dfrac{}{(x, x)}$ (for all $x \in X$)

rules $\dfrac{(x, y) \qquad (y, z)}{(x, z)}$ (for all $x, y, z \in X$)

we can use Rule Induction (Slide 20) to prove this

Rule Induction

Theorem. The subset $I \subseteq U$ inductively defined by a collection of axioms and rules is closed under them and is the least such subset: if $S \subseteq U$ is also closed under the axioms and rules, then $I \subseteq S$.

Given axioms and rules for inductively defining a subset of a set U, we say that a subset $S \subseteq U$ is **closed under the axioms and rules** if

- for every axiom $\dfrac{}{a}$, it is the case that $a \in S$
- for every rule $\dfrac{h_1 \; h_2 \; \cdots \; h_n}{c}$, if $h_1, h_2, \ldots, h_n \in S$, then $c \in S$.

Rule Induction

Theorem. The subset $I \subseteq U$ inductively defined by a collection of axioms and rules is closed under them and is the least such subset: if $S \subseteq U$ is also closed under the axioms and rules, then $I \subseteq S$.

We use the theorem as method of proof: given a property $P(u)$ of elements of U, to prove $\forall u \in I.\, P(u)$ it suffices to show

- **base cases:** $P(a)$ holds for each axiom $\dfrac{}{a}$
- **induction steps:** $P(h_1) \,\&\, P(h_2) \,\&\, \cdots \,\&\, P(h_n) \Rightarrow P(c)$ holds for each rule $\dfrac{h_1 \; h_2 \; \cdots \; h_n}{c}$

(To see this, apply the theorem with $S = \{u \in U \mid P(u)\}$.)

Example: reflexive-transitive closure

Given a binary relation $R \subseteq X \times X$ on a set X, its **reflexive-transitive closure** R^* is defined to be the smallest binary relation on X which contains R, is both transitive and **reflexive** ($\forall x \in X.\ (x,x) \in R^*$).

R^* is equal to the subset of $X \times X$ inductively defined by

axioms $\dfrac{}{(x,y)}$ (for all $(x,y) \in R$) $\dfrac{}{(x,x)}$ (for all $x \in X$)

rules $\dfrac{(x,y) \qquad (y,z)}{(x,z)}$ (for all $x,y,z \in X$)

we can use Rule Induction (Slide 20) to prove this, since $S \subseteq X \times X$ being closed under the axioms & rules is the same as it containing R, being reflexive and being transitive.