Chapter 6: Transition Graphs *

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.
A transition graph is defined by a 5-tuple:

- A finite set of states, \( Q \).
- A finite set of input symbols, \( \Sigma \).
- A non-empty set of start states, \( S \subseteq Q \).
- A set of final or accepting states \( F \subseteq Q \).
- A finite set, \( \delta \) of transitions, (directed edge labels) \( (u, s, v) \), where \( u, v \in Q \) and \( s \in \Sigma^* \).

Illustrate.
The Language Accepted by a Transition Graph

• Let \( A = (Q, \Sigma, S, F, \delta) \) be a transition graph.

• A successful path in \( A \) is one that starts in some start state and ends in some accepting state of \( A \).

• Let \( P \) be the set of all successful paths in \( A \).

• Let \( L \) be the set of words that are the concatenation of the sequence of edge labels of \( A \) corresponding to some successful path in \( A \).

• The language accepted by \( A \), denoted \( L(A) = L \).

Illustrate.
Transition Graphs: Some Observations

• If there is no factoring of a word $w$ that is the concatenation of edge labels of a successful path in $A$, then $w \notin L(A)$.

• Every finite automaton can be viewed as a transition graph.

• Since the reverse is not true, transition graphs generalize finite automata.
Illustrate transition graphs for the following building blocks:

- $L(A) = \emptyset$
- $L(A) = \{\Lambda\}$
- $L(A) = \Sigma$
- $L(A) = L(B)L(C)$, for transition graphs $B$ and $C$.
- $L(A) = L(B) \cup L(C)$, for transition graphs $B$ and $C$.
- $L(A) = \overline{L(B)}$, for transition graph $B$. 
Transition Graphs: Some Observations

• Every finite language is accepted by some transition graph. Illustrate.
• Given a transition graph $A$, it is unclear how to determine $L(A)$.
• We see soon why transitions graphs are introduced.
Generalized Transition Graphs

A generalized transition graph is defined by a 5-tuple:

• A finite set of states, \( Q \).
• A finite set of input symbols, \( \Sigma \).
• A non-empty set set of start states, \( S \subseteq Q \).
• A set of final or accepting states \( F \subseteq Q \).
• A finite set, \( \delta \) of transitions, (directed edge labels) \( (u, s, v) \), where \( u, v \in Q \) and \( s \) is a regular expression over \( \Sigma \).

Illustrate.
Generalized transition graphs are nondeterministic: Given a state and a [partially consumed] input, there may be more than 1 possible successor state.
∀ TG, ∃ an equivalent GTG with 1 final state

Proof: (Illustrate)


2. $A$ has 0, 1, or more than 1 final state.

   **Case 0 final states** Add a final state with no transition to it.

   **Case 1 final state** Do nothing; the given transition graph has the desired property.

   **Case more than 1 final state**

     (a) Add state $f$ to $Q'$.
(b) \( F' = \{ f \} \).
(c) For each \( s \in F \), add a \( \Lambda \)-transition from \( s \) to \( f' \).

3. Every string that could reach a final state in \( A \) can reach a final state in \( A' \), using a \( \Lambda \)-transition: \( L(A) \subseteq L(A') \).

4. Every string that can reach a final state in \( A' \) must first reach a final state in \( A \): \( L(A') \subseteq L(A) \).

5. Thus, \( L(A') = L(A) \).
∀ TG \ A, \ \exists \ A \ GTG \ A', \ L(A') = L^+(A).

Proof: (Illustrate)

1. Construct a new GTG, \ A'. Initially, \ A' ← \ A.

2. For each final state in \ A, add a Λ-transition in \ A' from it to the start state.

3. If a substring reaches a final state in \ A, it can reach the start state in \ A': L^+(A) ⊆ L(A').

4. Since the only transitions that are added are Λ-transitions from a final state to the start state, \ w ∈ L(A') ⇒ w = w_1 \cdots w_i \cdots w_k, where \ w_i ∈ L(A), for 1 ≤ i ≤ k: L(A') ⊆ L^+(A).

5. Thus, \ L(A') = L^+(A).