CHAPTER 11: DECIDABILITY *

Peter Cappello
Department of Computer Science
University of California, Santa Barbara
Santa Barbara, CA 93106
cappello@cs.ucsb.edu

• The corresponding textbook chapter should be read before attending this lecture.

• These notes are not intended to be complete. They are supplemented with figures, and other material that arises during the lecture period in response to questions.

*Based on Theory of Computing, 2nd Ed., D. Cohen, John Wiley & Sons, Inc.
DEFINITIONS

• A problem is **effectively solvable** if there is an *algorithm* for solving it (a procedure that completes after finitely many steps, the maximum of which is known in advance, but may depend on the size of the input).

• A problem whose solution is “yes” or “no” is a **decision problem**.

• An effective solution to a decision problem is a **decision procedure**.

• A decision problem that has a decision procedure is **decidable**.
**Theorem:** Let $A$ be an FA. The question “Is $L(A) = \emptyset$?” is decidable.

**Proof:**

1. $L(A) \neq \emptyset$ if and only if there is a path from $A$’s start state to some final state.

2. The following algorithm returns true if and only if there is a path from $A$’s start state to some final state.
boolean isEmpty(FA A) {
    paint $q_0$ blue;
    set.put($q_0$);
    while (! set.isEmpty() ) {
        $p = set.remove();$
        For each $a \in \Sigma$, 
        if ( $p' = \delta(p, a)$ is not blue ) {
            paint $p'$ blue;
            set.put($p'$);
        }
    }
    return ( there is a blue final state ) ? false : true;
}
**Theorem:** Let $A$ be an FA with $n$ states. If $L(A) \neq \emptyset$, $A$ accepts a word $w$, $|w| < n$.

**Proof:**

1. The shortest path from $A$’s start state to some final state, if any, can be no longer than $n - 1$: It cannot involve a circuit.
2. The concatenation of arc labels consists of less than $n$ letters.

Thus, “Is $L(A) = \emptyset$?” also can be answered by running $A$ on no more than

$$m^{n-1} + m^{n-2} + \cdots + m^0$$

words, where $m = |\Sigma|$.
**Theorem**: Let $A$ and $B$ be FA accepting $L_A$ and $L_B$, respectively, and $E$ and $F$ be regular expressions. The following questions are decidable:

1. Is $L_A = \emptyset$?
2. Is $L_A = L_B$?
3. Is $E$ equivalent to $F$ (i.e., do they denote the same language)?

**Proof**: 

1. This follows from our previous theorem.
2. $L_A = L_B \iff (L_A \cap \overline{L_B}) \cup (L_B \cap \overline{L_A}) = \emptyset$.
3. For each regular expression, construct an equivalent FA, using Kleene’s theorem. Use part 2 above to see if these 2 FA accept the same language.
**Finiteness**

**Theorem:** Let $R$ be a regular expression. $|L(R)| = \infty \iff R$ has a Kleene star that applies to something other than $\Lambda$.

**Proof:**

1. If $R$ has no Kleene star operator, it denotes a finite set.
2. $\Lambda^* = \Lambda$.
3. If the Kleene star operator is applied to something not equivalent to $\Lambda$, the resulting set is infinite.
**Theorem:** Let $A$ be an FA with $n$ states.

$L(A) = \infty \iff \exists w \in L(A), n \leq |w| < 2n.$

**Proof:**

1. If $\exists w \in L(A), n \leq |w| < 2n$, then $L(A) = \infty$.
   
   This follows from the pumping lemma: If there is any word $w$, $|w| \geq n$, then $w = xyz$, such that $xy^kz \in L(A), \forall k > 0$.

2. If $L(A) = \infty$ then $\exists w \in L(A), n \leq |w| < 2n$.

   (a) By the pumping lemma, $\exists w \in L(A), w = xyz$, and $|xy| \leq n$.
   (b) Thus, $|y| \leq n$.
   (c) Assume without loss of generality that the part of the accepting path associated with $x$ and $z$ do not contain any loops.
   (d) $xz \in L(A)$, and $|xz| < n$.
   (e) Let $k$ be the *smallest* exponent that makes $|xy^kz| \geq n$.
   
   Then, $|xy^kz| < 2n$. 

8
**Theorem:** Given FA $A$, the question “Is $L(A) = \infty$?” is decidable.

**Proof:**

1. Check each word $w$, $n \leq |w| < 2n$.
2. If any are accepted, $L(A) = \infty$; otherwise, $L(A) < \infty$.

When you study DFS, you should conceive of faster ways to test for finiteness.

However, the proof above, requires only that:

- you know how many states the FA has;
- you know if $w \in L(A)$, for any word $w$.

You do not need to be able to *examine* the FA.