Supplementary Lecture B

Collapsing Nondeterministic Automata

With respect to minimization, the situation for nondeterministic automata is not as satisfactory as that for deterministic automata. For example, minimal NFAs are not necessarily unique up to isomorphism (Miscellaneous Exercise 60). However, part of the Myhill–Nerode theory developed in Lectures 13 through 16 does generalize to NFAs. The generalization is based on the notion of *bisimulation*, an important concept in the theory of concurrency [87]. In this lecture we briefly investigate this connection.

The version of bisimulation we consider here is called *strong bisimulation* in the concurrency literature. There are weaker forms that apply too. We show that bisimulation relations between nondeterministic automata and collapsing relations on deterministic automata are strongly related. The former generalize the latter in two significant ways: they work for nondeterministic automata, and they can relate two different automata.

Bisimulation

Let

$$M = (Q_M,\ \Sigma,\ \Delta_M,\ S_M,\ F_M),$$
$$N = (Q_N,\ \Sigma,\ \Delta_N,\ S_N,\ F_N)$$

be two NFAs. Recall that for NFAs, $\Delta(p, a)$ is a set of states.

D. C. Kozen, *Automata and Computability*

Let $\approx$ be a binary relation relating states of M with states of N; that is, $\approx$ is a subset of $Q_M \times Q_N$. For $B \subseteq Q_N$, define

$$C_{\approx}(B) \stackrel{\text{def}}{=} \{p \in Q_M \mid \exists q \in B\ p \approx q\},$$

the set of all states of M that are related via $\approx$ to some state in B. Similarly, for $A \subseteq Q_M$, define

$$C_{\approx}(A) \stackrel{\text{def}}{=} \{q \in Q_N \mid \exists p \in A\ p \approx q\}.$$

The relation $\approx$ can be extended in a natural way to *subsets* of Q_M and Q_N: for $A \subseteq Q_M$ and $B \subseteq Q_N$,

$$\begin{aligned} A \approx B &\stackrel{\text{def}}{\iff} A \subseteq C_{\approx}(B) \text{ and } B \subseteq C_{\approx}(A) \\ &\iff \forall p \in A\ \exists q \in B\ p \approx q \text{ and } \forall q \in B\ \exists p \in A\ p \approx q. \end{aligned} \tag{B.1}$$

Note that $\{p\} \approx \{q\}$ iff $p \approx q$ and that $B \subseteq B'$ implies $C_{\approx}(B) \subseteq C_{\approx}(B')$.

Definition B.1 The relation $\approx$ is called a *bisimulation* if the following three conditions are met:

(i) $S_M \approx S_N$;

(ii) if $p \approx q$, then for all $a \in \Sigma$, $\Delta_M(p,a) \approx \Delta_N(q,a)$; and

(iii) if $p \approx q$, then $p \in F_M$ iff $q \in F_N$. □

Note the similarity of these conditions to the defining conditions of collapsing relations on DFAs from Lecture 13.

We say that M and N are *bisimilar* if there exists a bisimulation between them. The *bisimilarity class* of M is the family of all NFAs that are bisimilar to M. We will show that bisimilar automata accept the same set and that every bisimilarity class contains a unique minimal NFA that can be obtained by a collapsing construction.

First let's establish some basic consequences of Definition B.1.

Lemma B.2 *(i) Bisimulation is symmetric: if $\approx$ is a bisimulation between M and N, then its reverse*

$$\{(q,p) \mid p \approx q\}$$

is a bisimulation between N and M.

(ii) Bisimulation is transitive: if $\approx_1$ is a bisimulation between M and N and $\approx_2$ is a bisimulation between N and P, then their composition

$$\approx_1 \circ \approx_2 \stackrel{\text{def}}{=} \{(p,r) \mid \exists q\ p \approx_1 q \textit{ and } q \approx_2 r\}$$

is a bisimulation between M and P.

(iii) The union of any nonempty family of bisimulations between M and N is a bisimulation between M and N.

Proof. All three properties follow quite easily from the definition of bisimulation. We argue (iii) explicitly.

Let $\{\approx_i \mid i \in I\}$ be a nonempty indexed set of bisimulations between M and N. Define

$$\approx \stackrel{\text{def}}{=} \bigcup_{i \in I} \approx_i .$$

Thus

$$p \approx q \iff \exists i \in I\ p \approx_i q.$$

Since I is nonempty, $S_M \approx_i S_N$ for some $i \in I$, therefore $S_M \approx S_N$. If $p \approx q$, then for some $i \in I$, $p \approx_i q$. Therefore, $\Delta(p,a) \approx_i \Delta(q,a)$ and $\Delta(p,a) \approx \Delta(q,a)$. Finally, if $p \approx q$, then $p \approx_i q$ for some $i \in I$, whence $p \in F_M$ iff $q \in F_N$. □

Lemma B.3 *Let $\approx$ be a bisimulation between M and N. If $A \approx B$, then for all $x \in \Sigma^*$, $\widehat{\Delta}_M(A,x) \approx \widehat{\Delta}_N(B,x)$.*

Proof. Suppose $A \approx B$. For $x = \epsilon$,

$$\widehat{\Delta}_M(A,\epsilon) = A \approx B = \widehat{\Delta}_M(B,\epsilon).$$

For $x = a \in \Sigma$, since $A \subseteq C_\approx(B)$, if $p \in A$ then there exists $q \in B$ such that $p \approx q$. By Definition B.1(ii),

$$\Delta_M(p,a) \subseteq C_\approx(\Delta_N(q,a)) \subseteq C_\approx(\widehat{\Delta}_N(B,a)).$$

Therefore,

$$\widehat{\Delta}_M(A,a) = \bigcup_{p \in A} \Delta_M(p,a) \subseteq C_\approx(\widehat{\Delta}_N(B,a)).$$

By a symmetric argument, $\widehat{\Delta}_N(B,a) \subseteq C_\approx(\widehat{\Delta}_M(A,a))$. Therefore,

$$\widehat{\Delta}_M(A,a) \approx \widehat{\Delta}_N(B,a). \tag{B.2}$$

Proceeding by induction, suppose that $\widehat{\Delta}_M(A,x) \approx \widehat{\Delta}_N(B,x)$. By (B.2) and Lemma 6.1,

$$\begin{aligned}\widehat{\Delta}_M(A,xa) &= \widehat{\Delta}_M(\widehat{\Delta}_M(A,x),a) \\ &\approx \widehat{\Delta}_N(\widehat{\Delta}_N(B,x),a) \\ &= \widehat{\Delta}_N(B,xa).\end{aligned}$$

□

Theorem B.4 *Bisimilar automata accept the same set.*

Proof. Suppose $\approx$ is a bisimulation between M and N. By Definition B.1(i) and Lemma B.3, for any $x \in \Sigma^*$, $\widehat{\Delta}_M(S_M, x) \approx \widehat{\Delta}_N(S_N, x)$. By Definition B.1(iii), $\widehat{\Delta}_M(S_M, x) \cap F_M \neq \varnothing$ iff $\widehat{\Delta}_N(S_N, x) \cap F_N \neq \varnothing$. By definition of acceptance for nondeterministic automata, $x \in L(M)$ iff $x \in L(N)$. Since x is arbitrary, $L(M) = L(N)$. □

In fact, one can show that if M and N are bisimilar, then (B.1) is a bisimulation between the deterministic automata obtained from M and N by the subset construction (Miscellaneous Exercise 64).

As with the deterministic theory, minimization involves elimination of inaccessible states and collapsing. Here's how we deal with accessibility. Let $\approx$ be a bisimulation between M and N. The *support* of $\approx$ in M is the set $C_{\approx}(Q_N)$, the set of states of M that are related by $\approx$ to some state of N.

Lemma B.5 *A state of M is in the support of all bisimulations involving M if and only if it is accessible.*

Proof. Let $\approx$ be an arbitrary bisimulation between M and another automaton. By Definition B.1(i), every start state of M is in the support of $\approx$; and by Definition B.1(ii), if p is in the support of $\approx$, then every element of $\Delta(p, a)$ is in the support of $\approx$ for every $a \in \Sigma$. It follows inductively that every accessible state of M is in the support of $\approx$.

Conversely, it is not difficult to check that the relation

$$\{(p, p) \mid p \text{ is accessible}\} \tag{B.3}$$

is a bisimulation between M and itself. If a state is in the support of all bisimulations, then it must be in the support of (B.3), therefore accessible. □

Autobisimulation

Definition B.6 An *autobisimulation* is a bisimulation between an automaton and itself. □

Theorem B.7 *Any nondeterministic automaton M has a coarsest autobisimulation $\equiv_M$. The relation $\equiv_M$ is an equivalence relation.*

Proof. Let B be the set of all autobisimulations on M. The set B is nonempty, since it contains the identity relation at least. Let $\equiv_M$ be the union of all the relations in B. By Lemma B.2(iii), $\equiv_M$ is itself in B and is refined by every element of B. The relation $\equiv_M$ is reflexive, since the identity relation is in B, and is symmetric and transitive by Lemma B.2(i) and (ii). □

We can now remove inaccessible states and collapse by the maximal autobisimulation to get a minimal NFA bisimilar to the original NFA. Let

$$M = (Q,\ \Sigma,\ \Delta,\ S,\ F).$$

We have already observed that the accessible subautomaton of M is bisimilar to M under the bisimulation (B.3), so we can assume without loss of generality that M has no inaccessible states. Let $\equiv$ be $\equiv_M$, the maximal autobisimulation on M. For $p \in Q$, let $[p]$ denote the $\equiv$-equivalence class of p, and let $\gtrsim$ be the relation relating p to its $\equiv$-equivalence class:

$$[p] \stackrel{\text{def}}{=} \{q \mid p \equiv q\},$$
$$\gtrsim \stackrel{\text{def}}{=} \{(p, [p]) \mid p \in Q\}.$$

For any $A \subseteq Q$, define

$$A' \stackrel{\text{def}}{=} \{[p] \mid p \in A\}. \tag{B.4}$$

Lemma B.8 *For all $A, B \subseteq Q$,*

(i) $A \subseteq C_{\equiv}(B) \iff A' \subseteq B'$,

(ii) $A \equiv B \iff A' = B'$, and

(iii) $A \gtrsim A'$.

These properties are straightforward consequences of the definitions and are left as exercises (Miscellaneous Exercise 62).

Now define the quotient automaton

$$M' \stackrel{\text{def}}{=} (Q',\ \Sigma,\ \Delta',\ S',\ F'),$$

where Q', S', and F' refer to (B.4) and

$$\Delta'([p], a) \stackrel{\text{def}}{=} \Delta(p, a)'.$$

The function Δ' is well defined, because

$$\begin{aligned} [p] = [q] &\Rightarrow p \equiv q \\ &\Rightarrow \Delta(p,a) \equiv \Delta(q,a) && \text{Definition B.1(ii)} \\ &\Rightarrow \Delta(p,a)' = \Delta(q,a)' && \text{Lemma B.8(ii).} \end{aligned}$$

Lemma B.9 *The relation $\gtrsim$ is a bisimulation between M and M'.*

Proof. By Lemma B.8(iii), we have $S \gtrsim S'$, and if $p \gtrsim [q]$, then $p \equiv q$. Therefore,

$$\Delta(p,a) \gtrsim \Delta(p,a)' = \Delta'([p],a) = \Delta'([q],a).$$

This takes care of start states and transitions. For the final states, if $p \in F$, then $[p] \in F'$. Conversely, if $[p] \in F'$, there exists $q \in [p]$ such that $q \in F$; then $p \equiv q$, therefore $p \in F$. □

By Theorem B.4, M and M' accept the same set.

Lemma B.10 *The only autobisimulation on M' is the identity relation $=$.*

Proof. Let $\sim$ be an autobisimulation on M'. If $\sim$ related two distinct states, then the composition

$$\gtrsim \circ \sim \circ \lesssim, \tag{B.5}$$

where $\lesssim$ is the reverse of $\gtrsim$, would relate two non-$\equiv_M$-equivalent states of M, contradicting the maximality of $\equiv_M$. Thus $\sim$ is a subset of the identity relation.

On the other hand, if there is a state $[p]$ of M' that is not related to itself by $\sim$, then the state p of M is not related to any state of M under (B.5), contradicting Lemma B.5 and the assumption that all states of M are accessible. □

Theorem B.11 *Let M be an NFA with no inaccessible states and let $\equiv_M$ be the maximal autobisimulation on M. The quotient automaton M' is the minimal automaton bisimilar to M and is unique up to isomorphism.*

Proof. To show this, it will suffice to show that for any automaton N bisimilar to M, if we remove inaccessible states and then collapse the resulting NFA by its maximal autobisimulation, we obtain an automaton isomorphic to M'.

Using (B.3), we can assume without loss of generality that N has no inaccessible states. Let $\equiv_N$ be the maximal autobisimulation on N, and let N' be the quotient automaton.

By Lemmas B.2 and B.9, M' and N' are bisimilar. We will show that any bisimulation between M' and N' gives a one-to-one correspondence between the states of M' and N'. This establishes the result, since a bisimulation that is a one-to-one correspondence constitutes an isomorphism (Miscellaneous Exercise 63).

Let $\approx$ be a bisimulation between M' and N'. Under $\approx$, every state of M' is related to at least one state of N', and every state of N' is related to at most one state of M'; otherwise the composition of $\approx$ with its reverse would not be the identity on M', contradicting Lemma B.10. Therefore, $\approx$ embeds M' into N' injectively (i.e., in a one-to-one fashion). By a symmetric argument, the reverse of $\approx$ embeds N' into M' injectively. Therefore, $\approx$ gives a one-to-one correspondence between the states of M' and N'. □

An Algorithm

Here is an algorithm for computing the maximal bisimulation between any given pair of NFAs M and N. There may exist no bisimulation between M and N, in which case the algorithm halts and reports failure. For the case $M = N$, the algorithm computes the maximal autobisimulation. The algorithm is a direct generalization of the algorithm of Lecture 14.

As in Lecture 14, the algorithm will mark pairs of states (p, q), where $p \in Q_M$ and $q \in Q_N$. A pair (p, q) will be marked when a proof is discovered that p and q cannot be related by any bisimulation.

1. Write down a table of all pairs (p, q), initially unmarked.
2. Mark (p, q) if $p \in F_M$ and $q \notin F_N$ or vice versa.
3. Repeat the following until no more changes occur: if (p, q) is unmarked, and if for some $a \in \Sigma$, either
   - there exists $p' \in \Delta_M(p, a)$ such that for all $q' \in \Delta_N(q, a)$, (p', q') is marked, or
   - there exists $q' \in \Delta_N(q, a)$ such that for all $p' \in \Delta_M(p, a)$, (p', q') is marked,

   then mark (p, q).
4. Define $p \equiv q$ iff (p, q) is never marked. Check whether $S_M \equiv S_N$. If so, then $\equiv$ is the maximal bisimulation between M and N. If not, then no bisimulation between M and N exists.

One can easily prove by induction on the stages of this algorithm that if the pair (p, q) is ever marked, then $p \not\approx q$ for any bisimulation $\approx$ between M and N, because we only mark pairs that violate some condition in the definition of bisimulation. Therefore, any bisimulation $\approx$ is a refinement of $\equiv$. In particular, the maximal bisimulation between M and N, if it exists, is a refinement of $\equiv$. If $S_M \not\equiv S_N$, then the same is true for any refinement of $\equiv$; in this case, no bisimulation exists.

On the other hand, suppose $S_M \equiv S_N$. To show that the algorithm is correct, we need only show that $\equiv$ is a bisimulation; then it must be the maximal one. We have $S_M \equiv S_N$ by assumption. Also, $\equiv$ respects the transition functions of M and N because of step 3 of the algorithm and respects final states of M and N because of step 2 of the algorithm.

We have shown:

Theorem B.12 *The algorithm above correctly computes the maximal bisimulation between two NFAs if a bisimulation exists. If no bisimulation exists, the algorithm halts and reports failure. If both automata are the same, the algorithm computes the maximal autobisimulation.*